Technische Universität München
Fakultät für Elektrotechnik und Informationstechnik

Direction-of-Arrival Estimation and Model Order Selection for Systems with Subarray Sampling

Andreas Barthelme

Vollständiger Abdruck der von der Fakultät für Elektrotechnik und Informationstechnik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzender:

Prof. Dr.-Ing. Bernhard Seeber

Prüfende der Dissertation:

1. Prof. Dr.-Ing. Wolfgang Utschick
2. Prof. Magnus Jansson, Ph.D.

Die Dissertation wurde am 16.12.2020 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 23.08.2021 angenommen.

Acknowledgments

First of all, I would like to express my sincere gratitude to my supervisor Prof. Dr.-Ing. Wolfgang Utschick for his trust and advice throughout the last years, which was not only limited to our joint research, but helped me in my personal development as well.

I would like to say thank you to all of my colleagues at the methods of signal processing group at Technische Universität München for creating a really nice working environment. Discussing new ideas with you and jesting during the coffee breaks was always a pleasure and I am grateful that I can call some of you close friends.

Finally, I want to thank my family and friends, in particular, my wife Susi, for their continuous moral support.

Abstract

In this work, we consider direction-of-arrival estimation as well as model order selection for systems with subarray sampling. These systems operate with fewer radio frequency chains than antenna elements by utilizing a switching network that sequentially connects different subarrays to the radio frequency chains. Due to this sequential processing, the considered systems represent a special class of time-varying arrays, which allows for a cost-efficient design of direction finders.

In this context, we discuss improvements to existing algorithms and present new estimators that are based on machine learning techniques. In particular, for the direction-of-arrival estimation, we revisit classical estimators, whose objective function is derived from the stochastic model, discuss sparse recovery techniques for this problem, and present solution strategies that rely on neural networks. Thereby, the machine learning approaches follow three different rationales: end-to-end regression, classification, and full covariance matrix reconstruction. For the model order selection problem, we focus on an extension of the well known information criteria and neural network-based approaches.

By means of simulations, we show that the new machine learning based techniques are able to significantly outperform existing solutions in terms of accuracy and computational complexity. Especially, in the critical regions of low signal-to-noise-ratio, small number of snapshots and as many sources as radio frequency chains, existing methods cannot compete with the neural network-based estimators.

Zusammenfassung

In dieser Arbeit beschäftigen wir uns mit der Winkel- und Modellordnungsschätzung bei untergruppenabgetasteten Systemen. Diese Systeme arbeiten mit weniger Hochfrequenzketten als Antennen indem ein Schaltnetzwerk verwendet wird, das nacheinander verschiedene Untergruppen mit den Hochfrequenzketten verbindet. Durch diese sequentielle Verarbeitung stellen die betrachteten Systeme eine spezielle Form von zeitveränderlichen Antennenarrays dar, die einen kostengünstigen Entwurf von Peilern ermöglicht.

In diesem Zusammenhang diskutieren wir Verbesserungen von bestehenden Algorithmen und präsentieren neue Schätzer, die auf maschinellem Lernen basieren. Im Einzelnen betrachten wir klassische Schätzer, deren Zielfunktion aus dem stochastischen Modell abgeleitet wird, diskutieren Sparse Recovery Methoden und präsentieren Lösungsstrategien, die auf neuronalen Netzen aufbauen. Dabei folgen die verwendeten neuronalen Netze drei verschiedenen Grundgedanken: der Ende-zu-Ende Regression, der Klassifikation und der Rekonstruktion einer Kovarianzmatrix des vollabgetasteten Systems. Zur Modellordnungsschätzung betrachten wir eine Erweiterung der bekannten Informationskriterien und Verfahren basierend auf neuronalen Netzen.

Mit der Hilfe von Simulationen zeigen wir, dass die neuen Techniken basierend auf maschinellem Lernen bestehende Lösungen in ihrer Genauigkeit übertreffen und eine geringere Komplexität aufweisen. Besonders in den kritischen Regionen bei niedrigem Signal-zu-Rausch-Verhältnis, wenigen Beobachtungen und gleich vielen Quellen wie Hochfrequenzketten können bestehende Methoden nicht mit den Resultaten der neuronalen Netze mithalten.

Contents

Acronyms

AIC	Akaike Information Criterion
BIC	Bayesian Information Criterion
CDF	Cumulative Density Function
COMET	Covariance Matching Estimation Technique
CRB	Cramér-Rao Bound
CS	Compressed Sensing
DML	Deterministic Maximum Likelihood
DoA	Direction-of-Arrival
GLS	Generalized Least Squares
LMI	Linear Matrix Inequality
MAP	Maximum A Posteriori
MCE	Mean Cyclic Error
MDL	Minimum Description Length
ML	Maximum Likelihood
MMV	Multi Measurement Vector
MSE	Mean Squared Error
MSPE	Mean Squared Periodic Error
MUSIC	Multiple Signal Classification
MVDR	Minimum Variance Distortionless Response
NN	Neural Network
OMP	Orthogonal Matching Pursuit

PDF	Probability Density Function
ReLU	Rectified Linear Unit
RF	Radio Frequency
RMSE	Root Mean Squared Error
RMSPE	Root Mean Squared Periodic Error
SDP	Semidefinite Programming
SML	Stochastic Maximum Likelihood
SNR	Signal-to-Noise-Ratio
SOCP	Second Order Cone Program
SPICE	Sparse Iterative Covariance-based Estimation
UCA	Uniform Circular Array
ULA	Uniform Linear Array

1 Introduction

Direction-of-Arrival (DoA) estimation describes the process of estimating the directions from which electromagnetic waves are impinging onto an antenna array. This task is a crucial component for the localization of electromagnetic sources and reflectors, which is relevant in military and civil applications such as autonomous driving, aviation, astronomy, and mobile communications.

Traditionally, in these areas, suitable stochastic models for the received signals at the individual antenna elements are available. Derived from these models, a plethora of potent algorithms for DoA estimation have been proposed in the past (see, e.g., [1, 2]). These DoA estimation strategies encompass beamformer, super resolution and maximum likelihood techniques. Yet, there are still some applications, where existing solutions do not provide a satisfying performance.

In this work, we focus on systems that employ a subarray sampling strategy. The idea is to sequentially sample subarrays, instead of sampling all antenna elements simultaneously, to reduce the amount of required Radio Frequency (RF) chains. Since manufacturing high grade RF chains makes up a large amount of the costs of an antenna array, reducing the number of RF chains is a sound strategy for saving. Naturally, this cost reduction is not for free. Due to the lower number of RF chains, we cannot leverage the full array gain in a single time step, but have to switch between different subarray constellations. This does not only require multiple time steps, but also brings new challenges for the signal processing. In fact, subarray sampling is one of these applications for which classical methods fail to provide fully satisfying results, especially for more sources than RF chains [3].

Subarray sampling can be seen as a special case of the broader concept of time-varying arrays. DoA estimation methods for time-varying arrays can be categorized into two groups, which depend on the ratio of transmitting sources to simultaneously sampled antenna elements. The case of fewer sources than sampled antennas per time step has been first studied in [4]. In [5], the same authors have extended their analysis of the single source case to multiple sources. Therein, they have proposed to apply eigenstructure-based methods such as array interpolation and focusing matrices, a technique that is well known from wideband array processing [6, 7], to time-varying arrays. In [8, 9], a more direct approach to employing Multiple Signal Classification (MUSIC) and Minimum Variance Distortionless Response (MVDR) estimators in these systems has been studied. The more interesting—but also more demanding—scenario is the case with an equal amount of sources or more sources than simultaneously sampled antennas. Fundamental work for this case has been presented in [10]. There, the idea is to fit the subarray covariance matrices, which are parameterized by the DoAs, to the observed sample covariance matrices with the help of a Generalized Least Squares (GLS) criterion. Another approach that heavily relies on a special subarray sampling scheme has been proposed in [11]. There, an estimate of the covariance matrix of the fully sampled antenna array is first estimated, and then, a MUSIC estimator [12] is applied to the reconstructed full covariance matrix to obtain the DoA estimates. More recently, the non-coherent processing of partly calibrated arrays has been investigated in [13]. The estimation approach followed in [13] uses a sparse signal representation of the system model and is, due to the non-coherent processing approach, applicable to subarray sampling. Finally, in [3], we have proposed a machine learning-based approach to this DoA estimation problem that is able to outperform the previously discussed estimators, especially for low Signal-to-Noise-Ratio (SNR) and a small number of observations.

An accurate knowledge on the number of simultaneously transmitting sources plays a crucial role for the performance of the DoA estimation techniques summarized above. In general, this number is unknown a priori and has to be inferred from the observations. The estimation of this quantity is called model order selection, as the number of active sources defines the order of the underlying stochastic model. For systems with subarray sampling, this model order estimation problem has been first investigated in [14]. Later, in [3], we improved this method by deriving so called information criteria that utilize better performing DoA estimators and we extended our machine learning technique for model order selection [15] to the subarray sampling case.

1.1 Overview

In this work, we discuss the DoA estimation problem as well as model order selection for systems with subarray sampling. To that end, we consider classical methods based on the underlying stochastic models and present new machine learning-based solutions. The remainder of this thesis is structured as follows.

In *Chapter 2*, we describe the received signal model for the considered systems, which forms the basis for the subsequent discussions. Furthermore, we consider the parameter identifiability of the respective stochastic models, and in this context, we present and proof new sufficient and necessary conditions. We conclude this chapter with a discussion of the Cramér-Rao Bound (CRB), which gives a performance bound for any unbiased DoA estimator that can be derived directly from the distribution of the received signals.

Then, we discuss the first class of DoA estimators, which we refer to by "classical", in *Chapter 3*. In the first part of this chapter, we focus on well known Maximum Likelihood (ML) estimators. Apart from giving the respective optimization problems, we also talk about how these estimators can be evaluated in practice and we derive a Mean Squared Error (MSE) approximation for grid-based ML estimators. In the second part, we present the GLS estimator that has been proposed in [10] and show how the original optimization problem can be augmented to obtain feasible solutions also in the low snapshot domain (cf. [3]).

The next class of DoA estimators, which we focus on in *Chapter 4*, uses sparse recovery techniques. Hence, we first show how to formulate the DoA estimation task as such a sparse recovery problem. Then, we discuss two possible solution approaches: the Orthogonal Matching Pursuit (OMP) method, which is a well known greedy approach for sparse recovery, and an extension of the Sparse Iterative Covariance-based Estimation (SPICE) estimator [13], which has been developed as a non-coherent processing technique for partly calibrated arrays.

In *Chapter 5*, we present DoA estimation techniques based on machine learning. First, we discuss an end-to-end regression approach, which we have already published in [3]. Then, we talk about another class of data-driven DoA estimators, which pose the DoA estimation task as a classification problem. Here, we do not only present existing estimation strategies, but derive a new estimator inspired by classification chains. Finally, a completely new approach is investigated, where we use a Neural Network (NN) to estimate the covariance matrix of the fully sampled array from the sample covariance information of the individual subarrays. This full covariance matrix estimate is then used to infer the DoAs by a subsequently applied MUSIC estimator.

Chapter 6 discusses how to combine the previously presented estimation principles with consecutive refinement steps on the respective log-likehood function to obtain a computationally more efficient approximation of the ML estimator. The goal of these combined estimation approaches, which we denote by hybrid estimators, is to reduce the grid mismatch problem of the sparse recovery based methods and to improve the performance of the completely data-based machine learning techniques by leveraging knowledge on the stochastic model.

The performance of the different DoA estimation approaches is compared in

Chapter 7. Thereby, we differentiate between the case of fewer receivers than RF chains and the more demanding opposite case.

Model order selection, i.e., estimating the number of active sources, is presented in *Chapter 8*. After an explicit statement of the underlying estimation problem, we discuss the Maximum A Posteriori (MAP) estimator and the application of information criteria for systems with subarray sampling. Finally, we present a model order selection approach via NNs from our paper [3] and a new method that is based on the estimation of the full covariance matrix.

In *Chapter 9*, a comparison of the respective selection accuracy of the different model order estimators is presented. Here, we do not only investigate the accuracy over SNR and number of snapshots, but we also address how the NN-based approach from [3] can be used in a real world implementation of a direction finder.

Finally, we conclude the thesis by a brief summary and an outlook on possible extensions of the presented work in *Chapter 10*.

1.2 Notation

Throughout this work, we use lower case, bold letters to denote column vectors and upper case, bold letters for matrices. For sets, we use calligraphic, capital letters. A list of symbols and math operators used in this work is given in Table 1.1.

Symbol / Operator	Description
$\Re\{\bullet\}, \Im\{\bullet\}$	real part, imaginary part
$\mathbb{R}$	set of real numbers
$\mathbb{C}$	set of complex numbers
$\mathbf{0}_{M\times N}, \mathbf{0}_M$	all zeros matrix of size $M \times N$, all zeros vector of size M
$\mathbf{1}_{M\times N}, \mathbf{1}_M$	all ones matrix of size $M \times N$, all ones vector of size M
$\mathbf{I}_M$	identity matrix of size $M \times M$
$\mathbf{P}_{\boldsymbol{X}}, \mathbf{P}_{\boldsymbol{X}}^{\perp}$	projection matrices onto the range of $\boldsymbol{X}$ and nullspace of $\boldsymbol{X}$
$\mathbf{e}_i$	i-th canonical basis vector
$(\bullet)^*, (\bullet)^{\mathrm{T}}, (\bullet)^{\mathrm{H}}, (\bullet)^+$	complex conjugated, transpose, conjugate transpose, Moore-Penrose pseudo inverse
$[\bullet]_i, [\bullet]_{i,\cdot}$	i-th element, i-th row
$\lVert \bullet \rVert_2, \lVert \bullet \rVert_{\mathrm{F}}$	Euclidean norm, Frobenius norm
$\boldsymbol{X} \succ \mathbf{0}, \boldsymbol{X} \succeq \mathbf{0}$	$\boldsymbol{X}$ is positive definite, $\boldsymbol{X}$ is positive semidefinite

$\mathrm{null}\{\boldsymbol{X}\}$	nullspace of $\boldsymbol{X}$
$\mathrm{span}\{\boldsymbol{X}\}$	span (range space) of $\boldsymbol{X}$
$\mathrm{tr}\{\boldsymbol{X}\}$	trace of $\boldsymbol{X}$
$\det\{\boldsymbol{X}\}$	determinant of $\boldsymbol{X}$
$\mathrm{rank}\{\boldsymbol{X}\}$	rank of $\boldsymbol{X}$
$\boldsymbol{X}^{1/2}$	matrix square root of the positive semidefinite matrix $\boldsymbol{X}$
$\mathrm{chol}\{\boldsymbol{X}\}$	Cholesky decomposition of $\boldsymbol{X}$
$\mathrm{vec}\{\boldsymbol{X}\}$	vector obtained by column-wise stacking of the elements of $\boldsymbol{X}$
$\mathrm{diag}\{\boldsymbol{x}\}$	diagonal matrix, with the elements of $\boldsymbol{x}$ on its diagonal
$\mathrm{diag}\{x_1, \ldots, x_M\}$	diagonal matrix, with $x_1, \ldots, x_M$ on its diagonal
$\mathrm{blockdiag}\{\boldsymbol{X}_1, \ldots, \boldsymbol{X}_M\}$	block-diagonal matrix with blocks $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_M$ on its diagonal
Pr[•]	probability of a certain event
$\otimes, \circ$	Kronecker and Kathri-Rao matrix products
$\mathrm{mod}(x, a)$	x modulo a
$\lfloor x \rfloor$	floor of x
E[•]	expectation of a random variable
$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{C})$	normal distribution with mean μ and covariance matrix $\boldsymbol{C}$
$\mathcal{CN}(\boldsymbol{\mu}, \boldsymbol{C})$	circularly-symmetric complex normal distribution with mean μ and covariance matrix $\boldsymbol{C}$
$\mathcal{U}(a, b)$	uniform distribution between a and b
$\Phi(x)$	Gauss error function at x

Table 1.1: Symbols and Math Operators

System Model for Subarray Sampling 2

2.1 System Model

We consider a scenario, where we want to estimate the DoAs $\boldsymbol{\theta}$ of L transmitters located in the far-field of an antenna array consisting of M antennas. The narrowband transmit signals of the individual transmitters at the time n_{s} are gathered in the vector $\boldsymbol{s}(n_{\mathrm{s}})$. Let us now denote the so called array manifold for the DoAs $\boldsymbol{\theta}$ by $\boldsymbol{A}(\boldsymbol{\theta})$, which comprises the antenna array's responses due to its geometrical and RF properties parameterized by the DoAs $\boldsymbol{\theta}$. Then, we can express the received signal $\boldsymbol{y}(n_{\mathrm{s}})$ at the time instances n_{s} as[1]

$$\boldsymbol{y}(n_{\mathrm{s}}) = \boldsymbol{A}(\boldsymbol{\theta})\boldsymbol{s}(n_{\mathrm{s}}) + \boldsymbol{\eta}(n_{\mathrm{s}}), \tag{2.1}$$

where $\boldsymbol{\eta}(n_{\mathrm{s}})$ is some i.i.d. additive white Gaussian noise with covariance matrix $\boldsymbol{C}_{\boldsymbol{\eta}} = \sigma_\eta^2 \mathbf{I}$. Furthermore, we assume that the DoAs $\boldsymbol{\theta}$ are not changing in the time interval $n_{\mathrm{s}} = 1, \ldots, N_{\mathrm{s}}$ such that we can use N_{s} samples of $\boldsymbol{y}$ to jointly estimate $\boldsymbol{\theta}$.

Since high grade RF chains, which are used to obtain the received signals at each antenna, are quite costly, we investigate a scenario where we use fewer RF chains than receive antennas. This means that at each time instance n_{s}, we only observe a subset of antennas of the antenna array. Naturally, with this scheme, we do not leverage the full array gain. Hence, we employ a switching network to sequentially sample different subsets of antennas at each time step. This method has been previously investigated in [8, 9, 10, 11] theoretically and is successfully utilized in some of todays DoA estimation infrastructure.

To model this sequential sampling process, which we will refer to as subarray sampling, we introduce the switching matrices $\boldsymbol{G}^{(k)}$. To this end, let us first define the index set $\mathcal{G}^{(k)}$ that collects the indices of the active antennas of the k-th subarray. Then, the switching matrix $\boldsymbol{G}^{(k)}$ selects the response of the k-th subarray $\boldsymbol{y}^{(k)}(n_{\mathrm{s}})$ at

[1] For a basic introduction into the physical foundations behind the presented system model, i.e., how the presented base band model can be derived from the geometry and wave propagation model, we refer the reader to on of many textbooks on array processing, e.g., [2].

time instance n_s from the full vector of array responses $\boldsymbol{y}(n_\mathrm{s})$, i.e.,

$$\boldsymbol{y}^{(k)}(n_\mathrm{s}) = \boldsymbol{G}^{(k)}\boldsymbol{y}(n_\mathrm{s}), \tag{2.2}$$

with

$$\left[\boldsymbol{G}^{(k)}\right]_{i,\cdot} = \mathbf{e}^{\mathrm{T}}_{[\mathcal{G}^{(k)}]_i}. \tag{2.3}$$

In total, we consider K different subarrays, for which we gather N samples each, such that the number of total samples is $N_\mathrm{s} = KN$. W.l.o.g. we choose the number of active antennas of each subarray to be $|\mathcal{G}^{(k)}| = W$. Additionally, let us assume that there exists a fixed mapping from the time n_s to a pair of subarray and sample (k, n), then, we can replace the time dependence of $\boldsymbol{y}^{(k)}$ on n_s with $\boldsymbol{y}^{(k)}(n)$ for notational brevity. With this we can write

$$\tilde{\boldsymbol{y}}(n) = \tilde{\boldsymbol{A}}(\boldsymbol{\theta})\tilde{\boldsymbol{s}}(n) + \tilde{\boldsymbol{\eta}}(n), \tag{2.4}$$

with

$$\tilde{\boldsymbol{y}}(n) = \left[\boldsymbol{y}^{(1),\mathrm{T}}(n), \boldsymbol{y}^{(2),\mathrm{T}}(n), \dots, \boldsymbol{y}^{(K),\mathrm{T}}(n)\right]^{\mathrm{T}}, \tag{2.5}$$

$$\tilde{\boldsymbol{s}}(n) = \left[\boldsymbol{s}^{(1),\mathrm{T}}(n), \boldsymbol{s}^{(2),\mathrm{T}}(n), \dots, \boldsymbol{s}^{(K),\mathrm{T}}(n)\right]^{\mathrm{T}}, \tag{2.6}$$

$$\tilde{\boldsymbol{\eta}}(n) = \left[\boldsymbol{\eta}^{(1),\mathrm{T}}(n), \boldsymbol{\eta}^{(2),\mathrm{T}}(n), \dots, \boldsymbol{\eta}^{(K),\mathrm{T}}(n)\right]^{\mathrm{T}}, \tag{2.7}$$

$$\boldsymbol{A}^{(k)}(\boldsymbol{\theta}) = \boldsymbol{G}^{(k)}\boldsymbol{A}(\boldsymbol{\theta}), \tag{2.8}$$

$$\tilde{\boldsymbol{A}}(\boldsymbol{\theta}) = \begin{bmatrix} \boldsymbol{A}^{(1)}(\boldsymbol{\theta}) & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{A}^{(2)}(\boldsymbol{\theta}) & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \boldsymbol{A}^{(K)}(\boldsymbol{\theta}) \end{bmatrix}, \tag{2.9}$$

which resembles the well known form of the fully sampled antenna array in (2.1). We will use this compound form of the system with subarray sampling in (2.4) whenever it simplifies the notation.

Depending on the assumption about the transmit signals, we obtain two different stochastic models. In the first case, which we will refer to by the name of the resulting Deterministic Maximum Likelihood (DML) estimator, we assume that the transmit signals are unknown and arbitrary, i.e., the transmit signals are not modeled as a random variable, but deterministic unknowns. This leads to the following distribution of the received signals

$$\tilde{\boldsymbol{y}}(n) \sim \mathcal{CN}\left(\tilde{\boldsymbol{A}}(\boldsymbol{\theta})\tilde{\boldsymbol{s}}(n), \sigma_\eta^2 \mathbf{I}_{KW}\right). \tag{2.10}$$

In the second case, the transmit signals are treated as realizations from a random variable. This means that now the parameters of the distribution of $\tilde{\boldsymbol{s}}(n)$—instead of the individual realizations—parameterize the Probability Density Function (PDF) of the received signals $\tilde{\boldsymbol{y}}(n)$. Again, we name this case after its respective ML estimator, i.e., the Stochastic Maximum Likelihood (SML) case. Usually the transmit signals are modeled by a i.i.d. zero-mean complex Gaussian random variables, i.e.,

$$\boldsymbol{s}^{(k)}(n) \sim \mathcal{CN}\left(\mathbf{0}, \boldsymbol{C}_{\boldsymbol{s}}\right). \tag{2.11}$$

Furthermore, $\boldsymbol{s}^{(k)}(n)$ are uncorrelated over time, which can be written as

$$\mathrm{E}\left[\boldsymbol{s}^{(k)}(n)\boldsymbol{s}^{(\ell),\mathrm{H}}(m)\right] = \mathbf{0}, \quad \forall n \neq m, k \neq \ell. \tag{2.12}$$

Under these assumptions, the received signal is again Gaussian distributed according to

$$\tilde{\boldsymbol{y}}(n) \sim \mathcal{CN}\left(\mathbf{0}, \tilde{\boldsymbol{A}}(\boldsymbol{\theta})\left(\mathbf{I} \otimes \boldsymbol{C}_{\boldsymbol{s}}\right)\tilde{\boldsymbol{A}}^{\mathrm{H}}(\boldsymbol{\theta}) + \sigma_\eta^2 \mathbf{I}_{KW}\right). \tag{2.13}$$

2.2 Identifiability

The problem of identifiability naturally arises with any kind of estimation problem. A stochastic model is said to be identifiable if two distinct choices for the parameters $\boldsymbol{\varphi}$ never lead to the same PDF [16].

2.2.1 DML Identifiability

For fully sampled arrays, the identifiability of the DML system model has been studied in [17]. There, a sufficient identifiability condition is derived that provides an upper bound to the maximum number of active sources depending on the number of antennas and the correlation between the transmit signals. Here, we provide a similar identifiability condition for systems with subarray sampling, which ensures that the PDF of the DML case is unique in $\boldsymbol{\theta}$, i.e., two distinct choices $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$, $\boldsymbol{\theta} \neq \boldsymbol{\theta}'$ cannot lead to the same distribution (2.10).

Theorem 1 (Sufficient Identifiability Condition - DML). *Assume that for any choice of $q \leq W$ distinct DoAs $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_q]^{\mathrm{T}}$, at least one of the subarray steering matrices $\boldsymbol{A}^{(k)}(\boldsymbol{\theta})$ has full column rank and the corresponding signal matrix $\boldsymbol{S}^{(k)} = [\boldsymbol{s}^{(k)}(1), \ldots, \boldsymbol{s}^{(k)}(N)]$ has rank $r > 0$ and no row that consists of zeros only. Then, the inequality*

$$L < \frac{q+r}{2}, \tag{2.14}$$

provides a sufficient condition for the identifiability of the stochastic model of the DML case (2.10) with respect to the DoAs $\boldsymbol{\theta}$.

Proof. We closely follow the argumentation of [18, Theorem 1], to show that the mean of (2.10), and therewith, the PDF of $\tilde{\boldsymbol{Y}}$, with $\tilde{\boldsymbol{Y}} = [\tilde{\boldsymbol{y}}(1), \tilde{\boldsymbol{y}}(2), \dots, \tilde{\boldsymbol{y}}(N)]$, is unique with respect to $\boldsymbol{\theta}$ under the given constraints, i.e.,

$$\tilde{\boldsymbol{A}}(\boldsymbol{\theta})\tilde{\boldsymbol{S}} \neq \tilde{\boldsymbol{A}}(\boldsymbol{\theta}')\tilde{\boldsymbol{S}}', \tag{2.15}$$

where $\tilde{\boldsymbol{S}} = [\tilde{\boldsymbol{s}}(1), \dots, \tilde{\boldsymbol{s}}(N)]$, for any $\boldsymbol{\theta} \neq \boldsymbol{\theta}'$ and any $\tilde{\boldsymbol{S}}'$. Note that to show that (2.15) holds, it suffices to show that

$$\exists k \in \{1, \dots, K\} \quad \boldsymbol{A}^{(k)}(\boldsymbol{\theta})\boldsymbol{S}^{(k)} \neq \boldsymbol{A}^{(k)}(\boldsymbol{\theta}')\boldsymbol{S}'^{(k)}. \tag{2.16}$$

Now, let us consider two different cases analogously to [18].

Case 1: We have d pairs of equal DoAs $\theta_i = \theta'_j$ with $0 \leq d < 2L - q$. Let us denote the index of the subarray that fulfills the rank constraints in Theorem 1 by k. By comparing the dimension of the nullspace of $[\boldsymbol{A}^{(k)}(\boldsymbol{\theta}), \boldsymbol{A}^{(k)}(\boldsymbol{\theta}')] \in \mathbb{C}^{W \times 2L}$, denoted by ξ, with the rank of the matrix $[\boldsymbol{S}^{(k),\mathrm{T}}, -\boldsymbol{S}'^{(k),\mathrm{T}}]^{\mathrm{T}} \in \mathbb{C}^{2L \times N}$, denoted by ν, we see that (2.16) holds for

$$\xi = 2L - \operatorname{rank}\left\{\left[\boldsymbol{A}^{(k)}(\boldsymbol{\theta}), \boldsymbol{A}^{(k)}(\boldsymbol{\theta}')\right]\right\} < \operatorname{rank}\left\{\begin{bmatrix}\boldsymbol{S}^{(k)} \\ -\boldsymbol{S}'^{(k)}\end{bmatrix}\right\} = \nu. \tag{2.17}$$

Since any q subarray steering vectors with distinct DoAs are linearly independent, and we consider the case $q < 2L - d$ as stated above, we have

$$\xi = 2L - q. \tag{2.18}$$

If the inequality from (2.14) holds, we now see that

$$\xi = 2L - q < r. \tag{2.19}$$

Due to the construction of $[\boldsymbol{S}^{(k),\mathrm{T}}, -\boldsymbol{S}'^{(k),\mathrm{T}}]^{\mathrm{T}}$, we additionally have that r is smaller or equal to ν, which in turn proves that for the considered case $\xi < \nu$ holds.

Case 2: Now, we consider the case of $L > d \geq 2L - q$. Let us partition the DoAs $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$ into two sets $\boldsymbol{\theta} = [\boldsymbol{\theta}_{\mathrm{u}}, \boldsymbol{\theta}_{\mathrm{e}}]$ and $\boldsymbol{\theta}' = [\boldsymbol{\theta}'_{\mathrm{u}}, \boldsymbol{\theta}'_{\mathrm{e}}]$, respectively, where $\boldsymbol{\theta}_{\mathrm{e}} = \boldsymbol{\theta}'_{\mathrm{e}}$ corresponds to the d pairs with equal DoAs. Then, we can rewrite (2.16) as

$$\left[\boldsymbol{A}^{(k)}(\boldsymbol{\theta}_{\mathrm{u}}), \boldsymbol{A}^{(k)}(\boldsymbol{\theta}_{\mathrm{e}}), \boldsymbol{A}^{(k)}(\boldsymbol{\theta}'_{\mathrm{u}})\right]\begin{bmatrix}\boldsymbol{S}_{\mathrm{u}}^{(k)} \\ \boldsymbol{S}_{\mathrm{e}}^{(k)} - \boldsymbol{S}_{\mathrm{e}}'^{(k)} \\ -\boldsymbol{S}_{\mathrm{u}}'^{(k)}\end{bmatrix} \neq \mathbf{0}, \tag{2.20}$$

where $\boldsymbol{S}_{\mathrm{u}}^{(k)}$ and $\boldsymbol{S}_{\mathrm{e}}^{(k)}$ collect the rows of $\boldsymbol{S}^{(k)}$ that correspond to the columns in $\boldsymbol{A}^{(k)}(\boldsymbol{\theta}_{\mathrm{u}})$ and $\boldsymbol{A}^{(k)}(\boldsymbol{\theta}_{\mathrm{e}})$, respectively.

Since the number of columns in $[\boldsymbol{A}^{(k)}(\boldsymbol{\theta}_{\mathrm{u}}), \boldsymbol{A}^{(k)}(\boldsymbol{\theta}_{\mathrm{e}}), \boldsymbol{A}^{(k)}(\boldsymbol{\theta}'_{\mathrm{u}})]$ is $2L - d \leq q$ and they all belong to distinct DoAs, these columns are linearly independent, i.e., the matrix has full column rank. Therefore, the left hand side of (2.20) can become zero only for $\boldsymbol{S}_{\mathrm{u}}^{(k)} = \mathbf{0}$. Note that this would mean that the sources at $\boldsymbol{\theta}_{\mathrm{u}}$ would not send any signals, which violates the assumptions in Theorem 1. ■

Note that this sufficient condition for identifiability can only be fulfilled for $L < W$, as q and r are upper bounded by W. Furthermore, this condition does not ensure that the PDF (2.10) is unique in $\tilde{\boldsymbol{S}}$. In fact, Theorem 1 can be fulfilled even if there exists some subarray, for which $\mathrm{rank}\{\boldsymbol{A}^{(k)}(\boldsymbol{\theta})\} < L$. Then, there exist infinitely many $\boldsymbol{S}'^{(k)} = \boldsymbol{S}^{(k)} + \boldsymbol{U}$, with $\boldsymbol{U} \in \mathrm{null}\{\boldsymbol{A}^{(k)}(\boldsymbol{\theta})\}$, that lead to the same distribution (2.10). To obtain a sufficient condition that guarantees the uniqueness of $\tilde{\boldsymbol{S}}$ as well, we could simply extend the requirement of full column rank of at least one subarray steering matrix for each $\boldsymbol{\theta}$ to demanding full column rank for every subarray steering matrix. However, this is a very strict requirement that significantly limits the number of appropriate array designs.

We conclude the identifiability discussion of the DML case with a brief look at necessary conditions. It is clear that the PDF (2.10) is not unique in $\boldsymbol{\theta}$ if there exist two distinct DoAs $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$, $\boldsymbol{\theta} \neq \boldsymbol{\theta}'$, such that

$$\mathrm{span}\left\{\boldsymbol{A}^{(k)}(\boldsymbol{\theta})\right\} \subseteq \mathrm{span}\left\{\boldsymbol{A}^{(k)}(\boldsymbol{\theta}')\right\}, \quad \forall k = 1, \ldots, K. \tag{2.21}$$

This leads to the following trivial necessary condition for the identifiability with respect to $\boldsymbol{\theta}$.

Theorem 2 (Necessary Identifiability Condition - DML)**.** *Assume that for any choice of M distinct DoAs $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_M]^{\mathrm{T}}$, the steering matrix of the fully sampled array $\boldsymbol{A}(\boldsymbol{\theta})$ has full rank. Then, the inequality*

$$L < M, \tag{2.22}$$

is a necessary condition for the identifiability of the stochastic model of the DML case (2.10) with respect to the DoAs $\boldsymbol{\theta}$.

Proof. The steering matrix $\boldsymbol{A}(\boldsymbol{\theta})$ has full row rank for $L \geq M$. Furthermore, the sampling matrices $\boldsymbol{G}^{(k)}$ have full row rank W by construction. Hence, the subarray steering matrices $\boldsymbol{A}^{(k)}(\boldsymbol{\theta}) = \boldsymbol{G}^{(k)}\boldsymbol{A}(\boldsymbol{\theta})$ have full row rank as well. This means that the $\mathrm{span}\left\{\boldsymbol{A}^{(k)}(\boldsymbol{\theta})\right\} = \mathbb{C}^W$ for every $\boldsymbol{\theta}$ and k, and therefore (2.21) is fulfilled. ■

Analogously, one can show that the necessary condition $L < W$ holds if for every $\boldsymbol{\theta} \in \mathbb{R}^W$ the subarray steering matrices $\boldsymbol{A}^{(k)}(\boldsymbol{\theta})$ have full rank. However, as discussed above this assumption is of practical significance.

2.2.2 SML Identifiability

The identifiability of the fully sampled antenna array in the SML case has been investigated in [17] and [19]. For systems with subarray sampling, an extension of these results is not straight-forward and still remains an open problem. However, under the assumption of uncorrelated transmit sources, i.e.,

$$\boldsymbol{C_s} = \mathrm{diag}\{\sigma_{s,1}^2, \sigma_{s,2}^2, \dots, \sigma_{s,L}^2\}, \tag{2.23}$$

a sufficient condition has been derived for the noise-free case in [13]. There, the idea is to employ a vectorization of the covariance matrices $\boldsymbol{C_y}^{(k)} = \boldsymbol{A}^{(k)}(\boldsymbol{\theta})\boldsymbol{C_s}\boldsymbol{A}^{(k),\mathrm{H}}(\boldsymbol{\theta}) + \sigma_\eta^2\mathbf{I}_W$, which leads to the following representation of the covariance matrix

$$\mathrm{vec}\left\{\boldsymbol{C_y}^{(k)}\right\} = \breve{\boldsymbol{V}}^{(k)}(\boldsymbol{\theta})\boldsymbol{\lambda} + \sigma_\eta^2\mathrm{vec}\left\{\mathbf{I}_W\right\}, \tag{2.24}$$

where

$$\breve{\boldsymbol{V}}^{(k)}(\boldsymbol{\theta}) = \boldsymbol{A}^{(k),*}(\boldsymbol{\theta}) \circ \boldsymbol{A}^{(k)}(\boldsymbol{\theta}), \tag{2.25}$$

$$\boldsymbol{\lambda} = \left[\sigma_{s,1}^2, \sigma_{s,2}^2, \dots, \sigma_{s,L}^2\right]^{\mathrm{T}}. \tag{2.26}$$

After stacking the left hand side of (2.24) for every k in a large vector $\boldsymbol{c_y}$, we obtain

$$\boldsymbol{c_y} = \breve{\boldsymbol{V}}(\boldsymbol{\theta})\boldsymbol{\lambda} + \sigma_\eta^2\mathbf{1}_K \otimes \mathrm{vec}\left\{\mathbf{I}_W\right\}, \tag{2.27}$$

with the co-array manifold $\breve{\boldsymbol{V}}(\boldsymbol{\theta}) = [\breve{\boldsymbol{V}}^{(1),\mathrm{T}}(\boldsymbol{\theta}), \dots, \breve{\boldsymbol{V}}^{(K),\mathrm{T}}(\boldsymbol{\theta})]^{\mathrm{T}}$. Ignoring the noise, this new representation of the covariance matrix information in SML case now resembles the structure of the mean of the PDF in the DML scenario. Unsurprisingly, the conditions, under which an identification of $\boldsymbol{\theta}$ can be guaranteed, again show parallels to the sufficient conditions of the DML case.

Theorem 3 (Sufficient Identifiability Condition - SML [13])**.** *Assume that for any choice of q distinct DoAs $\boldsymbol{\theta} = [\theta_1, \dots, \theta_q]^{\mathrm{T}}$, the co-array manifold $\breve{\boldsymbol{V}}(\boldsymbol{\theta})$ has full column rank and the signal covariance matrix $\boldsymbol{C_s}$ is full rank and diagonal. Then, the inequality*

$$L \leq \left\lfloor \frac{q}{2} \right\rfloor, \tag{2.28}$$

provides a sufficient condition for the identifiability of the stochastic model of the noise-free SML case (2.13) with respect to the DoAs $\boldsymbol{\theta}$.

Proof. See [13]. ■

Again, we show that in the SML case the number of identifiable sources with subarray sampling is upper bounded by number of sensors of the full array under the same conditions as in Theorem 2.

Theorem 4 (Necessary Identifiability Condition - SML)**.** *Assume that for any choice of M distinct DoAs $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_M]^{\mathrm{T}}$, the steering matrix of the fully sampled array $\boldsymbol{A}(\boldsymbol{\theta})$ has full rank. Then, the inequality*

$$L < M, \tag{2.29}$$

is a necessary condition for the identifiability of the stochastic model of the SML case (2.13) with respect to the DoAs $\boldsymbol{\theta}$.

Proof. We show that for $L \geq M$, there exists a $\boldsymbol{\theta}'$, which is different from $\boldsymbol{\theta}$, and some $\boldsymbol{C}_{\boldsymbol{s}}'$ and $\sigma_\eta'^2$, such that

$$\boldsymbol{A}^{(k)}(\boldsymbol{\theta})\boldsymbol{C}_{\boldsymbol{s}}\boldsymbol{A}^{(k),\mathrm{H}}(\boldsymbol{\theta})+\sigma_\eta^2\mathbf{I}_W = \boldsymbol{A}^{(k)}(\boldsymbol{\theta}')\boldsymbol{C}_{\boldsymbol{s}}'\boldsymbol{A}^{(k),\mathrm{H}}(\boldsymbol{\theta}')+\sigma_\eta'^2\mathbf{I}_W, \quad \forall k = 1, \ldots, K. \tag{2.30}$$

Let us choose $\sigma_\eta'^2 = \sigma_\eta^2$, then we see that (2.30) holds if

$$\boldsymbol{A}^{(k)}(\boldsymbol{\theta})\boldsymbol{C}_{\boldsymbol{s}}\boldsymbol{A}^{(k),\mathrm{H}}(\boldsymbol{\theta}) = \boldsymbol{A}^{(k)}(\boldsymbol{\theta}')\boldsymbol{C}_{\boldsymbol{s}}'\boldsymbol{A}^{(k),\mathrm{H}}(\boldsymbol{\theta}'), \quad \forall k = 1, \ldots, K. \tag{2.31}$$

Note that due to the assumption in Theorem 4, the matrix $\boldsymbol{A}(\boldsymbol{\theta}')$ has full row rank for any $\boldsymbol{\theta}' \in \mathbb{R}^L$ with $L \geq M$. Therefore, we can express any $\boldsymbol{A}(\boldsymbol{\theta})$ by

$$\boldsymbol{A}(\boldsymbol{\theta}) = \boldsymbol{A}(\boldsymbol{\theta}')\boldsymbol{X}, \tag{2.32}$$

for every $\boldsymbol{\theta} \in \mathbb{R}^L$. Hence, for every $\boldsymbol{\theta}$ there exist some $\boldsymbol{\theta}'$ and $\boldsymbol{X}$ such that

$$\boldsymbol{A}^{(k)}(\boldsymbol{\theta}) = \boldsymbol{G}^{(k)}\boldsymbol{A}(\boldsymbol{\theta}) = \boldsymbol{G}^{(k)}\boldsymbol{A}(\boldsymbol{\theta}')\boldsymbol{X} = \boldsymbol{A}^{(k)}(\boldsymbol{\theta}')\boldsymbol{X}, \forall k = 1, \ldots, K. \tag{2.33}$$

Inserting (2.33) into (2.31), we obtain

$$\begin{aligned}\boldsymbol{A}^{(k)}(\boldsymbol{\theta})\boldsymbol{C}_{\boldsymbol{s}}\boldsymbol{A}^{(k),\mathrm{H}}(\boldsymbol{\theta}) &= \boldsymbol{A}^{(k)}(\boldsymbol{\theta}')\boldsymbol{X}\boldsymbol{C}_{\boldsymbol{s}}\boldsymbol{X}^{\mathrm{H}}\boldsymbol{A}^{(k),\mathrm{H}}(\boldsymbol{\theta}') \\ &= \boldsymbol{A}^{(k)}(\boldsymbol{\theta}')\boldsymbol{C}_{\boldsymbol{s}}'\boldsymbol{A}^{(k),\mathrm{H}}(\boldsymbol{\theta}'), \quad \forall k = 1, \ldots, K,\end{aligned} \tag{2.34}$$

i.e., for every $\boldsymbol{\theta}$ and $\boldsymbol{C}_{\boldsymbol{s}}$ there exists a $\boldsymbol{\theta}', \boldsymbol{\theta}' \neq \boldsymbol{\theta}$, and some positive semidefinite $\boldsymbol{C}_{\boldsymbol{s}}' = \boldsymbol{X}\boldsymbol{C}_{\boldsymbol{s}}\boldsymbol{X}^{\mathrm{H}}$, which lead to the same covariance, and therefore, the PDF (2.13) is not unique in $\boldsymbol{\theta}$ for $L \geq M$. ■

Note that this necessary condition considers the case of a full transmit covariance matrix $\boldsymbol{C}_{\boldsymbol{s}}$. If the transmit signals are known to be uncorrelated a priori, the bound on the number of sources can be relaxed. In that case, a similar necessary condition that limits L to be smaller than KW can be obtained by using the vectorized system model (2.27) and demanding linear independence for every KW columns of the co-array manifold $\breve{\boldsymbol{V}}(\boldsymbol{\theta})$. The proof then follows the proof of Theorem 2. This relaxed bound plays a role, when sparse linear arrays are considered, where also in the fully sampled case M sources or more can be estimated (see, e.g., [20, 21, 22]).

2.3 Cramér-Rao-Bound

Although the previous section provides a condition, which guarantees that an estimation of the underlying parameters of the stochastic model is possible, it makes no statement about the achievable estimation accuracy. For a theoretical assessment of the performance that we can expect from an estimator, we take a look at the CRB [23, 24] corresponding to the different stochastic models. The CRB is a lower bound for the variance of any unbiased estimator. It is given by the inverse of the so called Fisher information matrix $\boldsymbol{F}$, i.e.,

$$\boldsymbol{C}_{\hat{\boldsymbol{\varphi}}} \succeq \boldsymbol{F}^{-1}, \tag{2.35}$$

where $\boldsymbol{C}_{\hat{\boldsymbol{\varphi}}}$ denotes the covariance matrix of the estimate of the parameter vector $\boldsymbol{\varphi}$. Thereby, the Fisher information matrix $\boldsymbol{F}$ is defined as

$$\boldsymbol{F}(i,j) = \mathrm{E}\left[\frac{\partial \ln(p(\boldsymbol{x};\boldsymbol{\varphi}'))}{\partial \varphi_i'}\Big|_{\boldsymbol{\varphi}'=\boldsymbol{\varphi}} \frac{\partial \ln(p(\boldsymbol{x};\boldsymbol{\varphi}'))}{\partial \varphi_j'}\Big|_{\boldsymbol{\varphi}'=\boldsymbol{\varphi}} \right]. \tag{2.36}$$

2.3.1 DML CRB

In the context of the deterministic DoA estimation problem presented in the previous section, the CRB can be simplified. We employ the Slepian-Bangs formula [25] given by

$$\begin{aligned} \boldsymbol{F}(i,j) = 2\Re &\left\{ \frac{\partial \boldsymbol{\mu}^{\mathrm{H}}(\boldsymbol{\varphi}')}{\partial \varphi_i'}\Big|_{\boldsymbol{\varphi}'=\boldsymbol{\varphi}} \boldsymbol{C}_{\boldsymbol{x}}(\boldsymbol{\varphi}) \frac{\partial \boldsymbol{\mu}(\boldsymbol{\varphi}')}{\partial \varphi_j'}\Big|_{\boldsymbol{\varphi}'=\boldsymbol{\varphi}} \right\} \\ &+ \mathrm{tr}\left\{ \boldsymbol{C}_{\boldsymbol{x}}^{-1}(\boldsymbol{\varphi}) \frac{\partial \boldsymbol{C}_{\boldsymbol{x}}(\boldsymbol{\varphi}')}{\partial \varphi_i'}\Big|_{\boldsymbol{\varphi}'=\boldsymbol{\varphi}} \boldsymbol{C}_{\boldsymbol{x}}^{-1}(\boldsymbol{\varphi}) \frac{\partial \boldsymbol{C}_{\boldsymbol{x}}(\boldsymbol{\varphi}')}{\partial \varphi_j'}\Big|_{\boldsymbol{\varphi}'=\boldsymbol{\varphi}} \right\} \end{aligned} \tag{2.37}$$

that is applicable for any circularly-symmetric complex Gaussian distributed observations $\boldsymbol{x} \sim \mathcal{CN}\left(\boldsymbol{\mu}(\boldsymbol{\varphi}), \boldsymbol{C}_{\boldsymbol{x}}(\boldsymbol{\varphi})\right)$, to obtain the elements of $\boldsymbol{F}$. For the DML system model, this formula can be used with the observations $\boldsymbol{x} = \tilde{\boldsymbol{Y}}$ and parameter vector $\boldsymbol{\varphi} = [\boldsymbol{\theta}^{\mathrm{T}}, \mathrm{vec}\{\boldsymbol{S}^{(1)}\}^{\mathrm{T}}, \dots, \mathrm{vec}\{\boldsymbol{S}^{(K)}\}^{\mathrm{T}}, \sigma_\eta^2]^{\mathrm{T}}$ to compute the respective Fisher information matrix. From there, we obtain the CRB for the parameters of interest $\boldsymbol{\theta}$ by simply selecting the upper left $L \times L$ submatrix of $\boldsymbol{F}^{-1}$, i.e.,

$$\boldsymbol{C}_{\text{CRB}} = [\mathbf{e}_1, \dots, \mathbf{e}_L]^{\mathrm{T}} \boldsymbol{F}^{-1} [\mathbf{e}_1, \dots, \mathbf{e}_L]. \tag{2.38}$$

2.3.2 SML CRB

The CRB for the SML case has been derived in [10]. A more compact form of the CRB for the parameters of interest $\boldsymbol{\theta}$ can be obtained by utilizing the Schur complement (as in [13]). Then, the CRB reads as

$$\boldsymbol{C}_{\text{CRB}} = \left[\boldsymbol{\Delta}_1^{\mathrm{H}} \left(\tilde{\boldsymbol{R}} - \tilde{\boldsymbol{R}}\boldsymbol{\Delta}_2 \left(\boldsymbol{\Delta}_2^{\mathrm{H}} \tilde{\boldsymbol{R}} \boldsymbol{\Delta}_2\right)^{-1} \boldsymbol{\Delta}_2^{\mathrm{H}} \tilde{\boldsymbol{R}}\right) \boldsymbol{\Delta}_1\right]^{-1}, \tag{2.39}$$

with

$$\tilde{\boldsymbol{R}} = \mathrm{blockdiag}\left\{\boldsymbol{R}^{(1)}, \dots, \boldsymbol{R}^{(K)}\right\}, \tag{2.40}$$

$$\boldsymbol{R}^{(k)} = N(\boldsymbol{C}_{\boldsymbol{y}}^{(k),-\mathrm{T}} \otimes \boldsymbol{C}_{\boldsymbol{y}}^{(k),-1}), \tag{2.41}$$

$$\boldsymbol{\Delta}_1 = \frac{\partial \boldsymbol{c}_{\boldsymbol{y}}}{\partial \boldsymbol{\theta}^{\mathrm{T}}}, \tag{2.42}$$

$$\boldsymbol{\Delta}_2 = \left[\frac{\partial \boldsymbol{c}_{\boldsymbol{y}}}{\partial \boldsymbol{\varphi}_{\boldsymbol{s}}^{\mathrm{T}}}, \frac{\partial \boldsymbol{c}_{\boldsymbol{y}}}{\partial \sigma_\eta^2}\right], \tag{2.43}$$

where $\boldsymbol{\varphi}_{\boldsymbol{s}}$ gathers the L^2 free real parameters in $\boldsymbol{C}_{\boldsymbol{s}}$.

Interestingly, for $L \geq W$, the CRB does not go to zero if the SNR goes to infinity, in contrast to the fully sampled case. This phenomenon has been thoroughly investigated in [26].

Classical Estimation Methods 3

3.1 Maximum Likelihood Estimation

The ML principle is probably the most prominent concept in parameter estimation and has been thoroughly investigated in the context of DoA estimation (e.g. [1]). In general, the ML estimator selects the parameters $\boldsymbol{\varphi}$ that maximize the likelihood function

$$p(\boldsymbol{x};\boldsymbol{\varphi}), \tag{3.1}$$

i.e., the PDF parameterized by $\boldsymbol{\varphi}$ at the observation $\boldsymbol{x}$. In the context of DoA estimation, the distribution of interest is the PDF of the received signals, which is evaluated at the observation $\tilde{\boldsymbol{Y}}$. However, depending on the statistical properties of the transmit signals $\tilde{\boldsymbol{S}}$, we obtain two different likelihood functions for our system model in (2.4), which in turn leads to two different ML estimators, namely the deterministic and the stochastic ML estimator.

3.1.1 Deterministic Maximum Likelihood Estimation

For the DML estimator, we assume that the received signal is distributed according to (2.10). From there, it follows that the likelihood function for the DML case is given by

$$p_{\mathrm{DML}}(\tilde{\boldsymbol{Y}};\boldsymbol{\theta},\tilde{\boldsymbol{S}},\sigma_\eta^2) = \left(\frac{1}{\pi\sigma_\eta^2}\right)^{KNW} \prod_{n=1}^{N} \mathrm{e}^{-\|\tilde{\boldsymbol{y}}(n)-\tilde{\boldsymbol{A}}(\boldsymbol{\theta})\tilde{\boldsymbol{s}}(n)\|^2/\sigma_\eta^2}. \tag{3.2}$$

Instead of maximizing the likelihood function, we can maximize its logarithm instead, which in our case reads as

$$\ln\left(p_{\mathrm{DML}}(\tilde{\boldsymbol{Y}};\boldsymbol{\theta},\tilde{\boldsymbol{S}},\sigma_\eta^2)\right) = -KNW\ln\left(\pi\sigma_\eta^2\right) - \frac{1}{\sigma_\eta^2}\sum_{n=1}^{N}\|\tilde{\boldsymbol{y}}(n)-\tilde{\boldsymbol{A}}(\boldsymbol{\theta})\tilde{\boldsymbol{s}}(n)\|^2. \tag{3.3}$$

For fixed angles $\boldsymbol{\theta}$ and $W > L$, we can give closed form maxima of (3.3) for σ_η^2 and $\tilde{\boldsymbol{s}}(n)$ by

$$\tilde{\boldsymbol{s}}(n) = \tilde{\boldsymbol{A}}^{+}(\boldsymbol{\theta})\tilde{\boldsymbol{y}}(n), \tag{3.4}$$

and

$$\hat{\sigma}_{\eta}^{2} = \frac{1}{KW} \operatorname{tr}\left\{\mathbf{P}^{\perp}_{\tilde{\boldsymbol{A}}(\boldsymbol{\theta})} \hat{\boldsymbol{C}}_{\tilde{\boldsymbol{y}}}\right\}, \tag{3.5}$$

where the sample covariance matrix of the received signals is given by

$$\hat{\boldsymbol{C}}_{\tilde{\boldsymbol{y}}} = \operatorname{blockdiag}\{\hat{\boldsymbol{C}}_{\boldsymbol{y}}^{(1)}, \ldots, \hat{\boldsymbol{C}}_{\boldsymbol{y}}^{(K)}\}, \tag{3.6}$$

$$\hat{\boldsymbol{C}}_{\boldsymbol{y}}^{(k)} = \frac{1}{N} \sum_{n=1}^{N} \tilde{\boldsymbol{y}}^{(k)}(n) \tilde{\boldsymbol{y}}^{(k),\mathrm{H}}(n). \tag{3.7}$$

With the results from (3.4) and (3.5), we can eliminate some of the parameters in (3.3) to obtain the concentrated log-likelihood function

$$\ln\left(p_{\mathrm{DML}}(\tilde{\boldsymbol{Y}}; \boldsymbol{\theta})\right) = -KNW \ln(\pi) - KW - \ln\left(\operatorname{tr}\left\{\mathbf{P}^{\perp}_{\tilde{\boldsymbol{A}}(\boldsymbol{\theta})} \hat{\boldsymbol{C}}_{\tilde{\boldsymbol{y}}}\right\}\right). \tag{3.8}$$

Finally, from (3.8), the DML estimate $\hat{\boldsymbol{\theta}}_{\mathrm{DML}}$ for $W > L$ can be obtained by

$$\hat{\boldsymbol{\theta}}_{\mathrm{DML}} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \operatorname{tr}\left\{\mathbf{P}^{\perp}_{\tilde{\boldsymbol{A}}(\boldsymbol{\theta})} \hat{\boldsymbol{C}}_{\tilde{\boldsymbol{y}}}\right\}. \tag{3.9}$$

The optimization problem (3.9) is still non-convex, however, we can find the global optimum by employing a two stage optimization. First, we perform a grid search in $\boldsymbol{\theta}$. To that end, we evaluate the log-likelihood function at a finite amount of grid points. Then, the grid point with the largest log-likelihood is used as an initialization for a gradient approach. If the initialization was close enough to the optimal $\boldsymbol{\theta}$, the gradient approach converges to the global optimum of (3.9). Note that, as the domain of $\boldsymbol{\theta}$ is bounded, the distance between two grid points can become arbitrarily small by increasing the number of grid points. Therefore, an initialization of the gradient steps close to the optimum can be guaranteed by a sufficiently large number of grid points. Unfortunately, the complexity of the grid search grows linearly with the number of grid points, which itself grows exponentially with the number of sources. Hence, there is a trade-off between accuracy and computational complexity when we are evaluating the DML estimator in this way. This trade-off is discussed in more detail in Section 3.1.3.

For $L \geq W$, we cannot reduce the maximum likelihood estimation to a search in $\boldsymbol{\theta}$. Also note that in this case, we cannot guarantee that the DML optimization problem has a unique solution, as has been discussed in Section 2.2.1.

3.1.2 Stochastic Maximum Likelihood Estimation

As mentioned previously, we assume that the transmit signals follow a Gaussian distribution in the SML case, such that the received signals are distributed according to (2.13). The respective log-likelihood function reads as

$$\ln\left(p_{\mathrm{SML}}\left(\tilde{\boldsymbol{Y}};\boldsymbol{\theta},\boldsymbol{C}_{\boldsymbol{s}},\sigma_\eta^2\right)\right) = -N\left(\ln\left(\det\left\{\boldsymbol{C}_{\tilde{\boldsymbol{y}}}\right\}\right) + \mathrm{tr}\left\{\boldsymbol{C}_{\tilde{\boldsymbol{y}}}^{-1}\hat{\boldsymbol{C}}_{\tilde{\boldsymbol{y}}}\right\} + W\ln(\pi)\right), \tag{3.10}$$

with

$$\boldsymbol{C}_{\tilde{\boldsymbol{y}}} = \tilde{\boldsymbol{A}}(\boldsymbol{\theta})\left(\mathbf{I}\otimes\boldsymbol{C}_{\boldsymbol{s}}\right)\tilde{\boldsymbol{A}}^{\mathrm{H}}(\boldsymbol{\theta}) + \sigma_\eta^2\mathbf{I}_{KW}. \tag{3.11}$$

Unfortunately, we cannot find closed form maximizers of (3.10) for $\boldsymbol{C}_{\boldsymbol{s}}$ and σ_η^2 if $\boldsymbol{\theta}$ is fixed (not even for $L < W$), in contrast to the DML case. Hence, evaluating the SML estimator requires a multidimensional search in $L^2 + L + 1$ real parameters to find the maximum of the non-concave log-likelihood function. In practice, the associated computational load becomes intractable for any $L \geq 2$, which might at first raise the question of whether the SML model is of any practical relevance. However, based on the SML model multiple estimators with lower complexity can be derived, and for the special case of uncorrelated transmit signals, there exists a sufficient condition for the number of identifiable DoAs for $L \geq W$, as has been discussed in Section 2.2.2.

3.1.3 MSE Approximation for the Grid-Based DML Estimators

In this section, we provide a method for an efficient characterization of the achievable estimation accuracy of grid-based DML estimators[1]. The achievable performance of ML estimators falls in three different regions [28]. The most prominent of those regions is the asymptotic region, at high SNR, where the ML estimator attains the CRB. For very low SNR, the received signals are dominated by noise such that it becomes practically impossible to estimate the actual parameters, which is why this region is called the no-information region. Lastly, the threshold region comprises the SNR range in between the aforementioned regions, where we have a mixture of meaningful estimates, which lie close to the actual parameters, and some inaccurate estimates from observations that are dominated by noise. Ideally, when designing a direction finder for a real world application, we want to ensure that the estimates are reliable, i.e., an operation in the asymptotic region is desirable. To that end, the achievable performance of the estimation algorithm needs to be assessed before a deployment. One possible way to do this is by means of Monte Carlo simulations. However, to achieve meaningful results, the necessary number of Monte Carlo runs, and therefore, the computational complexity might be very high. Alternatively, lower bounds based on the underlying stochastic model can be evaluated. Apart from the

[1] The presented approach is a direct extension from the fully sampled case, which has been discussed in [27], and therefore, the presentation of this topic closely follows the discussion in [27].

well known CRB, which is only tight in the asymptotic region, multiple other lower bounds (e.g. Barankin-, Ziv-Zakai-, or Weiss-Weinstein-type bounds [29, 30, 31, 32]) have been proposed in the previous literature. Compared to the CRB, these lower bounds are tighter at low SNR, but they are still not able to accurately predict the ML performance in the threshold region.

Therefore, we present a way to obtain an approximation of the achievable MSE of a grid-based DML DoA estimator, which provides a more accurate assessment of the achievable performance compared to such stochastic lower bounds and has a much lower complexity compared to extensive Monte Carlo simulations. Inital work on this topic goes back to [28], where Athley proposed such an MSE approximation that is able to accurately predict the MSE for an ML estimator in the asymptotic region and its transition to the threshold region, for DoA estimation with gridless ML estimators and fully sampled antenna arrays. Moreover, the approximation provides a more accurate prediction of the achievable MSE than the Barankin bound. In contrast to [28], we investigate a grid-based estimator that works on a finite grid for the DoAs, which is relevant in two cases. First, scenarios where the array manifold is only known for a finite number of DoAs. Such a scenario occurs if the array manifold cannot be properly modeled by an analytic function, but is only available via measurement data, for example, from calibration measurements of an array installation in the field [33, Chapter 3]. Second, grid-based ML estimators can be used as an initialization for gridless ML estimators as we have discussed in the previous sections. The grid-based initialization should yield a point in the neighborhood of the global maximizer of the likelihood function. Otherwise, the subsequent gradient steps converge to a local maximum instead of the global maximum. Hence, outliers in the grid-based estimation lead to outliers for the gridless ML estimator. Naturally, this kind of error propagation can be eliminated if we use a very dense grid for the initialization. However, as we have already discussed, using very dense grids leads to a prohibitive complexity, due to its exponential growth in the number of sources L. This leads to a trade-off between computational complexity and robustness against outliers. Especially for finding the optimal number of grid points in regard to this trade-off, the presented MSE approximation shines.

3.1.3.1 Preliminaries

We consider the DML model for the received signals given in (2.10). The respective optimization task for the gridless DML estimator is shown in (3.9). To obtain the grid-based DML estimator, we evaluate the gridless estimator only on a finite, multidimensional grid of DoAs $\mathcal{T} = \{\check{\boldsymbol{\theta}}_1, \check{\boldsymbol{\theta}}_2, \ldots, \check{\boldsymbol{\theta}}_T\}$. The respective optimization problem then reads as

$$\begin{aligned}\hat{\boldsymbol{\theta}} &= \underset{\check{\boldsymbol{\theta}}\in\mathcal{T}}{\operatorname{argmax}}\ U(\check{\boldsymbol{\theta}}) \\ &= \underset{\check{\boldsymbol{\theta}}\in\mathcal{T}}{\operatorname{argmax}}\ \operatorname{tr}\left\{\mathbf{P}_{\tilde{\boldsymbol{A}}(\boldsymbol{\theta})}\hat{\boldsymbol{C}}_{\tilde{\boldsymbol{y}}}\right\}.\end{aligned} \tag{3.12}$$

Although we search for the best $\check{\boldsymbol{\theta}}$ on a finite grid, the true DoAs $\boldsymbol{\theta}^\star$ of our sources are not restricted to lie on the grid $\mathcal{T}$. Hence, the MSE of the grid-based estimator is given by

$$\mathrm{MSE} = \sum_{t=1}^{T} P_t \|\check{\boldsymbol{\theta}}_t - \boldsymbol{\theta}^\star\|^2, \tag{3.13}$$

where P_t denotes the probability that $\check{\boldsymbol{\theta}}_t$ is the maximizer of (3.12), i.e., the DML estimate. Naturally, this probability depends on the SNR, the signal realizations $\tilde{\boldsymbol{S}}$ and true DoAs $\boldsymbol{\theta}^\star$. Following the lines of [28], we decompose the MSE expression into two parts. The first part captures the local errors from estimates that lie in the vicinity of the actual DoAs $\boldsymbol{\theta}^\star$. The second part comprises the errors caused by outliers. Under outliers we understand estimates that are far away from the true DoAs $\boldsymbol{\theta}^\star$. This decomposition leads to the MSE expression

$$\mathrm{MSE} = \left(1 - \sum_i P_i^{\mathrm{out}}\right)\varepsilon^{\mathrm{loc}} + \sum_i P_i^{\mathrm{out}}\varepsilon_i^{\mathrm{out}}, \tag{3.14}$$

with the local error $\varepsilon^{\mathrm{loc}}$, and the probability and error of the i-th outlier, which are denoted by P_i^{out} and $\varepsilon_i^{\mathrm{out}}$, respectively.

In the following, we will show that for the MSE of the grid-based DML estimator this decomposition can be written as

$$\begin{aligned}\mathrm{E}\left[\|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^\star\|^2\right] = &\left(1 - \sum_{i=1}^{|\mathcal{S}|} P_i^{\mathrm{out}}\right)\sum_{\ell=1}^{|\mathcal{M}|} P_\ell^{\mathrm{loc}}\|\check{\boldsymbol{\theta}}_\ell^{\mathrm{main}} - \boldsymbol{\theta}^\star\|^2 \\ &+ \sum_{i=1}^{|\mathcal{S}|} P_i^{\mathrm{out}}\|\check{\boldsymbol{\theta}}_i^{\mathrm{side}} - \boldsymbol{\theta}^\star\|^2.\end{aligned} \tag{3.15}$$

3.1.3.2 Outlier Approximation

Let us first discuss the outlier approximation, which is the latter part of (3.15). As has been shown in [28], the distribution of gridless ML estimates is proportional to the respective asymptotic log-likelihood function $\bar{U}(\boldsymbol{\theta})$. This asymptotic log-likelihood is up to some constant terms equal to the concentrated log-likelihood function (3.8) for vanishing noise[2], i.e.,

$$\bar{U}(\boldsymbol{\theta}) = \operatorname{tr}\left\{\mathbf{P}_{\tilde{\boldsymbol{A}}(\boldsymbol{\theta})} \tilde{\boldsymbol{A}}(\boldsymbol{\theta}^{\star}) \left(\mathbf{I}_L \otimes \boldsymbol{C}_{\tilde{\boldsymbol{s}}}\right) \tilde{\boldsymbol{A}}^{\mathrm{H}}(\boldsymbol{\theta}^{\star})\right\}. \tag{3.16}$$

For a single source and a fully sampled array, this asymptotic log-likelihood becomes the well known beam pattern of the array at $\theta^{\star}$, i.e.,

$$\bar{U}(\theta) = \frac{|\boldsymbol{a}^{\mathrm{H}}(\theta)\boldsymbol{a}(\theta^{\star})|}{\|\boldsymbol{a}(\theta)\|_2 \|\boldsymbol{a}(\theta^{\star})\|_2}. \tag{3.17}$$

In that sense, we can think of $\bar{U}(\boldsymbol{\theta})$ as an extension of the beam pattern to subarray sampling and multiple sources. In general, $\bar{U}(\boldsymbol{\theta})$ has not only a global maximum at $\boldsymbol{\theta}^{\star}$, but also multiple local maxima. We refer to the area around the global maximum as mainlobe, whereas the area around the local maxima are called sidelobes[3].

From the proportionality between the distribution of the gridless ML estimates and $\bar{U}(\boldsymbol{\theta})$, it is clear that it is much more likely to obtain estimates near the sidelobe peaks than in the valleys of the asymptotic log-likelihood function. For estimates near the sidelobe peaks, we can approximate their induced error by the distance between the closest sidelobe peak and $\boldsymbol{\theta}^{\star}$, i.e., the for an estimate in the i-th sidelobe the error can be reasonably well approximated by $\|\boldsymbol{\theta}^{\star} - \boldsymbol{\theta}_i^{\text{side}}\|^2$, where $\boldsymbol{\theta}_i^{\text{side}}$ is the i-th sidelobe peak. Furthermore, to compute the probability of obtaining an ML estimate in the i-th sidelobe P_i^{out} a discretization of $U(\boldsymbol{\theta})$ at the sidelobe peaks is proposed in [28], i.e., P_i^{out} is the probability that $U(\boldsymbol{\theta})$ is larger at $\boldsymbol{\theta} = \boldsymbol{\theta}_i^{\text{side}}$ than at the mainlobe peak and any other sidelobe peak. With the help of the union bound, this probability can be further approximated by $\Pr[U(\boldsymbol{\theta}_i^{\text{side}}) > U(\boldsymbol{\theta}^{\star})]$, as it is unlikely that $U(\boldsymbol{\theta})$ is larger than $U(\boldsymbol{\theta}^{\star})$ at multiple sidelobe peaks simultaneously [28].

As for grid-based estimators, the log-likelihood function is sampled only at discrete grid points $\check{\boldsymbol{\theta}} \in \mathcal{T}$, in general, there are no grid points coinciding with the true DoAs $\boldsymbol{\theta}^{\star}$ and the sidelobe peaks. In fact, there might not even be a grid point lying in a specific sidelobe if the grid is very coarse and the respective sidelobe is narrow. Hence, we no longer use the sidelobe peaks $\boldsymbol{\theta}_i^{\text{side}}$ and the true DoAs $\boldsymbol{\theta}^{\star}$ to compute the

[2] Since the SML estimator does not offer a concentrated formulation of the log-likelihood, which no longer depends on the noise power and transmit covariance, a straight-forward extension of the presented approximation to the SML case is not possible.

[3] The borders between the mainlobe and the individual sidelobes are the valleys in the function $\bar{U}(\boldsymbol{\theta})$. This means that from any point in these lobes, a gradient ascent approach with a sufficiently small step size converges to the respective local maximum.

outlier probabilities, but we select one grid point as a representative for the mainlobe $\check{\boldsymbol{\theta}}_0$ and for each sidelobe $\check{\boldsymbol{\theta}}_i^{\text{side}}$. As representatives, we choose the grid point with the highest asymptotic log-likelihood $\bar{U}(\boldsymbol{\theta})$ among all grid points that belong to the respective lobe. Again, we follow the lines of [28] and approximate the probability of obtaining an estimate in the i-th sidelobe P_i^{out} by comparing the likelihood function at the representatives, i.e.,

$$P_i^{\text{out}} = \Pr\left[\bigcap_{\substack{j\in\mathcal{S}\\ j\neq i}} \left\{U(\check{\boldsymbol{\theta}}_i^{\text{side}}) > U(\check{\boldsymbol{\theta}}_j^{\text{side}})\right\} \cap \left\{U(\check{\boldsymbol{\theta}}_i^{\text{side}}) > U(\check{\boldsymbol{\theta}}_0)\right\}\right], \tag{3.18}$$

where the set $\mathcal{S}$ gathers all the sidelobe representatives. Applying the union bound to the complementary event and keeping only the dominant term we finally obtain

$$P_i^{\text{out}} \approx \Pr\left[U(\check{\boldsymbol{\theta}}_i^{\text{side}}) > U(\check{\boldsymbol{\theta}}_0)\right]. \tag{3.19}$$

The probability in (3.19) is the pairwise error probability between the i-th sidelobe representative $\check{\boldsymbol{\theta}}_i^{\text{side}}$ and the mainlobe representative $\check{\boldsymbol{\theta}}_0$.

In contrast to the fully sampled case [27, 28], for systems with subarray sampling, we cannot find an analytic expression for the pairwise error probabilities (3.19) by applying the results of [34, Appendix B]. However, we can employ the formulation for the multi source case from [27, 28] to evaluate the probabilities in (3.19).

$$\Pr\left[U(\check{\boldsymbol{\theta}}_i^{\text{side}}) > U(\check{\boldsymbol{\theta}}_0)\right] = \Pr\left[\tilde{\boldsymbol{y}}_N^{\text{H}}\boldsymbol{\Psi}_i\tilde{\boldsymbol{y}}_N > 0\right], \tag{3.20}$$

with the stacked received signals $\tilde{\boldsymbol{y}}_N = [\tilde{\boldsymbol{y}}^{\text{H}}(1), \ldots, \tilde{\boldsymbol{y}}^{\text{H}}(N)]^{\text{H}}$ and

$$\boldsymbol{\Psi}_i = \mathbf{I}_N \otimes (\mathbf{P}_{\tilde{\boldsymbol{A}}(\check{\boldsymbol{\theta}}_i^{\text{side}})} - \mathbf{P}_{\tilde{\boldsymbol{A}}(\check{\boldsymbol{\theta}}_0)}). \tag{3.21}$$

We can observe that the outlier probabilities depend on the distribution of a non-central quadratic form with the indefinite matrix $\boldsymbol{\Psi}_i$. Evaluating such a probability is non-trivial. We resort to the solution proposed by Athley in [28] that uses a saddlepoint approximation [35] to approximate the outlier probability (3.20). Using the cumulant generating function

$$\kappa(s) = \ln\left(M(s)\right) = \ln\left(\text{E}\left[\text{e}^{sq}\right]\right), \tag{3.22}$$

we can approximate the cumulative distribution of a quadratic form q by

$$\Pr\left[q < x\right] \approx \Phi\left(w + \frac{1}{w}\ln\left(\frac{v}{w}\right)\right), \tag{3.23}$$

with

$$w = \operatorname{sgn}(s_0)\sqrt{2(s_0 x - \kappa(s_0))}, \tag{3.24}$$

$$v = s_0 \sqrt{\left.\frac{\mathrm{d}^2}{\mathrm{d}^2 s}\kappa(s)\right|_{s=s_0}}, \tag{3.25}$$

and the saddle point s_0 defined by the unique root of

$$\frac{\mathrm{d}}{\mathrm{d}s}\kappa(s) = x, \tag{3.26}$$

that lies in the convergence region of the moment generating function $M(s)$ [36].

Following the lines of [28], the cumulant generating function for the quadratic form in (3.21) can be obtained by

$$\kappa(s) = \sum_i \frac{\mu_i^2 \zeta_i s}{1 - \zeta_i s} - N \sum_i \ln\left(1 - \zeta_i s\right), \tag{3.27}$$

with the non-zero eigenvalues ζ_i from the eigenvalue decomposition

$$\sigma_\eta^2 \left(\mathbf{P}_{\tilde{\boldsymbol{A}}(\check{\boldsymbol{\theta}}_o^{\text{side}})} - \mathbf{P}_{\tilde{\boldsymbol{A}}(\check{\boldsymbol{\theta}}_0)}\right) = \boldsymbol{Q}\boldsymbol{Z}\boldsymbol{Q}^{\mathrm{H}} = \sum_{i=1}^{KW} \zeta_i \boldsymbol{q}_i \boldsymbol{q}_i^{\mathrm{H}}, \tag{3.28}$$

and

$$\mu_i^2 = \frac{1}{\sigma_\eta^2} \sum_{n=1}^{N} \left| \boldsymbol{q}_i \tilde{\boldsymbol{A}}(\boldsymbol{\theta}^\star)\tilde{\boldsymbol{s}}(n) \right|_2^2. \tag{3.29}$$

The saddle point s_0 lies in the interval

$$\frac{1}{\min(\boldsymbol{Z})} < s_0 < \frac{1}{\max(\boldsymbol{Z})}, \tag{3.30}$$

to guarantee convergence of the moment generating function [28].

3.1.3.3 Local Error Approximation

Now, let us discuss the local error part in (3.15). It is well known that the gridless ML estimator is asymptotically efficient [37], i.e., it achieves the CRB for many snapshots and high SNR. In this region, all the errors can be explained by the noise properties and the local curvature of the log-likelihood function at the true DoAs $\boldsymbol{\theta}^\star$. Therefore, it is proposed in [28] to use the CRB to approximate local error contributions in the gridless case.

For $L < W$, the CRB decreases and approaches zero for increasing SNR [26]. However, the MSE of a grid-based DML estimator usually saturates for high SNR, as the minimum achievable error is determined by the distance between $\boldsymbol{\theta}^\star$ and the closest grid point. Therefore, in the grid-based case, the CRB is no longer a suitable local error approximation, but a more sophisticated method needs to be employed.

As the error contribution of the estimates that lie in the sidelobes have already been accounted for by the outlier approximation, we can focus on the mainlobe for the local error performance of the grid-based ML estimator. Hence, we can express the local error part of our MSE approximation by the sum of the errors induced by each mainlobe grid point with the respective probability to obtain this grid point as the ML estimate. This gives the local error

$$\varepsilon^{\text{loc}} = \sum_{\ell=1}^{|\mathcal{M}|} P_\ell^{\text{loc}} \|\check{\boldsymbol{\theta}}_\ell^{\text{main}} - \boldsymbol{\theta}^\star\|^2, \tag{3.31}$$

where the set $\mathcal{M}$ gathers all the grid points $\check{\boldsymbol{\theta}}_\ell^{\text{main}}, \ell = 1, \dots, |\mathcal{M}|$, that are located in the mainlobe and P_ℓ^{loc} is the probability to obtain the ℓ-th mainlobe grid point. In the following, we will explain how to determine P_ℓ^{loc} depending on the dimensionality of $\boldsymbol{\theta}$.

Univariate Case: First we consider, that we have a single source, whose DoA can be described by a scalar θ. Again under the assumption that no outlier occurs, the DML estimator selects the ℓ-th mainlobe grid point if

$$P_\ell^{\text{loc}} = \Pr\left[\bigcap_{\substack{j\in\mathcal{M}\\ j\neq\ell}} \left\{U(\check{\theta}_\ell^{\text{main}}) > U(\check{\theta}_j^{\text{main}})\right\}\right]. \tag{3.32}$$

Let us assume that $U(\theta)$ is quasi-concave in the region of the mainlobe of the asymptotic likelihood $\bar{U}(\theta)$. Furthermore, the grid points $\check{\theta}_i^{\text{main}}$ are sorted in ascending order. Then, it directly follows from the quasi-concavity of $U(\theta)$ that

$$U(\check{\theta}_\ell^{\text{main}}) > U(\check{\theta}_{\ell-1}^{\text{main}}) \implies U(\check{\theta}_\ell^{\text{main}}) > U(\check{\theta}_j^{\text{main}}), \forall j < \ell. \tag{3.33}$$

Equivalently, it holds that

$$U(\check{\theta}_\ell^{\text{main}}) > U(\check{\theta}_{\ell+1}^{\text{main}}) \implies U(\check{\theta}_\ell^{\text{main}}) > U(\check{\theta}_j^{\text{main}}), \forall j > \ell. \tag{3.34}$$

Thus, the probability from equation (3.32) collapses to:

$$P_\ell^{\text{loc}} = \Pr\left[U(\check{\theta}_\ell^{\text{main}}) > U(\check{\theta}_{\ell-1}^{\text{main}}) \wedge U(\check{\theta}_\ell^{\text{main}}) > U(\check{\theta}_{\ell+1}^{\text{main}})\right], \tag{3.35}$$

and

$$\begin{aligned} P_\ell^{\text{loc}} =& 1 - \text{Pr}\left[U(\check{\theta}_\ell^{\text{main}}) \leq U(\check{\theta}_{\ell-1}^{\text{main}})\right] \\ & - \text{Pr}\left[U(\theta_\ell^{\text{main}}) \leq U(\check{\theta}_{\ell+1}^{\text{main}})\right] \\ & + \text{Pr}\left[U(\check{\theta}_\ell^{\text{main}}) \leq U(\check{\theta}_{\ell-1}^{\text{main}}) \wedge U(\check{\theta}_\ell^{\text{main}}) \leq U(\check{\theta}_{\ell+1}^{\text{main}})\right]. \end{aligned} \tag{3.36}$$

The last term in the formula vanishes due to the quasi-concavity of $U(\theta)$. The remaining probabilities of the events $U(\check{\theta}_i^{\text{main}}) \leq U(\check{\theta}_{i-1}^{\text{main}})$ and $U(\check{\theta}_i^{\text{main}}) \leq U(\check{\theta}_{i+1}^{\text{main}})$ can be calculated using the formulas for the grid-based pairwise error probabilities presented for the outlier approximation.

Note that even if the quasi-concavity assumption does not hold, the expression in (3.35) gives an upper bound to the actual probability to select the ℓ-th grid point in (3.32). Recall that it is very unlikely to obtain estimates in the valley of the asymptotic log-likelihood $\bar{U}(\theta)$. This implies that the maximum of $U(\theta)$ is mostly close to the true DoA $\boldsymbol{\theta}^\star$. In that case, the mainlobe width of $U(\theta)$ is approximately equal to the mainlobe width of $\bar{U}(\theta)$, and therefore, the quasi-concavity is only violated near the valley of $\bar{U}(\theta)$. Hence, the error introduced by using the upper bound in (3.35) affects mostly the mainlobe grid points at the edge of the mainlobe. However, as discussed, the probability to obtain one of these grid points is anyway very low such that the impact on the overall local error in (3.31) is pretty limited.

Multivariate Case: For multiple sources, we obtain a multivariate log-likelihood function $U(\boldsymbol{\theta})$. Under the assumption of independence of the individual elements of the estimate $\hat{\boldsymbol{\theta}}$ in the vicinity of the true DoAs, a straightforward extension of the univariate case is possible (cf. [27]). However, in general, this assumption does not hold. Therefore, a different method needs to be employed that is able to cope with correlated entries of $\hat{\boldsymbol{\theta}}$ [27]. To this end, we utilize that for the gridless case $\hat{\boldsymbol{\theta}} \sim \mathcal{N}(\boldsymbol{\theta}^\star, \boldsymbol{C}_{\text{CRB}})$ holds in the asymptotic region [37]. If we know which $\hat{\boldsymbol{\theta}}$ in the gridless case correspond to estimating the ℓ-th grid point in the grid-based case, then, we can determine the probability of obtaining the ℓ-th grid point by integrating the Gaussian density of $\hat{\boldsymbol{\theta}}$ over the respective area. This means that we numerically determine the probabilities for each mainlobe grid point based on the L dimensional density of $\hat{\boldsymbol{\theta}}$. In contrast, a direct evaluation of the probabilities by plain numerical integration would require an integration over the $2KWN$ dimensional density of the received signals $\tilde{\boldsymbol{Y}}$. However, note that we require a description of the array manifold by an analytic function to infer the mainlobe grid point probabilities with this method, as otherwise the CRB for corresponding gridless estimators cannot be derived.

Evaluating an arbitrary integral over a Gaussian PDF is not trivial. However, numerical integration over a Gaussian density can be performed quite efficiently if the integration region is a polyhedron [38]. Hence, our goal is to obtain a polyhedron around the ℓ-th grid point that approximates the area in which the gridless $\hat{\boldsymbol{\theta}}$ is mapped to the ℓ-th grid point $\check{\boldsymbol{\theta}}_\ell^{\text{main}}$. To this end, let us consider the second order Taylor approximation of $U(\boldsymbol{\theta})$ around $\hat{\boldsymbol{\theta}}$, which reads as

$$U(\boldsymbol{\theta}) \approx U(\hat{\boldsymbol{\theta}}) + \frac{1}{2}(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})^{\mathrm{T}} \boldsymbol{H}(\hat{\boldsymbol{\theta}})(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}), \tag{3.37}$$

where $\boldsymbol{H}(\hat{\boldsymbol{\theta}})$ is the Hessian matrix of $U(\boldsymbol{\theta})$ evaluated at $\hat{\boldsymbol{\theta}}$. Since the maximum of $U(\boldsymbol{\theta})$ is at the $\hat{\boldsymbol{\theta}}$, there is no first order term in (3.37).

Now, let us assume that

$$\boldsymbol{H}(\hat{\boldsymbol{\theta}}) \approx \boldsymbol{H}(\boldsymbol{\theta}^\star), \tag{3.38}$$

if $\hat{\boldsymbol{\theta}}$ is close to $\boldsymbol{\theta}^\star$. Then, by using (3.37) and (3.38), the event $U(\check{\boldsymbol{\theta}}_\ell) > U(\check{\boldsymbol{\theta}}_j)$ can be approximated by the following linear decision rule in $\hat{\boldsymbol{\theta}}$

$$2(\check{\boldsymbol{\theta}}_j - \check{\boldsymbol{\theta}}_\ell)^{\mathrm{T}} \boldsymbol{H}(\boldsymbol{\theta}^\star)\hat{\boldsymbol{\theta}} - \check{\boldsymbol{\theta}}_j^{\mathrm{T}} \boldsymbol{H}(\boldsymbol{\theta}^\star)\check{\boldsymbol{\theta}}_j + \check{\boldsymbol{\theta}}_\ell^{\mathrm{T}} \boldsymbol{H}(\boldsymbol{\theta}^\star)\check{\boldsymbol{\theta}}_\ell > 0. \tag{3.39}$$

This means that the event of obtaining the ℓ-th grid point by the grid-based DML estimator, given by the intersection $\bigcap_{j \neq \ell} U(\check{\boldsymbol{\theta}}_\ell) > U(\check{\boldsymbol{\theta}}_j)$, can be approximately represented by the intersection of linear decisions, i.e., a polyhedron. Therefore, the corresponding probability P_ℓ^{loc} can be computed numerically by the efficient framework of [38].

3.2 Generalized Least Squares Estimation

As we have seen in the previous chapter, the SML system model enables the DoA estimation for more sources L than RF chains W under certain conditions (cf. Theorem 3). However, the optimization problem corresponding to the SML estimator turned out to be computationally too demanding such that a direct evaluation of the SML estimator is intractable. Therefore, Sheinvald and Wax [10] developed the so called GLS estimator for systems with subarray sampling. In this chapter, we will revisit the original GLS estimator as presented in [10] and propose some augmentations that enable its usage even for a small number of samples N.

Instead of utilizing the likelihood function of the stochastic model to obtain the parameter estimates, the GLS estimator uses a covariance matching criterion. As the PDF (2.13) is fully parameterized by it covariance matrix, the idea is to find the parameters $\boldsymbol{\theta}, \boldsymbol{C}_{\boldsymbol{s}}$, and σ_η^2 which produce the best fit between the respective covariance matrix and the sample covariance matrix of the observed received signals. This rationale is also used for the class of Covariance Matching Estimation Technique (COMET) estimators discussed in [39]. For the GLS estimator, the proposed cost function for the covariance matching reads as [10]

$$\sum_{k=1}^{K} \|\boldsymbol{T}^{(k)} \left(\hat{\boldsymbol{C}}_{\boldsymbol{y}}^{(k)} - \boldsymbol{C}_{\boldsymbol{y}}^{(k)}\right) \boldsymbol{T}^{(k),\mathrm{H}}\|_{\mathrm{F}}^2, \tag{3.40}$$

where $\boldsymbol{T}^{(k)}$ is a whitening filter, for which the choice $\boldsymbol{T}^{(k)} = \hat{\boldsymbol{C}}_{\boldsymbol{y}}^{(k),-1/2}$ leads to an asymptotically consistent and efficient estimator [10].

For the minimization of (3.40), closed form solutions of $\boldsymbol{C}_{\boldsymbol{s}}$ and σ_η^2 can be found for fixed $\boldsymbol{\theta}$ by solving a simple least squares problem if the positive semidefiniteness constraint on $\boldsymbol{C}_{\boldsymbol{s}}$ and the fact that any feasible σ_η^2 is larger than zero is neglected [10]. Unfortunately, this approach yields infeasible estimates for $\boldsymbol{C}_{\boldsymbol{s}}$ and σ_η^2 if the number of snapshots is small. Furthermore, the resulting optimization problem in $\boldsymbol{\theta}$ remains non-convex. Therefore, we consider a GLS solution that accounts for these constraints [3].

The GLS optimization problem including the constraints reads as

$$\min_{\boldsymbol{\theta},\boldsymbol{C}_{\boldsymbol{s}}\succeq\boldsymbol{0},\sigma_\eta^2\geq 0} \sum_{k=1}^{K} \|\boldsymbol{T}^{(k)} \left(\hat{\boldsymbol{C}}_{\boldsymbol{y}}^{(k)} - \boldsymbol{A}^{(k)}(\boldsymbol{\theta})\boldsymbol{C}_{\boldsymbol{s}}\boldsymbol{A}^{(k),\mathrm{H}}(\boldsymbol{\theta}) - \sigma_\eta^2\mathbf{I}_W\right) \boldsymbol{T}^{(k),\mathrm{H}}\|_{\mathrm{F}}^2. \tag{3.41}$$

For fixed $\boldsymbol{\theta}$ this problem can be written as a Semidefinite Programming (SDP) [40, Ch. 4.6.2]. To that end let us first introduce the slack variable b to rewrite the problem for fixed $\boldsymbol{\theta}$ as

$$\begin{aligned}
&\min_{b,\boldsymbol{C}_{\boldsymbol{s}},\sigma_\eta^2} \ b \\
&\text{s. t.:}\, b > 0, \quad \boldsymbol{C}_{\boldsymbol{s}} \succeq \boldsymbol{0}, \quad \sigma_\eta^2 \geq 0, \\
&\qquad \|\boldsymbol{E}\|_F^2 \leq b^2, \\
&\qquad \boldsymbol{E} = \tilde{\boldsymbol{T}}\left(\hat{\boldsymbol{C}}_{\tilde{\boldsymbol{y}}} - \tilde{\boldsymbol{A}}(\boldsymbol{\theta})\left(\mathbf{I}_K \otimes \boldsymbol{C}_{\boldsymbol{s}}\right)\tilde{\boldsymbol{A}}^{\mathrm{H}}(\boldsymbol{\theta}) - \sigma_\eta^2\mathbf{I}_{KW}\right)\tilde{\boldsymbol{T}}^{\mathrm{H}},
\end{aligned} \tag{3.42}$$

with $\tilde{\boldsymbol{T}} = \mathrm{blockdiag}\{\boldsymbol{T}^{(1)},\ldots,\boldsymbol{T}^{(K)}\}$ and $\hat{\boldsymbol{C}}_{\tilde{\boldsymbol{y}}} = \mathrm{blockdiag}\{\hat{\boldsymbol{C}}_{\boldsymbol{y}}^{(1)},\ldots,\hat{\boldsymbol{C}}_{\boldsymbol{y}}^{(K)}\}$. Now, we can replace the Frobenius norm of $\boldsymbol{E}$ by the Euclidean norm of the vectorization of $\boldsymbol{E}$ to obtain

$$\|\boldsymbol{E}\|_F^2 \leq b^2 \iff \|\mathrm{vec}\{\boldsymbol{E}\}\|_2^2 \leq b^2 \iff \begin{bmatrix} b\mathbf{I}_{K^2W^2} & \mathrm{vec}\{\boldsymbol{E}\} \\ \mathrm{vec}\{\boldsymbol{E}\}^{\mathrm{H}} & b \end{bmatrix} \succeq \boldsymbol{0}, \tag{3.43}$$

where the final equivalence to the Linear Matrix Inequality (LMI) can be shown by the Schur complement, since the LMI only holds if and only if its respective Schur complement is positive semidefinite [40, A.5.5]. Hence, the optimization problem reads as

$$\begin{aligned}
&\min_{b, \boldsymbol{C}_{\boldsymbol{s}}, \sigma_\eta^2} \; b \\
&\text{s.t.:}\, b > 0, \quad \boldsymbol{C}_{\boldsymbol{s}} \succeq \mathbf{0}, \quad \sigma_\eta^2 \geq 0, \\
&\qquad \begin{bmatrix} b\mathbf{I}_{K^2W^2} & \mathrm{vec}\{\boldsymbol{E}\} \\ \mathrm{vec}\{\boldsymbol{E}\}^{\mathrm{H}} & b \end{bmatrix} \succeq \mathbf{0}, \\
&\qquad \boldsymbol{E} = \tilde{\boldsymbol{T}} \left(\hat{\boldsymbol{C}}_{\tilde{\boldsymbol{y}}} - \tilde{\boldsymbol{A}}(\boldsymbol{\theta}) \left(\mathbf{I}_K \otimes \boldsymbol{C}_{\boldsymbol{s}} \right) \tilde{\boldsymbol{A}}^{\mathrm{H}}(\boldsymbol{\theta}) - \sigma_\eta^2 \mathbf{I}_{KW} \right) \tilde{\boldsymbol{T}}^{\mathrm{H}},
\end{aligned} \tag{3.44}$$

which is a SDP, since $\mathrm{vec}\{\boldsymbol{E}\}$ is linear in the optimization variables.

In the special case of uncorrelated transmit signals, the optimization problem reduces to a non-negative least squares problem, i.e., a quadratic program. Using similar steps as for the reformulation of (3.40) to a least squares objective in [10], i.e., vectorizing the whitened covariance matrices and exploiting the diagonal structure of $\boldsymbol{C}_{\boldsymbol{s}}$ (cf. (2.27)), we can write the optimization problem (3.41) as

$$\min_{\boldsymbol{\lambda} \geq \mathbf{0}, \sigma_\eta^2 \geq 0} \left\| \boldsymbol{\xi} - \boldsymbol{\Gamma} \begin{bmatrix} \boldsymbol{\lambda} \\ \sigma_\eta^2 \end{bmatrix} \right\|_2^2, \tag{3.45}$$

with

$$\boldsymbol{\xi} = [\boldsymbol{\xi}^{(1),\mathrm{T}}, \ldots, \boldsymbol{\xi}^{(K),\mathrm{T}}]^{\mathrm{T}}, \tag{3.46}$$

$$\boldsymbol{\xi}^{(k)} = \mathrm{vec}\left(\boldsymbol{T}^{(k)} \hat{\boldsymbol{C}}_{\boldsymbol{y}}^{(k)} \boldsymbol{T}^{(k),\mathrm{H}} \right), \tag{3.47}$$

$$\boldsymbol{\Gamma} = [\boldsymbol{\Gamma}^{(1),\mathrm{T}}, \ldots, \boldsymbol{\Gamma}^{(K),\mathrm{T}}]^{\mathrm{T}}, \tag{3.48}$$

$$\boldsymbol{\Gamma}^{(k)} = \left[\left(\boldsymbol{T}^{(k),*} \boldsymbol{A}^{(k),*}(\boldsymbol{\theta}) \right) \circ \left(\boldsymbol{T}^{(k)} \boldsymbol{A}^{(k)}(\boldsymbol{\theta}) \right), \;\; \mathrm{vec}\left(\boldsymbol{T}^{(k)} \boldsymbol{T}^{(k),\mathrm{H}} \right) \right]. \tag{3.49}$$

With the optimization problems above, signal and noise estimates for fixed $\boldsymbol{\theta}$ can be obtained. Therefore, the solution of the non-convex optimization problem (3.41) reduces to a search in $\boldsymbol{\theta}$, which can again be solved by a grid search approach.

Sparse Recovery Methods 4

In this chapter, we study another class of estimators that are based on a sparse representation of the considered system model. Motivated by the theoretical results from the framework of Compressed Sensing (CS) that go back to the seminal paper of [41], sparse recovery algorithms have found its way to more and more applications. The main result from the CS theory lies in the fact that under some conditions on the so called dictionary $\boldsymbol{D} \in \mathbb{C}^{F \times G}$ sparse signals $\boldsymbol{x}$ can be reconstructed from the noisy observations $\boldsymbol{z}$ given by

$$\boldsymbol{z} = \boldsymbol{D}\boldsymbol{x} + \boldsymbol{\eta}, \tag{4.1}$$

even if $F \ll G$.

The algorithms used to recover the sparse signal $\boldsymbol{x}$ from $\boldsymbol{z}$ can be grouped into three different categories namely convex optimization techniques, greedy methods and combinatorial approaches [42]. A detailed study of sparse methods for DoA estimation can be found in [43]. In this work, we will discuss a cost effective greedy approach as well as a technique that uses convex optimization for the signal reconstruction in Section 4.2. But first, we take a look on how the DoA estimation task can be modeled such that sparse recovery techniques can be applied.

4.1 DoA Estimation as a Sparse Recovery Problem

To develop the formulation of the DoA estimation problem as a sparse recovery task, we first look at the fully sampled case with a single snapshot. The received signal model in the fully sampled case, given by

$$\boldsymbol{y} = \boldsymbol{A}(\boldsymbol{\theta})\boldsymbol{s} + \boldsymbol{\eta}, \tag{4.2}$$

which already shows a structural resemblance to the CS observation model (4.1). However, in contrast to the CS framework, the received signal model in (4.2) features a dense rather than sparse signal vector $\boldsymbol{s}$. Furthermore, the steering matrix $\boldsymbol{A}(\boldsymbol{\theta}) = [\boldsymbol{a}(\theta_1), \ldots, \boldsymbol{a}(\theta_L)] \in \mathbb{C}^{M \times L}$ has more rows than columns. To achieve a form that is suitable for the CS framework, we construct a grid $\boldsymbol{\Theta} = \{\check{\theta}_1, \ldots, \check{\theta}_G\}$ consisting of $G \gg M$ grid-points that spans the whole field of view $[0, 2\pi)$. Let the matrix $\check{\boldsymbol{A}}$ collect all the steering vectors associated with the angular grid-points, i.e.,

$$\check{\boldsymbol{A}} = \left[\boldsymbol{a}(\check{\theta}_1), \ldots, \boldsymbol{a}(\check{\theta}_G)\right]. \tag{4.3}$$

Now, assume that the individual DoAs in $\boldsymbol{\theta}$ lie on the constructed grid, i.e., $\theta_\ell = \check{\theta}_{\rho(\ell)}, \ell = 1, \ldots, L$, then we can write

$$\boldsymbol{A}(\boldsymbol{\theta})\boldsymbol{s} = \check{\boldsymbol{A}}\check{\boldsymbol{s}}, \tag{4.4}$$

where all entries of $\check{\boldsymbol{s}}$ are zeros except for the entries with indices $\rho(\ell), \ell = 1, \ldots, L$. This means that now $\check{\boldsymbol{s}}$ is a sparse vector with $\operatorname{supp}(\check{\boldsymbol{s}}) = L$. Note that the information of interest, namely the DoAs, are now encoded in the support of $\check{\boldsymbol{s}}$. The resulting sparse recovery formulation of (4.2) reads as

$$\boldsymbol{y} = \check{\boldsymbol{A}}\check{\boldsymbol{s}} + \boldsymbol{\eta}. \tag{4.5}$$

For the received signal model with subarray sampling (2.1), we take similar steps as above for the fully sampled case. Now, each individual DoA θ is no longer represented by a single steering vector, but by K reduced steering vectors $\boldsymbol{a}^{(k)}(\theta) = \boldsymbol{G}^{(k)}\boldsymbol{a}(\theta), k = 1, \ldots, K$. Hence, one dictionary element (also called atom) is no longer a single vector, but a matrix

$$\tilde{\boldsymbol{A}}(\theta) = \text{blockdiag}\left\{\boldsymbol{a}^{(1)}(\theta), \ldots, \boldsymbol{a}^{(K)}(\theta)\right\}. \tag{4.6}$$

Then, the corresponding dictionary matrix $\check{\tilde{\boldsymbol{A}}}$ is the concatenation of these elements, i.e., $\check{\tilde{\boldsymbol{A}}} = [\tilde{\boldsymbol{A}}(\check{\theta}_1), \ldots, \tilde{\boldsymbol{A}}(\check{\theta}_K)]$. With this, we can write (2.1) for a single snapshot and under the assumption that all DoAs in $\boldsymbol{\theta}$ lie on the grid as

$$\tilde{\boldsymbol{y}} = \check{\tilde{\boldsymbol{A}}}\check{\tilde{\boldsymbol{s}}} + \tilde{\boldsymbol{\eta}}(n), \tag{4.7}$$

where $\check{\tilde{\boldsymbol{s}}}$ consists of zeros except for L blocks with K entries each. Each one of these blocks gathers the K transmit signals $s_\ell^{(k)}(n), k = 1, \ldots, K$, that stem from the ℓ-th source. Due to this association of each block to one source, the non-zero blocks do not appear at arbitrary positions in $\check{\tilde{\boldsymbol{s}}}$, but only at a position corresponding to a certain atom (4.6). The signal vector $\hat{\tilde{\boldsymbol{s}}}$ displays a special kind of structured sparsity, known as block-sparsity, which has led to an extension of the CS theory and algorithms [44, 45, 46].

Up to now, we considered the single snapshot case. As we have already introduced the concept of block sparsity, an extension to the multi snapshot case (also called

Multi Measurement Vector (MMV) case in the CS community) is straight forward. We construct a large observation vector $\tilde{\boldsymbol{y}}_N$ by stacking the individual observations for each snapshot, i.e., $\tilde{\boldsymbol{y}}_N = \mathrm{vec}\{\tilde{\boldsymbol{Y}}\}$. Then, we can write

$$\tilde{\boldsymbol{y}}_N = \check{\tilde{\boldsymbol{A}}}_N \check{\tilde{\boldsymbol{s}}}_N + \tilde{\boldsymbol{\eta}}_N, \tag{4.8}$$

where $\tilde{\boldsymbol{\eta}}_N = [\tilde{\boldsymbol{\eta}}(1)^{\mathrm{T}}, \ldots, \tilde{\boldsymbol{\eta}}(N)^{\mathrm{T}}]^{\mathrm{T}}$ and each atom in $\check{\tilde{\boldsymbol{A}}}_N$ is $\tilde{\boldsymbol{A}}_N(\theta) = \mathbf{I}_N \otimes \tilde{\boldsymbol{A}}(\theta)$. Therefore, $\check{\tilde{\boldsymbol{s}}}_N$ has L non-zero blocks of size KN.

An alternative way to obtain a sparse formulation of the DoA estimation problem in the SML case with uncorrelated transmit signals uses a sparse representation of the covariance matrix [47, 48, 13]. Again, using a grid $\boldsymbol{\Theta}$ for the individual DoAs, we can obtain a sparse representation of the k-th subarray covariance matrix by first writing

$$\boldsymbol{C}_{\boldsymbol{y}}^{(k)} = \check{\boldsymbol{A}}^{(k)} \,\mathrm{diag}\left\{\check{\boldsymbol{\lambda}}\right\} \check{\boldsymbol{A}}^{(k),\mathrm{H}} + \sigma_\eta^2 \mathbf{I}_W, \tag{4.9}$$

where $\check{\boldsymbol{\lambda}}$ is a L-sparse vector, whose g-th entry denotes the power from a source at $\check{\theta}_g$ and $\check{\boldsymbol{A}}^{(k)} = \boldsymbol{G}^{(k)} \check{\boldsymbol{A}}$. By stacking the vectorizations of $\boldsymbol{C}_{\boldsymbol{y}}^{(k)}$ and after some further reformulations similar to (2.24)-(2.27), we obtain

$$\boldsymbol{c}_{\boldsymbol{y}} = \check{\boldsymbol{V}} \check{\boldsymbol{\lambda}} + \sigma_\eta^2 \mathbf{1}_K \otimes \mathrm{vec}\left\{\mathbf{I}_W\right\}, \tag{4.10}$$

where the g-th atom in $\check{\boldsymbol{V}}$ is the co-array manifold $\check{\boldsymbol{V}}(\check{\theta}_g)$.

Note that when we derived the sparse representations above, we assumed that the actual DoAs $\boldsymbol{\theta}$ lie exactly on the grid $\boldsymbol{\Theta}$. This assumption is in general not feasible, which leads to a model mismatch. The errors induced by this so called grid mismatch problem are well studied in the literature and led to the development of many off-grid and gridless approaches that focus on alleviating this problem (see, e.g., [43]). In this work, we will mainly investigate an off-grid approach that combines the grid-based estimates of a sparse recovery method with gradient steps on the respective likelihood function (details on these hybrid estimators will be discussed in Chapter 6).

4.2 Sparse Recovery Algorithms

4.2.1 Orthogonal Matching Pursuit

As many other algorithms that are used to solve CS problems, the OMP method [49] has originally been developed before the major breakthrough in the CS theory [41]. The OMP algorithm is a greedy approach, and therefore, its computational complexity is much lower compared to convex optimization-based alternatives, such as basis pursuit. Nevertheless, under certain conditions, there exist reconstruction guarantees for OMP that ensure the perfect reconstruction in a noiseless case and limit the estimation error in the presence of noise [50, 51]. The same holds for the extension of OMP to block-sparse vectors, so called Block OMP (BOMP), as has been shown in [46, 52].

The main steps of OMP and BOMP are conceptually identical and can be summarized as follows. The algorithm starts with the observation vector as the residuum. In each step, we first search for the atom in the dictionary with the highest correlation to the residuum, i.e., the atom that can explain the most signal portions of the residuum, and add the DoA of the respective atom to the vector of DoA estimates. Then, we determine the new residuum by eliminating all the signal portions from the observation that can be explained by the current DoA estimates. This process is repeated until either the number of DoA estimates is equal to the known sparsity level of $\check{\boldsymbol{s}}$, or until the norm of the residuum is smaller then a predefined threshold if the sparsity level is unknown a priori. As we assume to know the model order L, which gives us the sparsity level of $\check{\boldsymbol{s}}$, we are performing a known number of steps. A summary of BOMP in pseudo-code for the system model in (4.8) can be found in Algorithm 1.

Algorithm 1: BOMP Algorithm for (4.8)

```
Initialize residuum $\boldsymbol{\rho} = \tilde{\boldsymbol{y}}_N$ and DoA estimates $\hat{\boldsymbol{\theta}} = \emptyset$
for $\ell = 1, \dots, L$ do
    $i_{\text{max}} = \text{argmax}_i \, \|\tilde{\boldsymbol{A}}_N^{\text{H}}(\check{\theta}_i)\boldsymbol{\rho}\|_2^2$
    $\hat{\boldsymbol{\theta}} = [\hat{\boldsymbol{\theta}}^{\text{T}}, \check{\theta}_{i_{\text{max}}}]^{\text{T}}$
    $\hat{\check{\boldsymbol{s}}} = \tilde{\boldsymbol{A}}_N^{+}(\hat{\boldsymbol{\theta}})\tilde{\boldsymbol{y}}$
    $\boldsymbol{\rho} = \tilde{\boldsymbol{y}}_N - \tilde{\boldsymbol{A}}_N(\hat{\boldsymbol{\theta}})\hat{\check{\boldsymbol{s}}}_N$
end
```

As mentioned above, reconstruction guarantees exist for OMP and BOMP [50, 51, 46, 52]. What is common for all of these guarantees is that the sufficient conditions for a successful reconstruction provide an upper bound for the sparsity level (in our case L) that depends on the so called mutual coherence of the dictionary. For a higher sparsity level, the coherence of the dictionary has to be smaller. In the case of a fully sampled array, the mutual coherence is given by

$$\mu = \max_{i \neq j} \frac{|\boldsymbol{a}^{\text{H}}(\check{\theta}_i)\boldsymbol{a}(\check{\theta}_i)|}{\|\boldsymbol{a}(\check{\theta}_i)\|_2 \|\boldsymbol{a}(\check{\theta}_j)\|_2}, \tag{4.11}$$

and the sufficient condition for an exact reconstruction in the noiseless case is given by

[50]

$$L < \frac{1}{2}(\mu^{-1} + 1). \tag{4.12}$$

Note that for noisy observations the sufficient condition is stricter than the inequality given above [51]. For BOMP with the system model (4.8) a similar condition [46] depends on the block-coherence

$$\mu_{\mathrm{B}} = \max_{i \neq j} \frac{1}{KN} \rho \left(\tilde{\boldsymbol{A}}'^{\mathrm{H}}_{N}(\check{\theta}_i) \tilde{\boldsymbol{A}}'_{N}(\check{\theta}_j) \right), \tag{4.13}$$

with the spectral norm ρ and the normalized atom $\tilde{\boldsymbol{A}}'_N(\theta)$, which is formed by normalizing the columns of $\tilde{\boldsymbol{A}}_N(\theta)$, and on the sub-coherence ν that describes the coherence between individual columns of an atom $\tilde{\boldsymbol{A}}_N(\theta)$, which in our case is zero by construction. In this scenario, exact reconstruction via BOMP for the noiseless case is guaranteed for

$$L < \frac{1}{2} \left(\frac{1}{KN} \mu_{\mathrm{B}}^{-1} + 1 \right). \tag{4.14}$$

Comparing the mutual coherence expression (4.11) with (3.17), we see that it is strongly connected to the beampattern of the antenna array. In fact, from the sufficient condition above, we can again identify a trade-off in the number of samples used for the DoA grid $\boldsymbol{\Theta}$, similar to the trade-off between complexity and outlier-probability for grid-based ML estimatiors discussed in Section 3.1.3. In general, the mutual coherence decreases if the beampattern is sampled by fewer samples, however, with fewer samples the grid mismatch problem increases.

Note that for the remainder of the work, we will only distinguish between OMP and BOMP if we want to emphasize some specific differences, otherwise we will simply refer to both of them as OMP; which of the two OMP algorithms can always be inferred from the used model (4.8) or (4.9).

4.2.2 Sparse Iterative Covariance-Based Estimation

The SPICE method has been developed in [47] and uses the sparse representation of the covariance matrix (4.9) for the DoA estimation in the fully sampled case. In [13], the SPICE method was extended to a non-coherent processing approach for partly calibrated arrays. In fact, a non-coherent processing of partly calibrated arrays, i.e., knowledge about fixed phase differences between the individual subarrays is not used, yields the same system model as for time-varying arrays [4]. As the considered subarray sampling is a special case of time-varying arrays, the application of the non-coherent processing method of [13] to the system model at hand is straight-forward.

In the following, we briefly discuss the main ideas behind this non-coherent processing approach, which we simply refer to as SPICE in the following (the distinction between the SPICE method [47] for the fully sampled case and the non-coherent approach [13] is clear from the considered system model). Furthermore, we present an extension of the iterative update rules for SPICE proposed for fully sampled arrays in [47] to systems with subarray sampling [3].

Starting from the sparse representation of the subarray covariance matrices in (4.9), the parameter estimates are found by a covariance matching scheme similar to [39, 10]. The cost function of the matching reads as

$$\sum_{k=1}^{K} \|\boldsymbol{C}_{\boldsymbol{y}}^{(k),-1/2}\left(\hat{\boldsymbol{C}}_{\boldsymbol{y}}^{(k)} - \boldsymbol{C}_{\boldsymbol{y}}^{(k)}\right)\hat{\boldsymbol{C}}_{\boldsymbol{y}}^{(k),-1/2}\|_{\mathrm{F}}^{2}. \tag{4.15}$$

After some reformulation steps (cf. [47]), the minimization of (4.15) can be written as [13]

$$\begin{aligned} &\min_{\boldsymbol{p},\sigma_\eta^2} \sum_{k=1}^{K} \operatorname{tr}\left\{\check{\boldsymbol{C}}_{\boldsymbol{y}}^{(k),-1}\hat{\boldsymbol{C}}_{\boldsymbol{y}}^{(k)}\right\} \\ &\text{s. t.: } \check{\boldsymbol{\lambda}} \geq \boldsymbol{0}, \sigma_\eta^2 \geq 0, \qquad , \\ &\qquad \sum_{g=1}^{G} w_g \check{\lambda}_g + \bar{w}\sigma_\eta^2 = 1. \end{aligned} \tag{4.16}$$

with the weights

$$w_g = \frac{1}{KW}\sum_{k=1}^{K} \boldsymbol{a}^{\mathrm{H}}(\check{\theta}_g)\boldsymbol{G}^{(k),\mathrm{H}}\hat{\boldsymbol{R}}_{\boldsymbol{y}}^{(k),-1}\boldsymbol{G}^{(k)}\boldsymbol{a}(\check{\theta}_g), \tag{4.17}$$

$$\bar{w} = \frac{1}{KW}\sum_{k=1}^{K} \operatorname{tr}\left\{\hat{\boldsymbol{C}}_{\boldsymbol{y}}^{(k),-1}\right\}. \tag{4.18}$$

Note that we added a missing factor of $1/K$ compared to [13, Equation (46)], as (cf. [47, Equation (17)])

$$\begin{aligned} &\sum_{k=1}^{K}\sum_{g=1}^{G} \check{\lambda}_g \boldsymbol{a}^{\mathrm{H}}(\check{\theta}_g)\boldsymbol{G}^{(k),\mathrm{H}}\hat{\boldsymbol{C}}_{\boldsymbol{y}}^{(k),-1}\boldsymbol{G}^{(k)}\boldsymbol{a}(\check{\theta}_g) \\ &\qquad + \sum_{k=1}^{K} \sigma_\eta^2 \operatorname{tr}\left\{\hat{\boldsymbol{C}}_{\boldsymbol{y}}^{(k),-1}\right\} \xrightarrow[N\to\infty]{} KW. \end{aligned} \tag{4.19}$$

The optimization problem (4.16) is a SDP (in fact, it can also be rewritten as a Second Order Cone Program (SOCP) [47]) and can be solved with any general purpose solver. Alternatively, we solve (4.16) by some iterative update rules as has been proposed in [47, Section III] for the fully sampled array. The resulting iterative update rules in the $i+1$-th iteration for subarray sampling can be derived by following the derivations in [47, Section III][1] and result in

$$\check{\lambda}_g^{[i+1]} = \check{\lambda}_g^{[i]} \frac{\left\| \sum_{k=1}^{K} \boldsymbol{a}^{\mathrm{H}}(\check{\theta}_g) \boldsymbol{G}^{(k),\mathrm{H}} \check{\boldsymbol{C}}_{\boldsymbol{y}}^{(k),-1} \hat{\boldsymbol{C}}_{\boldsymbol{y}}^{(k),1/2} \right\|_2}{w_g^{1/2} \xi^{[i]}}, \tag{4.20}$$

$$\sigma_\eta^{2,[i+1]} = \sigma_\eta^{2,[i]} \frac{\left(\sum_{k=1}^{K} \mathrm{tr}\left\{ \check{\boldsymbol{C}}_{\boldsymbol{y}}^{(k),-1} \hat{\boldsymbol{C}}_{\boldsymbol{y}}^{(k)} \check{\boldsymbol{C}}_{\boldsymbol{y}}^{(k),-1} \right\} \right)^{1/2}}{\bar{w}^{1/2} \xi^{[i]}}, \tag{4.21}$$

with

$$\begin{aligned} \xi^{[i]} = & \sum_{g=1}^{G} w_g^{1/2} \check{\lambda}_g^{[i]} \left\| \sum_{k=1}^{K} \boldsymbol{a}^{\mathrm{H}}(\check{\theta}_g) \boldsymbol{G}^{(k),\mathrm{H}} \check{\boldsymbol{C}}_{\boldsymbol{y}}^{(k),-1} \hat{\boldsymbol{C}}_{\boldsymbol{y}}^{(k),1/2} \right\|_2 \\ & + \bar{w} \sigma_\eta^{2,[i]} \sum_{k=1}^{K} \mathrm{tr}\left\{ \check{\boldsymbol{C}}_{\boldsymbol{y}}^{(k),-1} \hat{\boldsymbol{C}}_{\boldsymbol{y}}^{(k)} \check{\boldsymbol{C}}_{\boldsymbol{y}}^{(k),-1} \right\}. \end{aligned} \tag{4.22}$$

[1] Due to notational brevity, we omit the full derivation of these update rules. However, let us briefly summarize the necessary steps. First, an equivalent optimization problem to (4.16) is formulated. This equivalent problem can be solved by an alternating optimization. For each step of this alternating optimization a closed form solution can be found. Finally, the iterative update rules follow from combining these closed form solutions.

5 Machine Learning-Based Estimators

In contrast to the previous estimators, which derived their estimation strategy directly from the underlying stochastic model, we discuss data-based estimators in this chapter. The use of data-based machine learning techniques for DoA estimation dates back to Rastogi et al. [53], who first described the use of a NN for such a task. A comprehensive review of the published literature on machine learning techniques for DoA estimation up to the early 2000s can be found in [54]. Later, motivated by the astonishing results of NNs in the image and speech processing domain, the methods shifted towards larger fully connected and convolutional multilayer NNs [55].

Existing NN approaches for DoA estimation fall into three categories [3]. The idea behind the methods of the first category of NNs is to pose the DoA estimation problem as a classification problem (e.g., [56, 57, 58, 59, 60, 61]). To this end, the field of view is split into several non-overlapping sectors. Now, the NN should determine, which of these sectors contains an active source. For the single source case, this problem is a simple multiclass classification problem, where the number of sectors corresponds to the number of classes. However, for multiple sources, finding a suitable model becomes much more difficult. A possible solution in that case can be obtained by modeling the DoA estimation as a multilabel classification problem, which we will discuss in more detail in Section 5.3.

For the second category, the goal of the NN is to estimate a discretized spatial spectrum, from which we can derive the DoAs. One possible spatial spectrum that can be utilized as such a proxy is the transmit power spectrum [62]. In this case, the resulting training procedure shows a strong resemblance to a multilabel classification approach based on the binary relevance principle (see Section 5.3). Alternatively, in [63], the MUSIC spectrum that corresponds to the observed received signals is used as the target of a regression network.

A more direct approach is the rationale behind methods from the third category. There, the idea is to produce the DoA estimates at the output of the NN. Then, the cost function of interest, e.g., MSE, can be directly used for the training of these regression networks. This approach has the advantage that it works without a discretization of the field of view. Due to this discretization, the methods of the previous categories only work if there is a large enough angular spread between two sources, such that each grid point or sector can be associated with exactly one source. In [64], two different DoA regression networks are used to resolve two narrowly spaced sources. The choice between the two networks is realized by an SNR classification network. A more general approach is presented in [65], where a NN is proposed that simultaneously estimates the number of sources and their respective DoAs.

In the remainder of this chapter, we first discuss the design of the training data set and a common architecture of the proposed NNs. Then, we present a regression approach from the third category, before we take a look at several classification-based approaches to the DoA estimation task. Here, we will also discuss similarities between some classification approaches and spatial spectrum estimation methods, which fall in category number two. Finally, a new NN-based scheme is presented that obtains the DoA estimates by reconstructing the covariance matrix of the full array from the subarray measurements.

5.1 Data, Preprocessing and Architecture

In contrast to the case of more RF chains than sources, for $L \geq W$, the achievable performance of classical estimators and sparse recovery methods is not satisfying for a small number of snapshots (cf. Chapter 7). Hence, a special attention lies on the case of $L \geq W$, when we develop the NN-based estimators. Therefore, we use data from the SML system model (2.13) for the training of the NNs, because then, sufficient identifiability conditions exist (see Section 2.2.2). Thereby, the model order L is fixed. For each realization, the L DoAs $\boldsymbol{\theta}$ are each drawn from a uniform distribution over the complete field of view for the covariance reconstruction and regression-based NNs. To ensure that two sources do not lie in the same sector when applying a classification-based scheme, we first select L distinct sectors randomly and draw the DoA within each sector again from a uniform distribution. According to (2.13), the noise and transmit signal realizations follow a complex normal distribution. To ensure identifiability, the transmit signals are assumed to be uncorrelated. Furthermore, we apply some kind of data preprocessing by fixing the power of the strongest source to $\sigma^2_{s,\text{max}} = 1$, which works as some kind of data normalization. The power of each weaker source in decibel is drawn from a uniform distribution between $0\,\text{dB}$ and $\sigma^2_{s,\text{min}}$. This ensures that the power of a source cannot become arbitrarily small, which would effectively reduce the model order. The noise power is uniformly distributed between $\sigma^2_{\eta,\text{min}}$ and $\sigma^2_{\eta,\text{max}}$ as well. With these settings, we can produce arbitrarily many data samples, each consisting of KN i.i.d. received signal realizations

$\boldsymbol{y}^{(k)}(n), n = 1, \ldots, N, k = 1, \ldots, K$, for the training set. This allows us to feed new, previously unseen realizations to the NN in each step of the gradient descent of the learning algorithm, which makes the training inherently robust towards overfitting. As we know the true DoAs for each data sample, the proposed NN estimators are based on supervised learning.

As input data, we do not directly pass the complex-valued received signal realizations, since state-of-the-art machine learning frameworks such as TensorFlow [66] can only model real-valued NNs. One way to overcome this problem is passing the real and imaginary parts of the received signals to the NN. However, we use a different kind of preprocessing of the input data. As has been shown in [67, 15], sample covariance matrix information is a suitable format for the input data of NNs when we are working in the DoA context. This is not very surprising, as the stochastic model of the input data (2.13) is uniquely parameterized by the subarray covariance matrices. Hence, we first form the K subarray sample covariance matrices $\hat{\boldsymbol{C}}_{\boldsymbol{y}}^{(k)}$, $k = 1, \ldots, K$, from the received signal realizations. Then, we stack their real parameters, i.e., their diagonal elements and the real and imaginary parts of their upper triangle, in one large vector per data sample. The size of the input data is therefore KW^2, and thus, it does not depend on the number of snapshots N, which allows us to pass data samples with a different number of snapshots to the same network.

To obtain some comparability between the different NN approaches, we use a very similar architecture for each network. We use fully connected, feedforward NNs with N_h hidden layers, each consisting of N_u neurons. For the non-linear activation function of the hidden layers, we employ the Rectified Linear Unit (ReLU). Therefore, the architectural differences between the different NNs that we discuss below lie in the structure of the respective output layers, which depend on the individual cost function of each method.

5.2 End-to-End Regression

The end-to-end regression approach discussed in this section falls in the third of the previously discussed categories and has been published in [3]. For the considered NN, the output layer consists of L neurons, which directly provide the estimates of the DoAs $\hat{\boldsymbol{\theta}}$. Therefore, the ideal outputs would be the true DoAs $\boldsymbol{\theta}$ of each data sample. To achieve this goal, we train the NN directly on the cost function of interest.

The most common cost function for parameter estimation is the MSE. However, the MSE criterion does not reflect that for DoA estimation the DoAs are subject to a 2π-periodicity. A more suitable cost function for DoA estimation is therefore the

Mean Squared Periodic Error (MSPE), given by

$$\mathrm{MSPE}(\theta, \hat{\theta}) = \mathrm{E}_{\theta}\left[\left|\mathrm{mod}_{[-\pi,\pi)}\left(\theta - \hat{\theta}\right)\right|^2\right], \tag{5.1}$$

that has been proposed [68], or the Mean Cyclic Error (MCE) [69], which is given by

$$\mathrm{MCE}(\theta, \hat{\theta}) = \mathrm{E}_{\theta}\left[2\left(1 - \cos\left(\theta - \hat{\theta}\right)\right)\right]. \tag{5.2}$$

Both criteria coincide with the MSE for small deviations of $\hat{\theta}$ from the true DoA θ. An advantage of the MCE over the MSPE lies in its differentiability at every point, whereas the MSPE is non-differentiable at π. Although this non-differentiability of the MSPE is only at one point, and hence, can be simply replaced by its left derivative without any adverse impact on the learning procedure, we use the MCE with its continuous derivative for the cost function of the NN.

Apart from the periodicity of the individual DoAs, an additional property should hold for any distance between two DoAs. The order of the DoAs in $\hat{\boldsymbol{\theta}}$, for $L > 1$, should be irrelevant for the value of the cost function. To this end, we take the minimum of the sum of the element-wise errors between the true DoA and all permutations of $\hat{\boldsymbol{\theta}}$ when we compute the cost function $f(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}})$, i.e.,

$$f(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}}) = \min_{\boldsymbol{\Pi}} \sum_{\ell=1}^{L} f\left(\theta_\ell, \boldsymbol{\pi}_\ell^{\mathrm{T}} \hat{\boldsymbol{\theta}}\right), \tag{5.3}$$

where we minimize over all permutation matrices $\boldsymbol{\Pi} = [\boldsymbol{\pi}_1, \ldots, \boldsymbol{\pi}_L]^{\mathrm{T}}$. Such a minimization over all permutations in the cost function of the NN adds a significant computational load on the training procedure. However, in our simulations we observed that if we compute the minimum over all permutations as in (5.3) and feed the labels sorted in ascending order to the NN, then, at the end of the training, it produces ordered DoA estimates where the optimal permutation matrix $\boldsymbol{\Pi}$ is the same for all realizations. Furthermore, our simulations showed that if the minimization over all permutations matrices is omitted, the NN produces the outputs in the optimal order such that we can simply use the sum of the element-wise MCEs for the cost function. In the following, we refer to this NN as *MCENet*.

5.3 End-to-End Classification

In this section, we discuss several schemes that formulate the DoA estimation problem as a classification task. All of the presented methods share a common notion. The field of view from $[0, 2\pi)$ is split into G distinct sectors with the same width $\theta_{\mathrm{w}} = 2\pi/G$, such that the g-th sector ranges from $[(g-1)\theta_{\mathrm{w}}, g\theta_{\mathrm{w}})$. Now, we are no longer interested in obtaining the continuous estimates of the DoAs directly, as in the previous section, but in the indices of the sectors that include an active source. If the NN determines that the g-th sector contains an active source, we can obtain an estimate for the DoA of this source by taking the mid point of this sector $(g - \frac{1}{2})\theta_{\mathrm{w}}$.

5.3.1 Single Source Case

We first consider the single source case. For this scenario, the formulation of the classification problem is straight-forward and has been presented in [56, 70]. Since there is only one source, we are trying to find the one sector out of all G sectors, which is most probable to include the active source. This is a standard multiclass classification problem, which can be solved by training on the so called categorical cross-entropy loss function. To that end, we encode the index of the true sector for each training data sample in a one-hot vector, i.e., the label of the true sector is 1, whereas the label for all other sectors is 0. At the neurons of the output layer a softmax operation is applied, which produces G outputs $z(g), g = 1, \dots, G$, between zero and one, whose sum is again one [71]. In combination with a training based on the cross-entropy loss, given by

$$\max_{\boldsymbol{w}} \ln \left(z(g^* | \boldsymbol{x}; \boldsymbol{w}) \right). \tag{5.4}$$

where $\boldsymbol{w}$ are the weights and biases of the NN and g^* is the index of the active sector of the training data sample $\boldsymbol{x}$, these output values $z(g)$ can be interpreted as estimates of the posterior probabilities for each class g conditioned on $\boldsymbol{x}$[1]. As has been discussed in [72], the training based on (5.4) can be interpreted as a heuristic approach to the theoretically optimal MAP estimator, as the NN is tuned to maximize the estimate of the posterior probability of the correct class g^* of the input vector $\boldsymbol{x}$.

5.3.2 Multiple Source Case

For more than one source, things get a bit more difficult. First of all, let us assume that in each sector, there can be at most one source. Otherwise, if the number of active sectors is smaller then the number of sources L, we do not know which active sector contains multiple sources. Note that this assumption requires G to be sufficiently large.

[1] Hence, we denote the elements of the output vector $\boldsymbol{z}$ by $z(\cdot)$ to emphasize that they can be interpreted as a density.

The DoA estimation task can then be expressed as a multilabel-multiclass classification problem, also referred to as multidimensional classification (see, e.g., [73, 74]), i.e., we try to find a subset consisting of L labels, where each label can attain G different classes.

5.3.2.1 Power-Set Method

A direct extension of the multiclass formulation from the single source case to multiple sources leads to the *label power-set method* [75]. There, we assign each combination of sectors to one class. Since we do not care for the order of the DoA estimates and two sources cannot lie in the same sector, for modeling the DoA estimation problem with the label power-set method it suffices to consider all unordered L-tuples of sectors without repetition, i.e., the necessary number of classes is given by the binomial coefficient $\binom{G}{L}$. This means that, for $G \gg L$, the number of classes, which is equal to the number of neurons in the output layer, grows exponentially with the number of sources. Hence, the power-set method quickly becomes computationally infeasible.

5.3.2.2 Binary Relevance Method

A more tractable alternative is the *binary relevance approach* [75], which has been used for DoA estimation in [58, 60, 61]. The idea is to train a network that realizes G binary classifiers, i.e., one for each sector, where the g-th classifier tries to produce the probability that a given data sample stems from a scenario where there is an active source in the g-th sector. The label vector $\boldsymbol{v}$ for each training data sample is therefore a vector of length G, which has a one at each element that corresponds to an active sector and zeros elsewhere. To obtain outputs $z(g), g = 1, \dots, G$, between 0 and 1 that can be interpreted as probabilities, a sigmoid activation is employed at every output neuron. The loss function for the whole network is the sum of the binary cross-entropies of each binary classifier, i.e.,

$$-\sum_{g=1}^{G} \left[v_g \ln\left(z(g)\right) + (1 - v_g) \ln\left(1 - z(g)\right)\right]. \tag{5.5}$$

This approach has two major drawbacks. First, with the binary relevance method, the a priori knowledge about the number of sources L is not directly taken into account, but only through the training data[2]. And second, the output of g-th binary classifier is

[2] In scenarios where the model order is not known a priori, this can be an advantage, as a single NN can be trained for multiple model orders and the model order selection can be accomplished by the same NN (e.g., using thresholding of the output).

formed independently from the decision of the other $G-1$ binary classifiers. This means that the g-th classifier does not consider the output of the classifier for sector $g', g' \neq g$, which might indicate that a source lying in sector g' is much more probable than in sector g. Especially in the considered DoA estimation scenario, this is a problem, as the outputs between two binary classifiers corresponding to neighboring sectors are highly correlated. This can be seen in the Fig. 5.1, which shows the outputs $\boldsymbol{z}$ of the trained NN for an exemplary realization in blue. We would expect that the resulting probabilities are close to zero except for three sectors such that the DoA estimates can be found by taking the three largest entries (or thresholding if the number of sources is unknown), but in fact, they take the shape similar to a spectrum with three modes around the true DoAs. This resembles the spatial spectra that we know from MUSIC or beamformer methods, where this behavior can be explained by the respective spectrum generating function and the fact that there are only small differences between the steering vectors for two closely spaced DoAs. Hence, for these methods the L largest peaks in the spatial spectrum are chosen as the estimates, instead of the L largest points. For DoA estimation based on a binary relevance classification this peak finding procedure has been proposed in [58]. We will refer to the NN based on the binary relevance scheme in combination with peak detection as *BRClaNet*.

Note that there is some connection between the discussed binary relevance scheme and the estimation of the power spectrum via NNs presented in [62]. In both cases, the field of view is first discretized, then, the training objective tries to obtain non-zero outputs for the true positions while bringing the other outputs close to zero. Finally, the largest peaks in the respective spatial spectrum serve as the DoA estimates.

A similar approach to the BRClaNet is considered in [59], where the output layer again consists of G neurons. In contrast to the binary relevance scheme, a softmax activation is employed in the last layer. The authors claim to train the network with the cross-entropy loss function, however, in our attempts to replicate this method, this led to a divergence of the NN. Instead, we scale the outputs of the softmax activation $\boldsymbol{z}$ before we apply the cross-entropy loss function, i.e.,

$$-\sum_{g=1}^{G} v_g \ln\left(Lz(g)\right) \tag{5.6}$$

is used as the training objective. However, an interpretation of this approach, which we will refer to by *CEClaNet*, as an approximation of the MAP estimator, as in the single source case, does no longer suggest itself. Again, we obtain a spatial spectrum at the outputs (cf. Fig. 5.1), which is not very surprising since CEClaNet shares the rationale behind the training loss with BRClaNet and the power spectrum estimation in

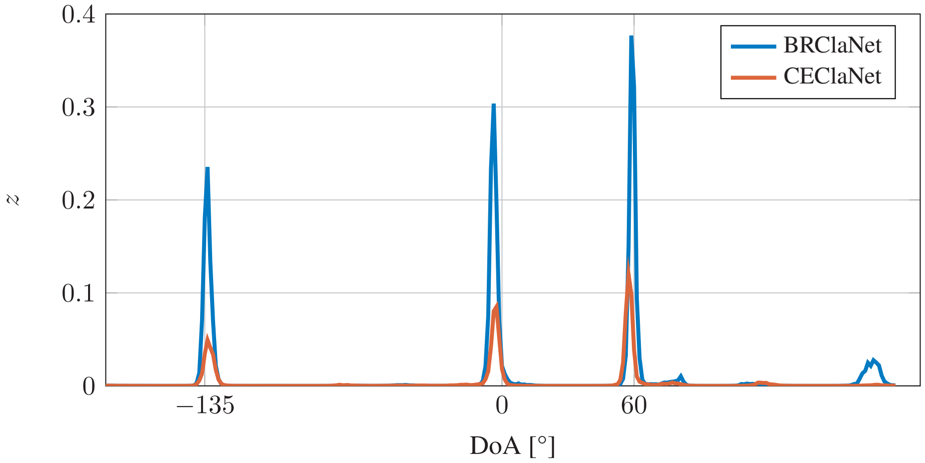


Figure 5.1: Exemplary Classifier Output for Three Equally Powered Sources at $\boldsymbol{\theta} = [-135°, 0°, 60°]$ at 20 dB SNR and $N = 10$.

[62], i.e., increasing the outputs corresponding to the active sectors, while maintaining the outputs of the inactive sensors close to zero.

5.3.2.3 Classifier Chains

Another approach to multilabel classification that is able to cope with dependencies between labels are binary classifier chains [76, 77]. One of these binary classifier chains consists of a concatenation of G binary classifiers (one for each label), where each node in this chain bases its decision on the original input data sample and the output of the previous nodes. One problem with such an architecture is that its performance is heavily dependent on the order of the binary classifiers. Therefore, the averaging of the outputs over an ensemble of binary classifier chains with different orders have been investigated in [76].

For the DoA estimation problem, we propose a different architecture, which has been heavily inspired by these binary classifier chains. As we have a multilabel-multiclass classification problem, where we know the number of true labels L a priori, we can construct a classifier chain consisting of L multiclass classifiers. This means that each node of this multiclass classifier chain shall provide an estimate for one of the L sources. In other words, the proposed classifier chain estimates the DoAs sequentially. To that end, each multiclass classifier in the chain has—apart from the input data—the structure of a multiclass classifier for a single source as discussed above, i.e., a neural network with G output neurons, whose outputs are subject to a

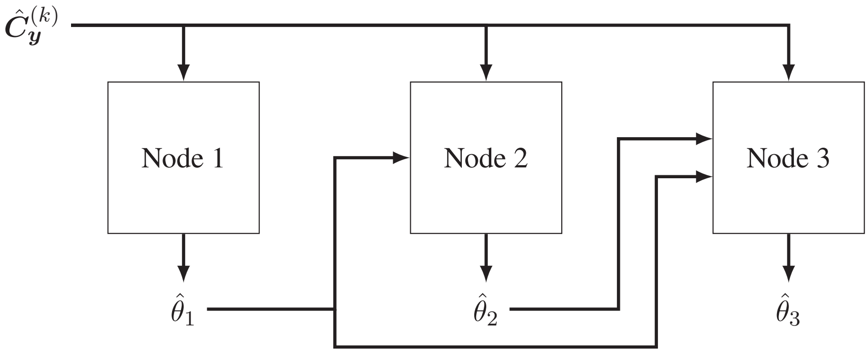


Figure 5.2: Illustration of ChainNet Architecture for $L = 3$.

softmax activation and is trained based on a cross-entropy loss function. Again, each node in the chain obtains the original sample covariance matrices of the subarrays and the DoA estimates of the previous nodes as its input (see Fig. 5.2). We refer to such a multiclass classifier chain as *ChainNet*[3].

What we did not talk about yet, is how the training for each of those nodes looks like. Each multiclass classifier shall produce the DoA estimate for one of the sources, however, it is still unclear which source should be estimated by which classifier. Therefore, we propose to estimate the sources in an ascending order of their true DoAs[4] similar to the order in which the MCENet approach produces its outputs. For the training of the individual nodes, we can follow two different paths. One way is training the first classifier as for the single source case. Then, using the first node of the classifier chain to estimate the first DoA for the whole training set. This information alongside the subarray sample covariance information is subsequently used in the second node, etcetera. Another option is to train all of the nodes simultaneously with the assumption that the estimates from the previous nodes are perfect. In that case, the training data for the second node again consists of the sample covariance matrices of the subarrays and the perfect information on which sector comprises the first source. This variant does not only allow to train all the nodes simultaneously, but provided a better estimation accuracy in our simulations.

Finally, we want to discuss a final augmentation to the multiclass classifier approach, denoted by *ProjNet*. In ChainNet, every node has to learn how to take the previous DoA

[3] After the submission of this dissertation, a more detailed discussion of the ChainNet approach has been published in [78].

[4] We have also investigated an ordering of the sources by their aggregated instantaneous transmit power. However, this approach resulted in a very poor estimation accuracy for more than two sources.

estimates into account such that it can identify the next source. So to say, the NN has to find a way to ignore the portions of the input data, which already led to the previous estimates. This reminds us of the OMP algorithm discussed in Section 4.2.1. There, we also estimate the source DoAs in an iterative procedure and use an orthogonal projection in between steps to remove the signal portions that can be explained by the already found estimates. Such a projection step can also be used between two nodes of the classifier chain. Then, the input for the ℓ-th multiclass classifier is no longer the original subarray covariance information and the $\ell - 1$ previous DoA estimates $\hat{\boldsymbol{\theta}}_{\ell-1}$, but the projected version of the original input data, i.e., the projected subarray covariance matrices

$$\mathbf{P}^{\perp}_{\boldsymbol{A}^{(k)}(\hat{\boldsymbol{\theta}}_{\ell-1})} \hat{\boldsymbol{C}}^{(k)}_{\boldsymbol{y}} \mathbf{P}^{\perp}_{\boldsymbol{A}^{(k)}(\hat{\boldsymbol{\theta}}_{\ell-1})}, \tag{5.7}$$

and the $\ell - 1$ previous DoA estimates $\hat{\boldsymbol{\theta}}_{\ell-1}$[5]. In other words, we replace the search for the atom with the highest correlation in the OMP algorithm with a multiclass classification network.

5.4 Covariance Matrix Recovery

The last method that we investigate in detail follows a concept, which does not perfectly fit in one of the three NN categories discussed above. Similar to the methods from the second category, the NN estimates some proxy that can be used to infer the DoAs. However, for this approach, the NN does not estimate a spectrum of some sense, but we try to reconstruct a covariance matrix for the fully sampled array from the subarray sample covariance matrices.[6] This estimate of the full covariance matrix can then be used with any kind of classical DoA estimation algorithm such as MUSIC [12] to obtain the DoA estimates $\hat{\boldsymbol{\theta}}$.

At first, one might wonder if the estimation of the full covariance matrix from subarray observations can be feasible. Recall that the subarray observations include all the necessary information to estimate the parameters of the PDF of the received signals (2.13), i.e., the DoAs $\boldsymbol{\theta}$, the transmit powers $\boldsymbol{\lambda}$, and the noise power σ_η^2. From these parameters, we can naturally form an estimate of the full covariance matrix, by utilizing the steering vectors of the full array. Hence, there exists a function that maps the sample covariance matrices of the subarrays to a matrix that has the form of a covariance matrix of the fully sampled array. Then, by the universal approximation theorem

[5] One might consider not feeding the previous DoA estimates to the next stage, as the respective signal portions have been removed. However, this approach led to a much worse performance in our simulations.

[6] The discussed approach has been published in [79] after the submission of this thesis.

[80, 81], a multilayer, feedforward NN of appropriate width is able to approximate this mapping. So, this covariance matrix reconstruction approach is feasible in general and a NN should be able to solve this task.

The more interesting question is if the NN estimates the parameters internally and reconstructs the full covariance matrix from there, or if there is a more direct mapping between the subarray covariance matrices and their fully sampled counterpart. In the former case, any subsequent classical DoA estimation scheme that is applied to the full covariance matrix estimate can only be as good as the internal DoA estimate of the NN. In that case, a NN which directly produces these parameter estimates at its output should be superior, as we know the mapping from these parameters to the respective full covariance matrix perfectly and can therefore train the NN with the same cost function as in the case where the output forms the full covariance matrix estimate. On the other hand, if the NN finds a direct mapping from the subarray covariance matrices to the full covariance information, then, this suggests that there is some additional structural information embedded in the full covariance matrix that can be leveraged by the NN.

In fact, such structural information is well known for the case of Uniform Linear Arrays (ULAs), where the fully sampled covariance matrix is a hermitian Toeplitz matrix. For ULAs, we can construct a primitive estimator for the full covariance matrix by averaging the (weighted) entries of the subarray covariance matrices that correspond to the same covariance lag[7] and completing the full covariance matrix from these estimates, similar to the construction of the co-array manifold of sparse linear arrays (e.g., [82]). For example, suppose that we have a four element ULA and two RF chains. We write the covariance matrix of the full array as

$$\boldsymbol{C}_{\text{full}} = \begin{bmatrix} c_0 & c_1 & c_2 & c_3 \\ c_1^* & c_0 & c_1 & c_2 \\ c_2^* & c_1^* & c_0 & c_1 \\ c_3^* & c_2^* & c_1^* & c_0 \end{bmatrix}, \tag{5.8}$$

where c_i denotes the i-th covariance lag. Furthermore, let us assume that for the k-th subarray we always sample the first antenna and the $(k+1)$-th antenna, i.e., $\mathcal{G}_k = \{1, k+1\}$. Then each of the $K = 3$ subarrays samples the zeroth covariance

[7] For ULAs, combinations of two antenna elements that belong to the same covariance lag have the same distance between both antennas. The resulting entries in the full covariance matrix are located on the same off-diagonal.

lag twice and the k-th covariance lag once, i.e.,

$$\boldsymbol{C}_{\boldsymbol{y}}^{(k)} = \begin{bmatrix} c_0 & c_k \\ c_k^* & c_0 \end{bmatrix}. \tag{5.9}$$

Then a primitive estimate of the full covariance matrix from the subarray sample covariance matrices

$$\hat{\boldsymbol{C}}_{\boldsymbol{y}}^{(k)} = \begin{bmatrix} \hat{c}_{0,1}^{(k)} & \hat{c}_k^{(k)} \\ \hat{c}_k^{*,(k)} & \hat{c}_{0,2}^{(k)} \end{bmatrix} \tag{5.10}$$

can be constructed by

$$\hat{\boldsymbol{C}}_{\text{full}} = \begin{bmatrix} \gamma_0 & \gamma_1 & \gamma_2 & \gamma_3 \\ \gamma_1^* & \gamma_0 & \gamma_1 & \gamma_2 \\ \gamma_2^* & \gamma_1^* & \gamma_0 & \gamma_1 \\ \gamma_3^* & \gamma_2^* & \gamma_1^* & \gamma_0 \end{bmatrix}, \tag{5.11}$$

with

$$\gamma_0 = \frac{1}{2K} \sum_{k=1}^{K} \sum_{i=1}^{2} \hat{c}_{0,i}^{(k)}, \tag{5.12}$$

$$\gamma_k = \hat{c}_k^{(k)}. \tag{5.13}$$

In the low snapshot domain, we can replace (5.13) with

$$\gamma_k = \frac{\gamma_0}{\sum_{i=1}^{2} \hat{c}_{0,i}^{(k)}} \hat{c}_k^{(k)}, \tag{5.14}$$

to combat differences in the amplitude of the transmit signal realizations between different snapshots. However, note that for both variants, the resulting estimate of the full covariance matrix (5.11) is not guaranteed to be positive semidefinite, especially if the number of snapshots is low.

Now, let us discuss the design of a NN for the covariance matrix reconstruction. Again, we use a supervised learning approach, where now, the labels are the ideal full covariance matrices for each realization. By ideal full covariance matrix, we understand the data generating covariance matrix for a certain realization, i.e., the matrix

$$\boldsymbol{C}_{\text{full}} = \boldsymbol{A}(\boldsymbol{\theta}) \operatorname{diag}\{\boldsymbol{\lambda}\} \boldsymbol{A}^{\mathrm{H}}(\boldsymbol{\theta}) + \sigma_\eta^2 \mathbf{I}_M, \tag{5.15}$$

where the realizations of $\boldsymbol{\theta}$, $\boldsymbol{\lambda}$ and σ_η^2 have been drawn according to the data model described in Section 5.1. This means, that the labels do not depend on the observed transmit signal and noise realizations but the true second oder moments, in contrast to the sample covariance matrices of the subarrays that are used for the input data.

For the output format and cost function of the NN, we investigate two different approaches. In the first approach, which we will refer to by *ReconNet*, the outputs of the network are estimates of the real parameters of the full covariance matrix, i.e., its diagonal entries, and the real and imaginary parts of the upper triangular elements. Therefore, the ReconNet has M^2 output neurons. From these parameter estimates, we can reconstruct an estimate of the full covariance matrix, which we denote by $\hat{\boldsymbol{C}}_{\text{recon}}$. The training on the network is then based on the MSE between the label covariance matrix $\boldsymbol{C}_{\text{full}}$ and the estimate $\hat{\boldsymbol{C}}_{\text{recon}}$. Note that the estimate of the full covariance matrix $\hat{\boldsymbol{C}}_{\text{recon}}$ is not a positive semidefinite matrix, which leads us to the second approach, which we denote by *GramNet*. There, the output consists of $2M^2$ neurons, which form the real and imaginary entries of a matrix $\boldsymbol{F} \in \mathbb{C}^{M\times M}$. The estimate of the full covariance matrix is then formed by the regularized Gramian $\hat{\boldsymbol{C}}_{\text{gram}} = \boldsymbol{F}^{\text{H}}\boldsymbol{F} + \varepsilon\mathbf{I}_M$. This ensures that the estimate $\hat{\boldsymbol{C}}_{\text{gram}}$ is not only hermitian, but a positive definite matrix (with smallest eigenvalue larger or equal to ε). Hereby, the regularization parameter ε is chosen much smaller than $\sigma^2_{\eta,\text{min}}$ such that its influence on the estimate is negligible, but large enough to guarantee strict positive definiteness during training. With this, we can use any distance $d(\boldsymbol{A}, \boldsymbol{B})$ between two positive definite matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ to obtain a cost function

$$f_{\text{gram}}\left(\boldsymbol{C}_{\text{full}}, \hat{\boldsymbol{C}}_{\text{gram}}\right) = \left(d\left(\boldsymbol{C}_{\text{full}}, \hat{\boldsymbol{C}}_{\text{gram}}\right)\right)^2, \tag{5.16}$$

for the training of GramNet.

We can find many different distance measures between two positive definite matrices. One of them is the well known Frobenius norm, which can be used to compute a distance between any two matrices of the same size. However, we are especially interested in distances, which measure the length of the shortest path between the two matrices that lies completely inside the positive definite cone. So called geodesic distances fulfill this property (see, e.g., [83, Chapter 7.1]). For the positive definite cone, the *affine invariant distance*, given by [84, Chapter 6.1]

$$d_{\text{AffInv}}(\boldsymbol{A}, \boldsymbol{B}) = \left\| \ln\left(\boldsymbol{A}^{-1/2}\boldsymbol{B}\boldsymbol{A}^{-1/2}\right)\right\|_{\text{F}}, \tag{5.17}$$

where $\ln(\boldsymbol{A})$ denotes the matrix logarithm of $\boldsymbol{A}$, and the *log-euclidean distance*, which reads as [85]

$$d_{\text{LogEuc}}(\boldsymbol{A}, \boldsymbol{B}) = \|\ln(\boldsymbol{A}) - \ln(\boldsymbol{B})\|_{\text{F}} \tag{5.18}$$

are such geodesic distances. Apart from these geodesic distances, we study the use of several other non-geodesic distance measures summarized in Table 5.1, which have

Name	$d(\boldsymbol{A},\boldsymbol{B})$
Frobenius	$\|\boldsymbol{A}-\boldsymbol{B}\|_\mathrm{F}$
Affine Invariant [84, Chapter 6.1]	$\left\|\ln\left(\boldsymbol{A}^{-1/2}\boldsymbol{B}\boldsymbol{A}^{-1/2}\right)\right\|_\mathrm{F}$
Log-Euclidean [85]	$\|\ln(\boldsymbol{A})-\ln(\boldsymbol{B})\|_\mathrm{F}$
Cholesky-Frobenius [87]	$\|\operatorname{chol}\{\boldsymbol{A}\}-\operatorname{chol}\{\boldsymbol{B}\}\|_\mathrm{F}$
J-Divergence [88]	$\frac{1}{2}\sqrt{\operatorname{tr}\{\boldsymbol{A}\boldsymbol{B}^{-1}+\boldsymbol{B}\boldsymbol{A}^{-1}-2\mathbf{I}\}}$
Jensen-Bregman LogDet Divergence [89]	$\sqrt{\ln(\det\{\frac{1}{2}(\boldsymbol{A}+\boldsymbol{B})\})-\frac{1}{2}\ln(\det\{\boldsymbol{A}\boldsymbol{B}\})}$
Bures-Wasserstein [90]	$\sqrt{\operatorname{tr}\{\boldsymbol{A}\}+\operatorname{tr}\{\boldsymbol{B}\}-2\operatorname{tr}\{(\boldsymbol{A}^{1/2}\boldsymbol{B}\boldsymbol{A}^{1/2})^{1/2}\}}$

Table 5.1: Distance Measures for Positive Definite Matrices

been proposed for positive semidefinite matrices (cf. [86]), for the cost function of GramNet.

We conclude this section by a brief summary of the MUSIC estimator [12], as we will use it to obtain DoA estimates from the estimated full covariance matrix in our simulations. The main concept behind the MUSIC estimator lies in the spectral representation of the true full covariance matrix $\boldsymbol{C}_\text{full}$. The eigenvectors corresponding to the L largest eigenvalues span the so called signal subspace, in which the steering vectors corresponding to the true DoAs lie. The space spanned by the other eigenvectors is denoted by noise subspace and is orthogonal to the signal subspace. Therefore, we can estimate the noise subspace from the covariance estimate $\hat{\boldsymbol{C}}$ by means of an eigenvalue decomposition and identifying the $M-L$ smallest eigenvalues. The respective eigenvectors $\boldsymbol{U}_\text{noise}$ form a basis of the estimated noise subspace. Hence, for an ideal estimate of the noise subspace, any steering vector that lies in the signal subspace is orthogonal to $\boldsymbol{U}_\text{noise}$. In practice, the estimate of the noise subspace is not ideal, therefore, we cannot expect that there exist steering vectors that are truly orthogonal to the estimated noise subspace. Instead, we look for the steering vectors whose projection on to the estimated noise space is small, i.e., the angle between these steering vectors and the noise subspace is large. To that end, we evaluate the so called MUSIC spectrum

$$S_\text{MUSIC}(\theta)=\frac{1}{\boldsymbol{a}^\mathrm{H}(\theta)\boldsymbol{U}_\text{noise}\boldsymbol{U}_\text{noise}^\mathrm{H}\boldsymbol{a}(\theta)}, \tag{5.19}$$

and identify the DoA estimates by finding the L largest peaks in $S_\text{MUSIC}(\theta)$.

Hybrid DoA Estimation 6

As we have already discussed in Chapter 4, the considered sparse recovery methods are grid-based estimators, which suffer from the so called grid mismatch problem. Similarly, the GLS estimator presented in Section 3.2, the classification NNs as well as a MUSIC estimator applied to the covariance recovery NNs provide estimates for a finite grid of DoA values. Increasing the number of grid-points to reduce the grid mismatch is not always feasible due to the resulting increase in computational complexity or the coherence of the dictionary. Instead, we can employ a so called *hybrid DoA estimation* scheme. By hybrid DoA estimation we understand the combination of two different estimation approaches in a two-stage process. In the first step, we evaluate one of the estimators, which has been presented in the previous chapters. Then, we improve the estimate obtained in the first step by gradient steps on the respective likelihood function. From another perspective, hybrid DoA estimation is an approximation of the ML estimator, where the initial grid search on the likelihood function is replaced by a different estimator with reduced complexity. In that sense, hybrid DoA estimation cannot only alleviate the grid mismatch problem, but also improve the estimates of the MCENet, which can produce continuous estimates by construction, as it provides a direct way to combine the purely data based approach with the knowledge about the underlying stochastic model.

In the case of the NN-based estimators described in Chapter 5, an additional intermediate step is necessary in the SML case. As these estimators do not provide estimates for the noise variance and transmit covariance matrix, we have to find initial values of these parameters for the subsequent gradient steps. To that end, we can use the GLS estimates of these nuisance parameters for the fixed angular estimates $\hat{\boldsymbol{\theta}}$, which requires the solution of a convex optimization problem as discussed in Section 3.2. In the DML case, explicit estimates of the noise power and the transmit signals are not necessary as the gradient steps can be implemented on the concentrated log-likelihood function (3.8).

At the end of this brief chapter, we want to address an issue that we encountered when performing gradient steps on the SML likelihood in the case of more sources than RF chains. There, we observed that the gradient is often dominated by the derivative w.r.t. the noise variance σ_η^2, i.e., the partial derivative in this direction is by magnitudes larger than for the other directions. This, in turn, can lead to a slow progress in DoA estimates, which are the parameters of interest, if a simple gradient ascent approach is employed. Instead, we propose to use a block coordinate ascent method that alternates between updating the DoA estimates, the estimate of the signal covariance $\boldsymbol{R_s}$, and an estimate of σ_η^2. This circumvented the aforementioned problem and led to a much faster convergence in our simulations, as the stepsize can be adapted individually in each step of the block coordinate ascent.

Simulation Results – DoA Estimation 7

In this chapter, we assess the achievable MSE performance of the previously presented DoA estimation algorithms by means of Monte Carlo simulations. After discussing some of the common parameters for our simulations, we take a look at scenarios with fewer sources than RF chains. Finally, we present simulation results for a scenario with an equal number of sources and RF chains.

7.1 Simulation Setting and Neural Network Parameters

In our simulations, we focus on Uniform Circular Arrays (UCAs) with a field of view from 0 to 2π. For a UCA with M omnidirectional antennas, the steering vector is given by

$$\boldsymbol{a}_{\mathrm{UCA}}(\theta) = \begin{bmatrix} \mathrm{e}^{-\mathrm{j}2\pi\frac{R}{\lambda}\cos(\theta)\cos(\phi)} \\ \mathrm{e}^{-\mathrm{j}2\pi\frac{R}{\lambda}\cos(\theta-\frac{2\pi}{M})\cos(\phi)} \\ \vdots \\ \mathrm{e}^{-\mathrm{j}2\pi\frac{R}{\lambda}\cos(\theta-\frac{2\pi(M-1)}{M})\cos(\phi)} \end{bmatrix}, \tag{7.1}$$

where θ is the azimuth angle and ϕ is the elevation angle to the source, R denotes the array radius, and λ is the wavelength of the impinging electro-magnetic wave. In the following we assume that all the sources lie in the same plane as the antenna elements, such that the elevation ϕ is 0 for all sources, and therefore, the DoAs are fully described by the azimuth angles $\boldsymbol{\theta}$. If not stated otherwise, the considered array consists of $M = 9$ antennas and the ratio of radius to wavelength is 1. Furthermore, we consider a scenario with $W = 3$ RF chains, which are used to sample $K = 4$ subarray configurations. The selected antennas per subarray are given in Table 7.1, which uses a clockwise numbering of the antenna elements of the UCA.

In general, we use an oversampling factor $Q = 32$ for the DoA estimation algorithms. This means that for the sparse recovery methods OMP and SPICE the field of view is sampled by Q times M is $G = 288$ grid points. Similarly, the MUSIC spectrum of the NN-based methods from Section 5.4 is evaluated on the same grid and the number of sectors for the classification based methods in Section 5.3 is also $Q \cdot M$.

k	Antenna Elements
1	$1,\ 2,\ 9$
2	$1,\ 3,\ 8$
3	$1,\ 4,\ 7$
4	$1,\ 5,\ 6$

Table 7.1: Subarray Sampling Scheme

Parameter	Value
$\sigma_{s,\mathrm{min}}$	$-9\,\mathrm{dB}$
$\sigma_{\eta,\mathrm{min}}$	$-10\,\mathrm{dB}$
$\sigma_{\eta,\mathrm{max}}$	$30\,\mathrm{dB}$
N_h	4
N_u	4096
Weight Initialization	Glorot [91]
Batch Size	256
Optimizer	Adam [92]
Learning Rate	10^{-4}
Samples per Training Set	$64 \cdot 10^6$

Table 7.2: Simulation Parameters DoA Estimation

However, due to the exponentially growing complexity of the initial grid search in the ML and GLS estimators, we utilize a reduced oversampling factor in these cases. For $L = 2$, we choose an oversampling factor of 16, and for $L = 3$, Q is only 8, as otherwise the evaluation of meaningful amount of Monte Carlo simulations becomes intractable.

As discussed in Chapter 5, the architecture of the employed NNs is identical apart from the input and output layers. The common parameters for the hidden layers, the training set, and the optimizers is summarized in Table 7.2. The respective dimensions of the input and output layers of each network can be easily inferred from the parameter values provided above.

7.2 Fewer Sources than RF Chains

7.2.1 MSE Approximation

First, we verify the MSE approximation presented in Section 3.1.3. In Fig. 7.1, we consider the single source case and show the Root Mean Squared Periodic Error (RMSPE) of the grid-based DML estimator and the corresponding grid-based MSE approximation for different SNR. For comparison, we add the performance of the gridless DML estimator that is initialized by a grid search, Athley's MSE approximation for the gridless case [28], and the CRB (cf. Section 2.3.1). To obtain accurate DML estimates, 10^5 Monte Carlo runs were performed for each SNR. The source is located at the fixed DoA $\theta^\star = 10.5^\circ$. Furthermore, we consider a single snapshot only and the respective transmit signal is 1. The oversampling factor is $Q = 1$, i.e., we have an extremely coarse grid. In Fig. 7.1, we see that our proposed grid-based approximation can predict the threshold point, i.e., the transition from the threshold to asymptotic region, accurately. In contrast, Athley's MSE approximation predicts the threshold point at a significantly lower SNR. Additionally, we can identify that with the initialization on a very coarse grid, Athley's gridless approximation can no longer predict the performance of the gridless DML estimator, as the subsequent gradient steps cannot recover from a grid-based outlier initialization.

The effect of the local error approximation can be observed in Fig. 7.2 for an oversampling factor of 12. In this case, we can see that the local error approximation is not only an accurate approximation for the high SNR regime (above 16 dB SNR), where the MSE saturates to the grid resolution, but also between 8 dB and 16 dB. There, we are also operating in the asymptotic region and the grid-based approximation gives a more precise prediction than the CRB approximation that is used by Athley's method.

For multiple sources, we see similar results as for the single source case. Fig. 7.3 depicts the simulation results for two sources at 0.25° and 50.5°, respectively, for an oversampling factor of 16. Again, the transition between the asymptotic region and the threshold region is well predicted in this scenario. More importantly, Fig. 7.3 shows that our derived local error approximation is able to accurately approximate the MSE in the asymptotic region (above 16 dB), which validates the proposed local error approximation also for the multiple source case.

As has been discussed in [27] there exist some pathologic cases for very low oversampling factors, when the true DoAs lie centered between two or more grid points, and the asymptotic log-likelihood $\bar{U}(\boldsymbol{\theta})$ is almost equal for these grid points, but significantly lower than the asymptotic log-likelihood at the true DoAs $\bar{U}(\boldsymbol{\theta}^\star)$. Then, it

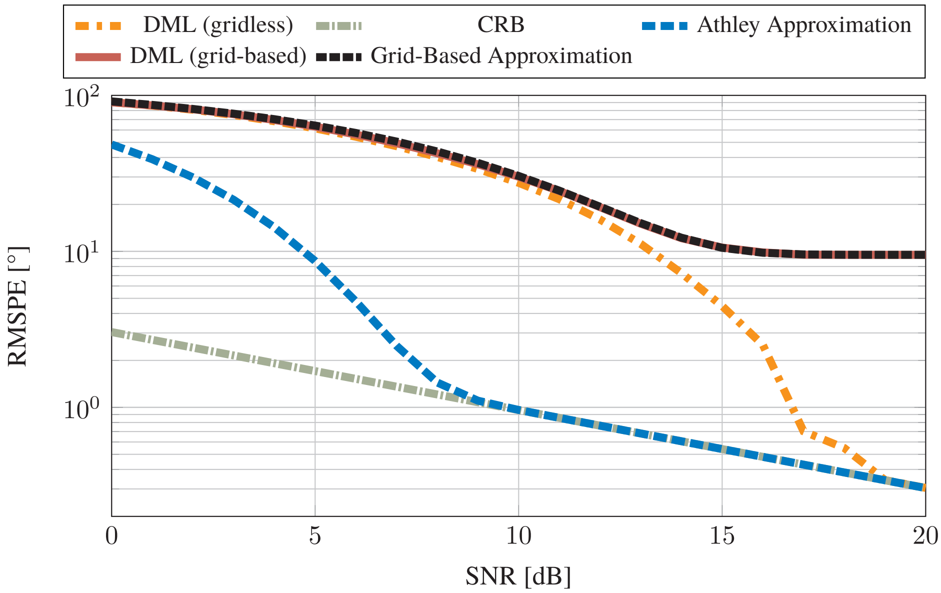


Figure 7.1: Approximated RMSPE vs. Simulated RMSPE, Single Source at $\theta^\star = 10.5°$, Constant Signal ($s = 1$), $N = 1$, 10^5 Monte Carlo Trials, $Q = 1$.

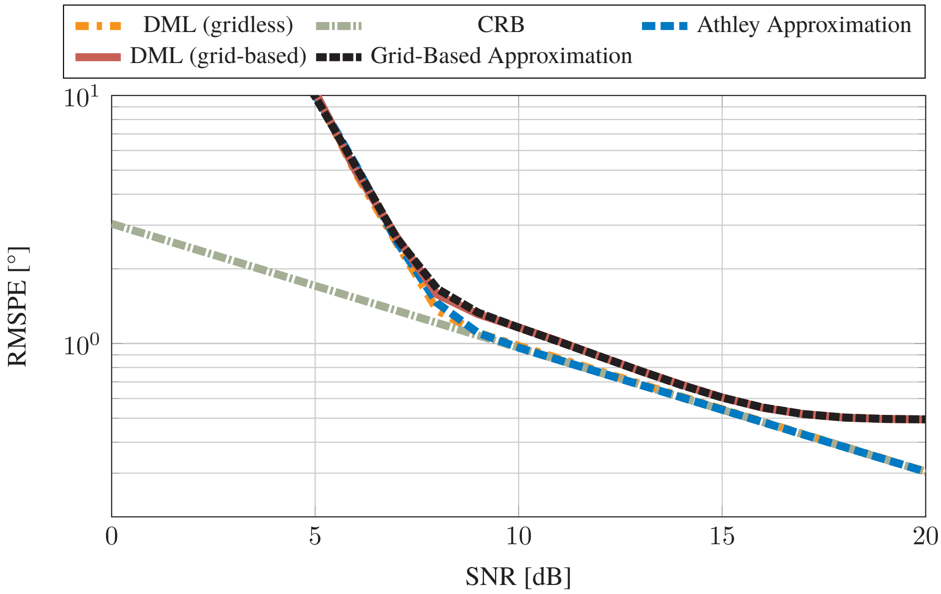


Figure 7.2: Approximated RMSPE vs. Simulated RMSPE, Single Source at $\theta^\star = 10.5°$, Constant Signal ($s = 1$), $N = 1$, 10^5 Monte Carlo Trials, $Q = 12$.

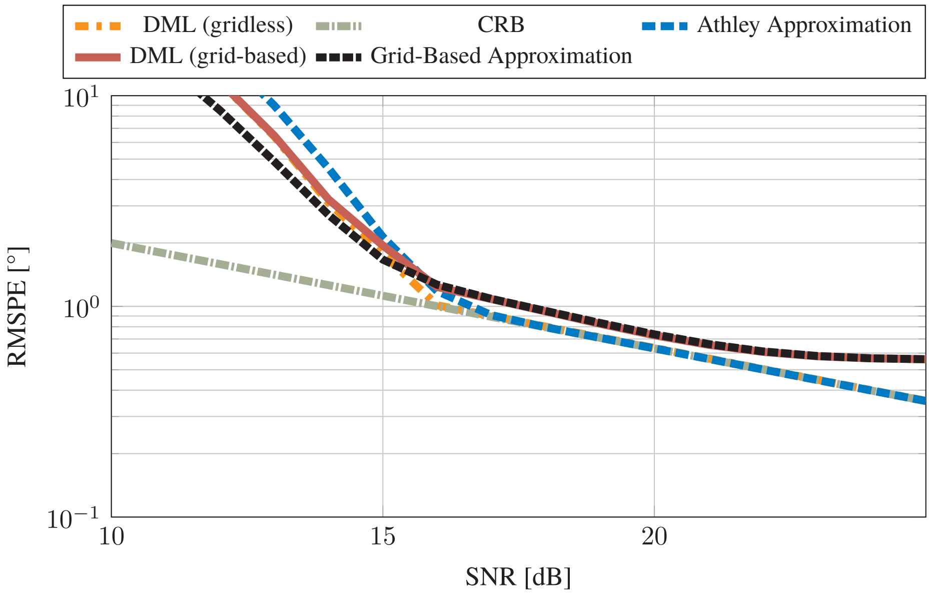


Figure 7.3: Approximated RMSPE vs. Simulated RMSPE, Two Sources at $\boldsymbol{\theta}^{\star} = [0.25°, 50.5°]^{\mathrm{T}}$, Constant Signal ($\boldsymbol{s} = [1, 1]^{\mathrm{T}}$), $N = 1$, 10^5 Monte Carlo Trials, $Q = 16$.

can happen that the selected mainlobe grid point $\check{\boldsymbol{\theta}}_0$ is no longer representative for the whole mainlobe. In order to clarify this, we may obtain a high outlier probability P_i^{out} for specific sidelobe i by computing the pairwise error probability between $\check{\boldsymbol{\theta}}_0$ and $\check{\boldsymbol{\theta}}_i^{\text{side}}$. However, if we compute the error probability between this sidelobe representative and the mainlobe grid-point with the second highest $\bar{U}(\boldsymbol{\theta})$, we may obtain a very low P_i^{out}. Then, some of the noise realizations, which lead to $U(\check{\boldsymbol{\theta}}_i^{\text{side}}) > U(\check{\boldsymbol{\theta}}_0)$, do not result in an estimate in the i-th sidelobe, but have the maximum of the corresponding log-likelihood $U(\boldsymbol{\theta})$ at the mainlobe grid point with the second highest asymptotic log-likelihood $\bar{U}(\boldsymbol{\theta})$. Such a scenario then leads to an overestimation of the outlier probability by the proposed grid-based MSE approximation.

7.2.2 Comparison of DoA Estimators

Now, we assess the performance of the DoA estimators presented in the previous chapters for fewer RF chains than sources. To this end, we performed Monte Carlo simulations with $L = 2$ equally powered sources. The DoAs for each realization are drawn from a uniform distribution between 0 and 2π. This enables a fair comparison of all the methods, in contrast to focusing on a fixed angle, where the results of the NN-based methods might be skewed by the training set realizations. Note that we do not enforce a minimal separation between the sources and we do not evaluate multiple noise and transmit signal realizations for each DoA. Therefore, we include the results of a *Genie ML* estimator as a benchmark for the achievable performance rather than comparing the algorithms with the CRB[1]. The Genie ML estimator is an ML estimator that is initialized with the true DoAs, instead of performing a grid search, and from there uses a gradient approach to find the closest local maximum of the likelihood function, which is does not coincide with the true DoAs due to noise.

In Fig. 7.4, we show the RMSPE averaged over 10^4 Monte Carlo runs for a selection of the discussed DoA estimation algorithms[2]. We can observe that the hybrid ChainNet approach attains the Genie ML performance. Second comes the hybrid MCENet. The DML estimator, which uses a reduced oversampling factor of 16 in this case, is able to attain the MCENet performance at high SNR. From the sparse recovery methods, the hybrid SPICE method achieves the best results, while the OMP based estimators ("OMP" and "CovOMP" for the system models (4.8) and (4.9), respectively) achieve the worst performance of all the presented approaches. However, apart from the hybrid ChainNet approach, none of the hybrid DoA estimators achieves the performance of the Genie ML benchmark. This means that the considered DoA estimators are affected by outliers. To confirm this conclusion, we take a look at the RMSPE of the best 95% of realizations for each method in Fig. 7.5. There, we see that apart from the OMP-based approaches[3], the hybrid DoA estimators achieve the Genie ML bound, i.e., the large difference between some of the hybrid DoA estimators and the Genie ML in Fig. 7.4 comes from less than 5% of the realizations. This means that the differences observed in Fig. 7.4 reflect the robustness of the different methods with respect to outliers.

To obtain a better comparison of the susceptibility to outliers of the different methods, we show a cutout of the empirical Cumulative Density Functions (CDFs) at 20 dB SNR in Fig. 7.6. Again, we can observe that the best performing method of Fig. 7.4, viz. the hybrid ChainNet estimator, does not experience any outliers at this SNR. Also, we see that the hybrid MCENet approach and the DML estimator have the least amount of outlier estimates of all the other estimators, which explains their good

[1] By pairing only one realization of the noise and transmit signals with each DoA, an immense number of Monte Carlo trials would be necessary to obtain a meaningful bound by the CRB. Furthermore, for closely spaced DoAs, the evaluation of the CRB becomes numerically instable, and also, the true value of the CRB may become so large that the periodicity of the angular domain must be considered (cf. [69]).

[2] The results of the missing DoA estimators can be found in Appendix A.

[3] The high susceptibility of the OMP methods to outliers is discussed in more detail in Appendix B

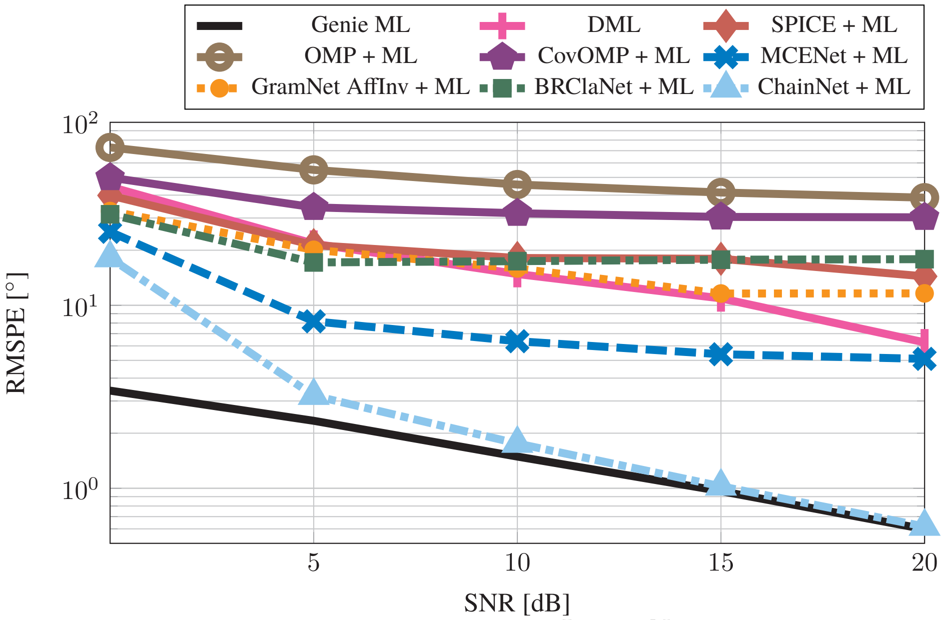


Figure 7.4: RMSPE vs. SNR, $L = 2$, $N = 10$.

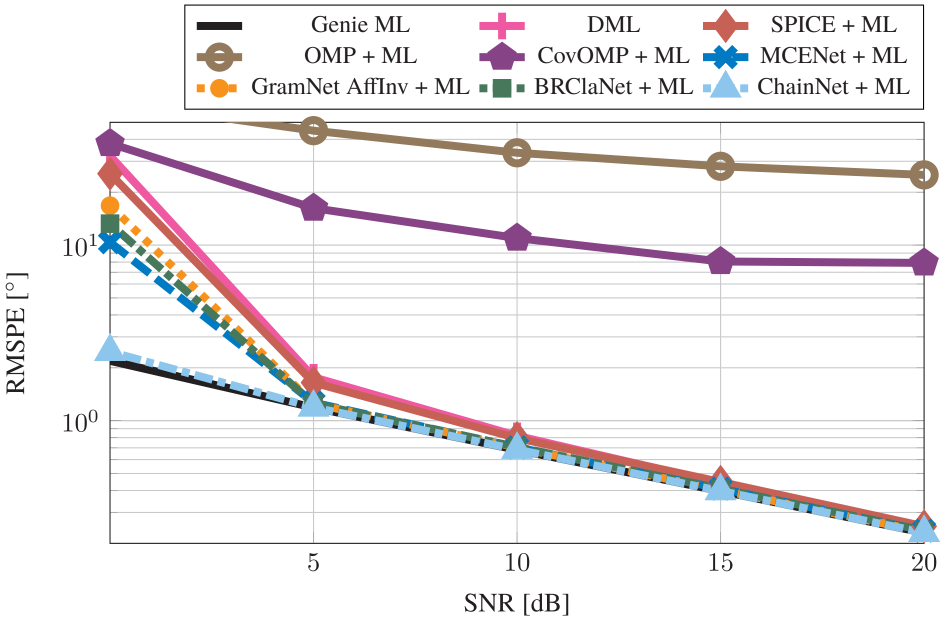


Figure 7.5: RMSPE vs. SNR, $L = 2$, Top 95% of Realizations, $N = 10$.

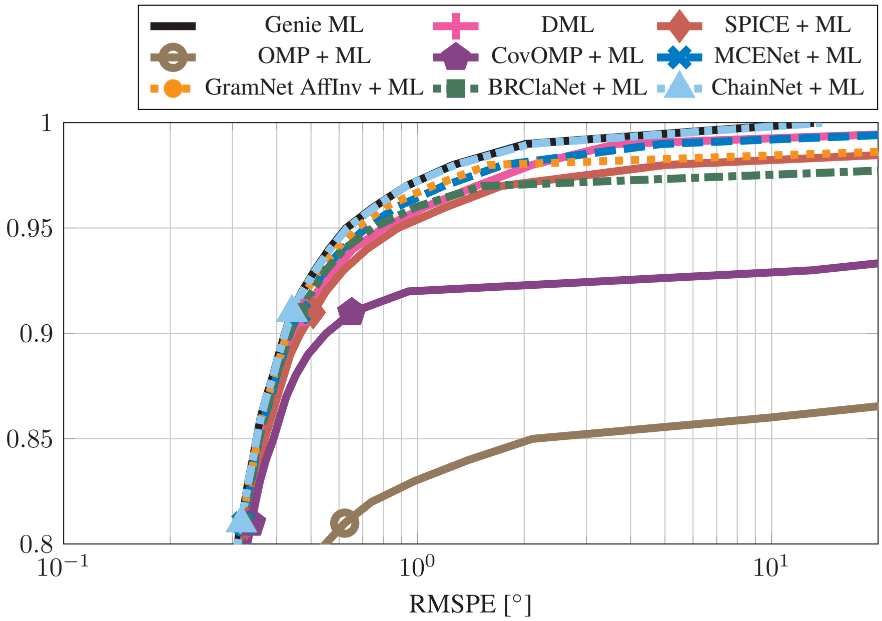


Figure 7.6: Empirical Cumulative Density Function, $L = 2$, SNR= 20 dB, $N = 10$.

performance in Fig. 7.4.

Interestingly, the realizations that lead to outliers in the estimates of the different methods are not the same. This can be observed in Fig. 7.4, where we filtered the realizations with a distance smaller than 10° from all the Monte Carlo runs. In that case, the hybrid GramNet approach, which has been trained on the affine invariant distance and applies a MUSIC estimator on the reconstructed full covariance matrix, and the DML estimator are now able to also achieve the Genie ML results. Their performance is apparently heavily affected by closely spaced sources. Also for the hybrid BRClaNet, we observe a significant improvement, as for closely spaced sources the peak detection in the NN output becomes an issue. On the contrary, introducing a minimum separation between the sources has no major impact on the hybrid MCENet results. The same applies to the hybrid SPICE method, however, the hybrid OMP estimators are benefitting from the introduced minimum distance (cf. Appendix B).

Apart from the ChainNet approach, the MCENet estimator showed a high robustness with respect to outliers. This robustness can be motivated by its training objective, which is the cost function of interest, i.e., minimizing the distance between the estimates and the true DoAs. As we have seen, outliers lead to a huge degradation. Therefore, the training objective lies first and foremost on outlier reduction. However, this

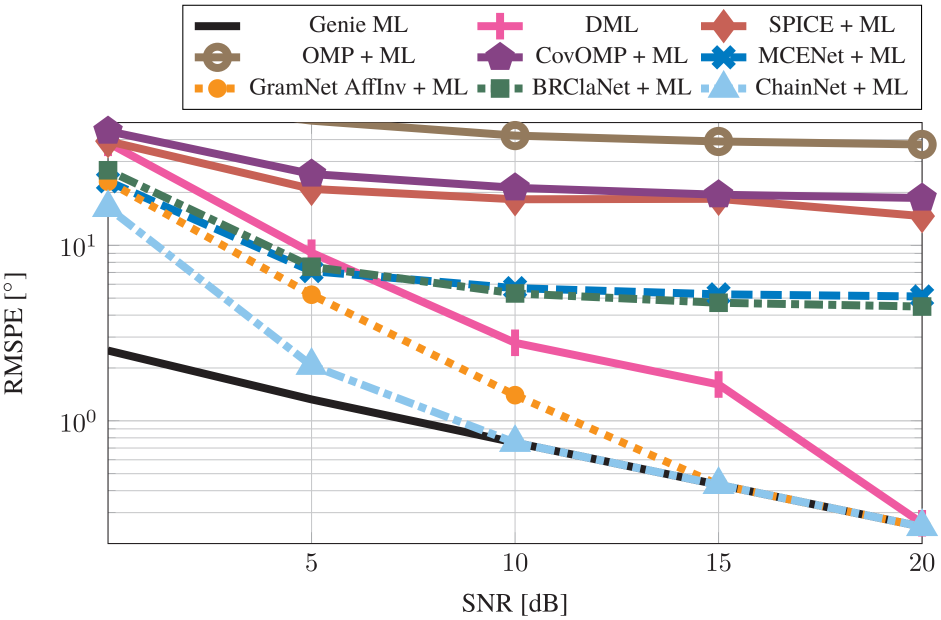


Figure 7.7: RMSPE vs. SNR, $L = 2$, 10° Gap Between Sources, $N = 10$.

comes at a cost, which can be seen in Fig. 7.8, where we show the results of the non-hybrid version of the algorithms. The MCENet approach provides robustness against outliers, but without subsequent gradient steps, its performance for the non-outlier realizations cannot compete with most of the other methods. In contrast, the ChainNet estimator does not suffer from a trade-off between robustness and high accuracy for all realizations. Additionally, the DML estimator and the non-hybrid GramNet approach yield a comparable accuracy as the ChainNet method for all realizations, and still feature a higher outlier robustness than SPICE, especially for sources that are not narrowly spaced.

7.3 Equal Number of Sources and RF Chains

For $L = 3$ sources, we are operating in the domain of $L \geq W$. Here, a direct evaluation of the ML estimators becomes practically infeasible. For this case, we plot the simulation results in Fig. 7.9 for 10^3 Monte Carlo trials with randomly drawn DoAs, similar to the simulations for $L = 2$. Now, with more sources, none of the presented algorithms achieves the performance of the Genie ML estimator. Furthermore, the hybrid ChainNet and MCENet approaches still provide the best RMSPE averaged over all DoA realizations. A method that we have not considered before is the

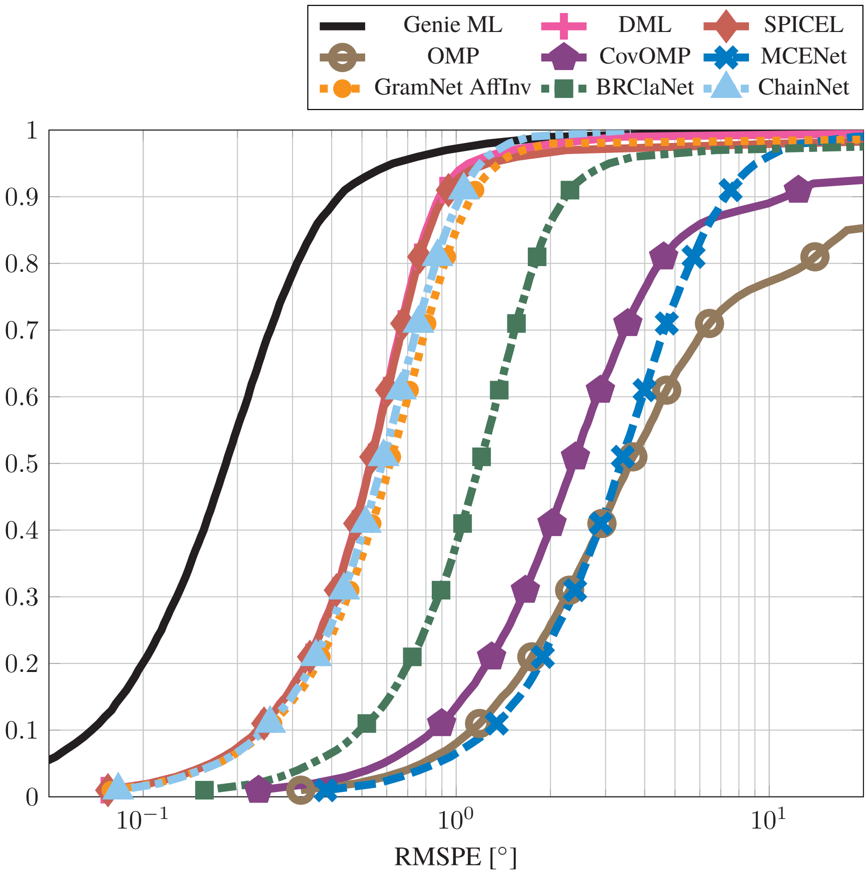


Figure 7.8: Empirical Cumulative Density Function, $L = 2$, SNR$= 20\,\text{dB}$, $N = 10$.

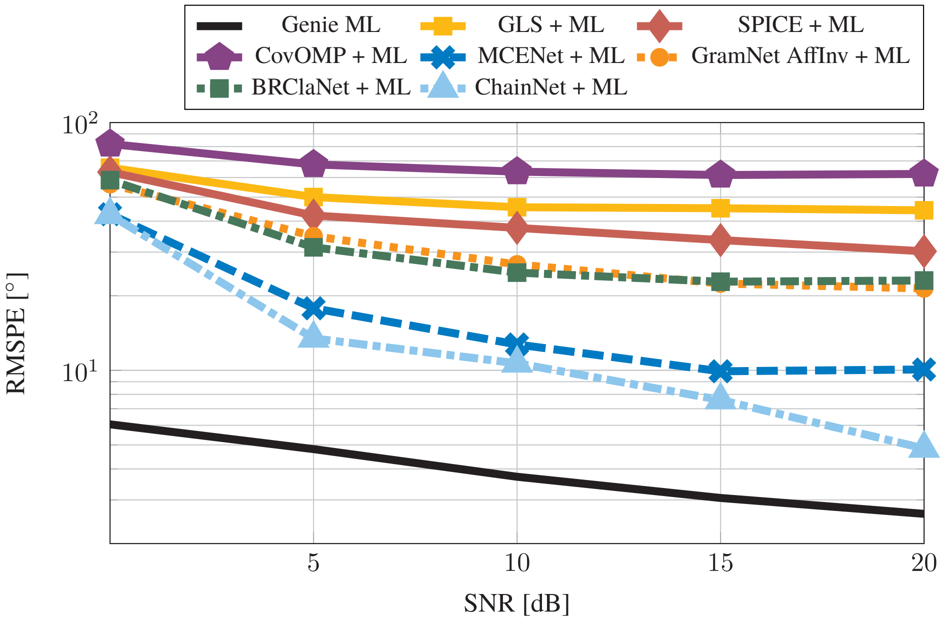


Figure 7.9: RMSPE vs. SNR, $L = 3$, $N = 10$.

GLS estimator[4]. Unfortunately the hybrid GLS estimator (operating on a grid with oversampling factor of 8) provides an unsatisfying performance, especially compared to the more cost efficient hybrid SPICE method, which is not based on a NN as well.

As for the two source case, we show the results for the top 90% of realizations in Fig. 7.10 and the performance for all realizations with a minimum gap of 10° between two sources in Fig. 7.11. Again, we can observe in Fig. 7.10 that the machine learning-based methods achieve the Genie ML performance for 90% of the realizations. However, removing the 10% worst estimates is not enough for the classical methods to prevent all of the outliers. Introducing a 10° gap between sources, also yields similar results as for the two source case. Fig. 7.11 shows a significant improvement for the hybrid GramNet and BRClaNet approaches, which use a peak detection algorithm, and are able to draw even with the hybrid ChainNet and MCENet estimators for the remaining realizations. Yet, this gap is not enough for the GramNet to attain the Genie ML results, as has been the case for $L = 2$.

Again, we can observe that the MCENet estimator is only viable with a subsequent gradient approach on the likelihood function in Fig. 7.12 and Fig. 7.13, which show the empirical CDF for the hybrid and non-hybrid DoA estimators. In contrast, the

[4] For $L < W$, the DML estimator provides a better alternative, with the same computational complexity.

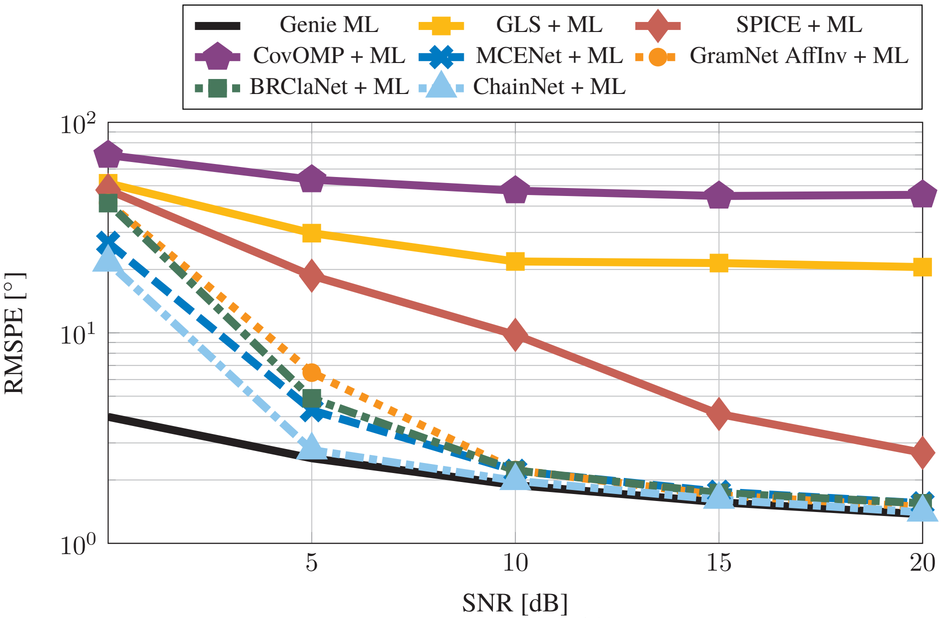


Figure 7.10: RMSPE vs. SNR, $L = 3$, Top 90% of Realizations, $N = 10$.

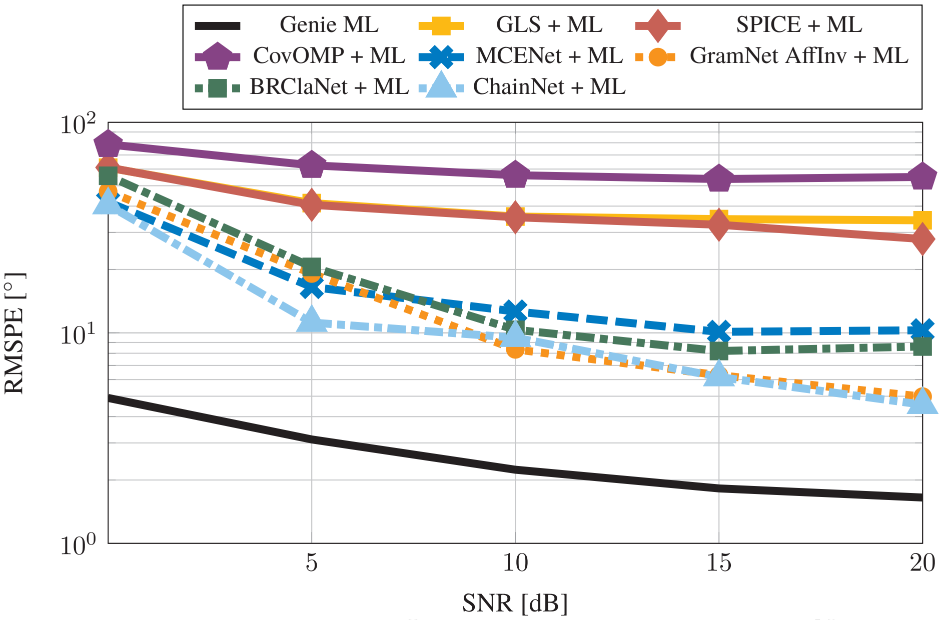


Figure 7.11: RMSPE vs. SNR, $L = 3$, $10°$ Gap Between Sources, $N = 10$.

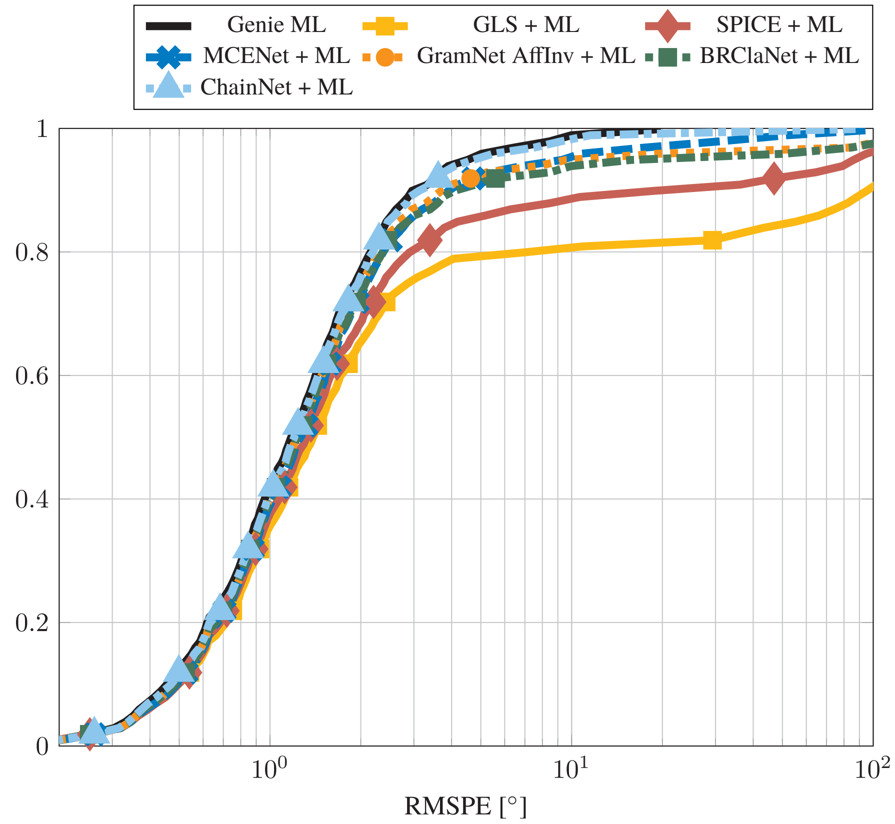


Figure 7.12: Empirical Cumulative Density Function of the Hybrid Estimators, $L = 3$, SNR$=$ 20 dB, $N = 10$, $M = 9$.

ChainNet and GramNet approaches yield a very good estimation performance even in their non-hybrid forms, while suffering from fewer outliers than the SPICE estimator.

What we have not discussed yet, is how the different algorithms behave for a varying number of snapshots. To that end, we first take a look at Fig. 7.14, which shows the empirical CDFs for $N = 1000$ snapshots as solid lines. The CDF plot for $N = 10$, shown in Fig. 7.12, has been added to Fig. 7.14 as dashed lines. From the CDFs for $N = 1000$, we can see that more snapshots improve not only the overall achievable RMSPE (shift to the left), but we have fewer outliers than for $N = 10$. Especially, the SPICE method profits heavily from the increased number of observations. This behavior of the SPICE estimator is not surprising, as it is based on a covariance-matching criterion, similar to the GLS estimator, which has been proven to be a consistent estimator (for a sufficiently dense grid) [10]. For a high

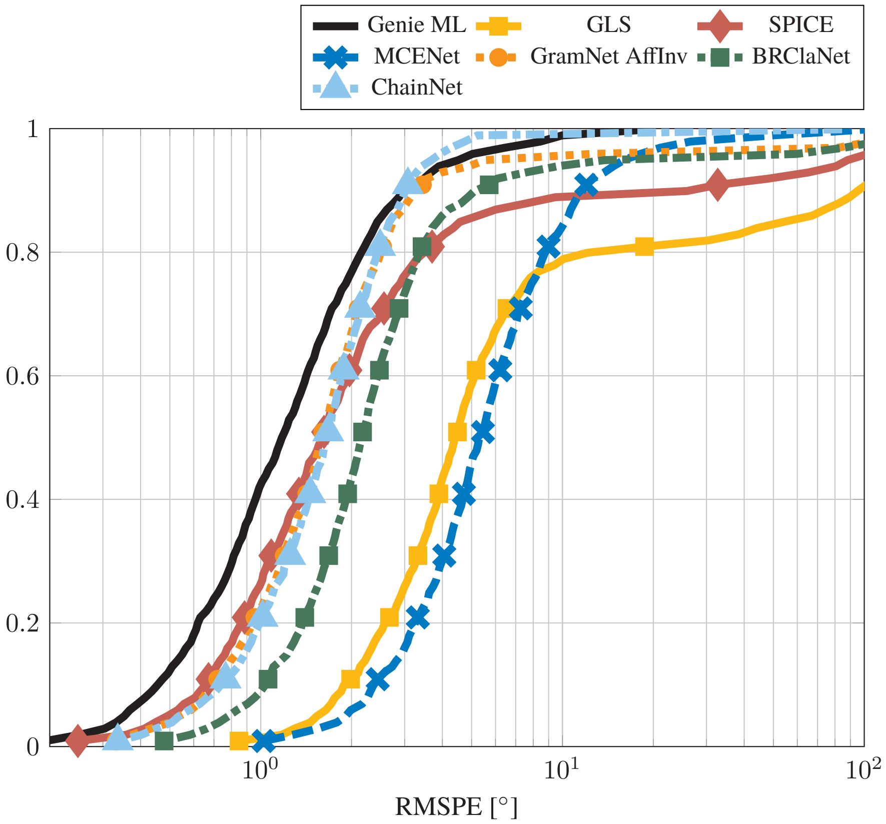


Figure 7.13: Empirical Cumulative Density Function, $L = 3$, SNR= 20 dB, $N = 10$, $M = 9$.

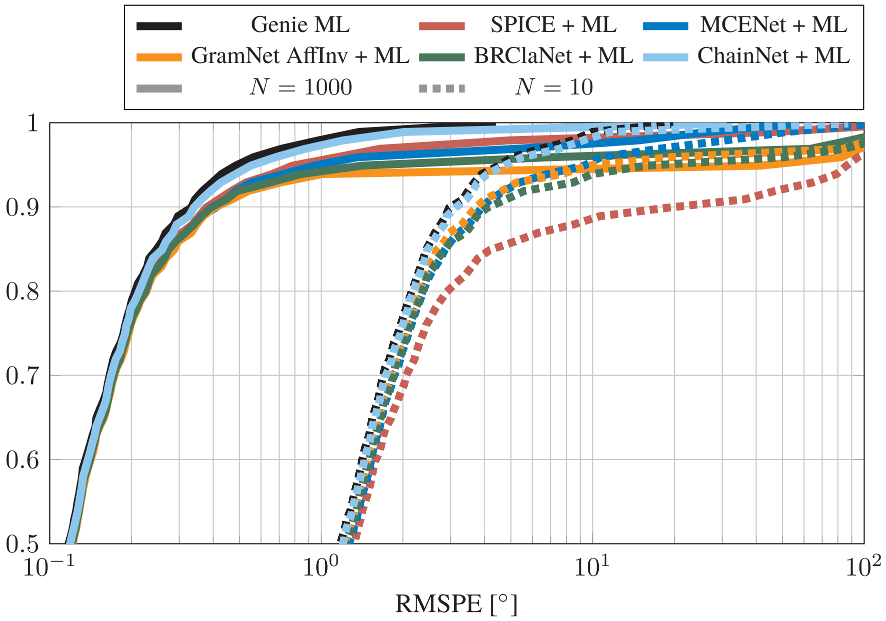


Figure 7.14: Comparison of Empirical Cumulative Density Functions for Varying N, $L = 3$, SNR= 20 dB, $M = 9$.

number of snapshots N, the sample covariance matrices are consistent estimates of the true subarray covariance matrices, which again justifies the validity of the covariance-matching objective.

In a similar fashion to Fig. 7.14, the CDFs for a differing number of antenna elements are compared in Fig. 7.15. The plot shows the results for an antenna array with $M = 25$ antennas and $W = 3$ RF chains in comparison to the previously considered 9-element antenna array. Again, we observe an improvement of the overall RMSPE and a decline in the number of outliers. However, 25 antennas are not enough to fully close the gap between the SPICE algorithm and the NN-based methods. Note that with the number of antennas, the number of subarrays that need to be sampled increases as well. Depending on the time that is required to switch between two subarray constellations, a trade-off between more antenna elements and an increased number of snapshots is necessary.

Last but not least, we take a brief look at the computational complexity of the presented estimators. To this end, we provide computation times of the MCENet, ChainNet, GramNet, GLS and SPICE algorithms in Table 7.3. Naturally, computation times do not achieve the same validity as a rigorous complexity analysis in Landau

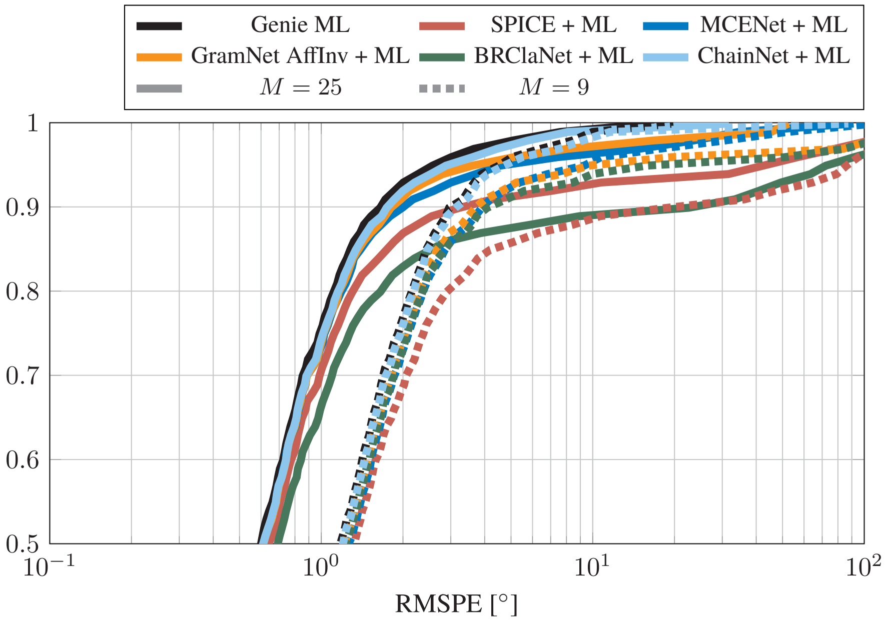


Figure 7.15: Comparison of Empirical Cumulative Density Functions for Varying M, $L = 3$, SNR$= 20\,\mathrm{dB}$, $N = 10$.

notation, due to their dependence on the used hardware and implementation, however, they may still yield some qualitative insights. The figures given in Table 7.3 denote the times to compute estimates for 1000 realizations in MATLAB on a simulation server equipped with two Intel Xeon Gold 6134 processors. Apart from the necessary computation time for the hybrid approaches, we also show results for the evaluation of each algorithm without consecutive gradient steps. From our simulations, we see that without the consecutive gradient approach, the required time for the MCENet inference steps is about one tenth of the evaluation time of the "SPICE" estimator[5]. The ChainNet evaluation takes roughly the time of three MCENet inferences, as expected. A similar time is required for the GramNet estimator, due to the additional MUSIC algorithm that is applied after the NN evaluation, but it is still significantly faster than "SPICE". In comparison, the iterative solution of the SPICE method "SPICE iter." with a fixed iteration count of 10^4 iterations[6] and the GLS estimator, whose complexity grows exponentially with the number of sources, are again about a factor

[5] Here, the "SPICE" implementation uses YALMIP [93] for modeling the optimization problem, which is then solved by the MOSEK solver [94].

[6] In fact, 10^4 iterations is still not enough to obtain the same solution as the general purpose solver (cf. [3]).

	MCENet	ChainNet	GramNet	SPICE	SPICE iter.	GLS
non-hybrid	5.4 s	15.8 s	15.3 s	127.2 s	1002.8 s	1486.2 s
hybrid	52.3 s	61.8 s	62.5 s	174.7 s	1045.3 s	1697.9 s

Table 7.3: Computation Times of DoA Estimators

10 slower than the "SPICE" solution. For the hybrid versions of these algorithms, we see that these steps, which have been implemented by a block coordinate ascent, take roughly the same time for the hybrid MCENet, ChainNet, GramNet, and SPICE approaches[7]. In contrast, for the hybrid GLS approach the gradient steps take much longer to converge, due to the numerous poor initial estimates provided by the GLS estimator. Note that the computation times for BRClaNet are not listed here, but are expected to be close to the MCENet times, as the additional effort for the peak search is negligible.

[7] Note that the required time for the gradient steps heavily depends on the target accuracy. A looser stopping criterion may significantly reduce the required computation times. For the presented simulations, the stopping criterion for the gradient steps is very tight ($< 10^{-6}$ absolute change in the log-likelihood). This high accuracy is necessary to achieve meaningful results for the information criteria discussed in the next chapter.

Model Order Selection 8

Up to this point, we have assumed that we know the number of sources L. Knowing the correct number of sources is essential for obtaining accurate DoA estimates. However, in general, this a priori knowledge is not available. Instead, L has to be estimated from the same observations that the DoA estimation is based on. This problem is called model order selection and will be the subject of this chapter.

Model order selection problems do not only arise for DoA estimation, but in a plethora of different applications. Two excellent overview articles that summarize the most common techniques for model order selection are [95] and [96]. In [95], the focus lies on so called information criteria, which are arguably the most well known and broadly used methods for this task. We will discuss information criteria in more detail in Section 8.2. Apart from information criteria, the authors of [96] present further parametric and non-parametric model order selection techniques, such as Bayes factors and cross validation, and discuss common pitfalls and misconceptions in this field.

In the context of DoA estimation, information criteria have been the method of choice over the last decades. The most famous work on model order selection for DoA estimation problems is probably [18]. There, the authors derive a formulation of the information criteria for the SML case that only depends on the eigenvalues of the sample covariance matrix. The major benefit of the closed form solutions for the information criteria derived therein is that they work without a direct computation of the maximum likelihood estimates of the DoAs, which are computationally expensive. In [97], it has been shown that this estimator is consistent, i.e., the probability of a misclassification goes to zero for increasing N.

Employing NNs for model order selection has been first proposed in [98, 67] for differentiating between no source, one source, or two sources. To this end, the authors train the NN on an MSE cost function, where each class is mapped to a point in the two dimensional plane, similar to a modulation scheme in communications. Recently, the model order selection problem for fully sampled arrays has been revisited for state of the art feedforward NNs in [65, 99, 15].

In the context of subarray sampling, the formulation of the information criteria presented in [18] is no longer applicable. Instead, the authors of [14] propose to simply replace the ML estimates of the DoAs, which are usually needed for the evaluation of information criteria, by GLS estimates. However, as we have seen in Chapter 7, the GLS estimator suffers heavily from outliers, while having a high computational complexity. Therefore, in [3], utilizing hybrid DoA estimators for this task has been investigated. Moreover, an extension of the NN approach to model order selection [15] to systems with subarray sampling, which provides significant improvements in the selection accuracy compared to information criteria based methods, has been discussed in [3].

This chapter is structured as follows. First, we will formulate the model order selection problem and discuss the theoretically optimal MAP estimator. Then, we will take a look at information criteria and how they can be applied to systems with subarray sampling. Finally, we discuss an end-to-end NN-based model order selection approach[1] and a method that uses the covariance reconstruction technique presented in Section 5.4.

8.1 Problem Formulation and MAP Estimator

The goal of model order selection is to select the model with the "best predictive power" $\hat{p}$ from a set of model candidates p_ℓ, with model order ℓ in the set $\mathcal{L}$, based on a finite number of observations [96]. Here, the predictive power is measured by the out-sample prediction loss given by [96]

$$\mathrm{E}\left[f(\boldsymbol{Y};\hat{p})\right], \tag{8.1}$$

where $\boldsymbol{Y}$ follows the distribution of the true data-generating model and f is a loss function that measures the goodness of fit between the samples $\boldsymbol{Y}$ and the model $\hat{p}$, e.g., the negative log-likelihood. In the DoA estimation scenario, the model selection problem fits to the parametric framework [96], i.e., the true model is included in the set of model candidates, since we assume perfect knowledge of the array manifold. In that case, the true data-generating model generally is the best model. This means that we try to find the true number of sources L from the set of all possible model orders $\mathcal{L} = \{0, 1, \dots, L_{\text{max}}\}$.

In theory, the estimator that maximizes the probability to select the true model order $L \in \mathcal{L}$ is the MAP estimator, as has been proven in [95]. The MAP selection

[1] The extension of the information criteria to systems with subarray sampling and the model order selection with NNs has been recently published in [3]. Throughout this chapter, we closely follow our presentation therein.

rule reads as [95]

$$\begin{aligned}\hat{L} &= \operatorname*{argmax}_{\ell \in \mathcal{L}} p(\mathcal{H}_\ell | \boldsymbol{Y}) \\ &= \operatorname*{argmax}_{\ell \in \mathcal{L}} p(\boldsymbol{Y} | \mathcal{H}_\ell) p(\mathcal{H}_\ell),\end{aligned} \tag{8.2}$$

where $\mathcal{H}_\ell$ denotes the hypothesis that the true model order is ℓ. However, in general, the MAP estimator cannot be evaluated as the true prior $p(\mathcal{H}_\ell)$ is generally unknown, and more importantly, the probability

$$p(\boldsymbol{Y}|\mathcal{H}_\ell) = \int_{\mathcal{S}_\ell} p_\ell(\boldsymbol{Y}|\boldsymbol{\varphi}_\ell) p_\ell(\boldsymbol{\varphi}_\ell)\, \mathrm{d}\boldsymbol{\varphi}_\ell, \tag{8.3}$$

with the parameters of the ℓ-th model $\boldsymbol{\varphi}_\ell$, which lie in the space $\mathcal{S}_\ell$, and their prior $p_\ell(\boldsymbol{\varphi}_\ell)$, cannot be computed efficiently.

8.2 Information Criteria

Since MAP estimation is generally not available, several other techniques have been developed in the past. Among these model order selection methods, there exists a group of approaches denoted by information criteria (see [95] for details), which all feature a common structure. They all base their model order estimate on the sum of the negative log-likelihood function of the observations at the respective ML estimates of the parameters $\hat{\boldsymbol{\varphi}}_{\mathrm{ML},\ell}$ and a penalty term $c(\ell)$, which combats overfitting of the model order. Therefore, the model order estimate $\hat{L}$ can be written as

$$\hat{L} = \operatorname*{argmin}_{\ell \in \mathcal{L}} -\ln\left(p_\ell\left(\boldsymbol{Y}; \hat{\boldsymbol{\varphi}}_{\mathrm{ML},\ell}\right)\right) + c(\ell). \tag{8.4}$$

Note that the importance of the penalty term can be seen from the stochastic models (2.10) and (2.13). In both cases, for varying model order L, we have a family of nested stochastic models, i.e., the stochastic model for a certain number of sources L includes the models with model order smaller than L for a special choice of the parameters $\boldsymbol{\varphi}$. Therefore, the log-likelihood of $\boldsymbol{Y}$ is monotonically increasing in the number of sources. Hence, for $c(\ell) = 0, \ell \in \mathcal{L}$, we would always obtain the maximum model order L_{max} from (8.4).

As mentioned above, for fully sampled arrays and under the SML system model, an analytic expression for the value of the log-likelihood depending on the eigenvalues of the sample covariance matrix has been derived in [18]. The idea is to use a reparameterization of the received signal covariance matrix by its eigenvalues and eigenvectors, instead of the DoAs and signal covariance matrix. Then, it can be shown that the value of the log-likelihood function that is parameterized by the ML estimates of the eigenvalues $\hat{\boldsymbol{\psi}}$, eigenvectors and the noise power σ_η^2 reads as [18]

$$\ln\left(p_\ell\left(\boldsymbol{Y};\hat{\boldsymbol{\varphi}}_{\mathrm{ML},\ell}\right)\right)=N\left[\ln\left(\prod_{i=\ell+1}^{M}\hat{\psi}_i\right)-(M-\ell)\ln\left(\frac{1}{M-\ell}\sum_{i=\ell+1}^{M}\hat{\psi}_i\right)\right]. \tag{8.5}$$

The huge benefit of this method is that compared to computing ML estimates of the DoAs for each model order $\ell\in\mathcal{L}$, which has a demanding complexity, only the eigenvalue decomposition of the sample covariance matrix has to be computed once. Additionally, an advantage of this parameterization is that it works without any knowledge on the array manifold, i.e., the method works without any array calibration.

Unfortunately, this reparameterization does not work for systems with subarray sampling. There, the likelihood function depends on multiple subarray covariance matrices each of dimension $W\times W$ with W potentially smaller than L_{max}. Evaluating the ML estimators for each model order provides no feasible solution as well, as the complexity becomes quickly intractable for $L_{\text{max}}>2$. Therefore, replacing the ML estimates of the parameters with the GLS estimates has been proposed in [14]. As this might be a viable option for a very large number of snapshots, where the GLS estimator provides consistent estimates, we observed that the GLS estimator fails to produce reasonable estimates in the low snapshot domain. Additionally, the complexity of the GLS estimators also scales poorly with the maximum number of sources L_{max}. Therefore, we propose to replace the ML estimates in (8.4) by one of the better performing hybrid estimators presented in the previous chapters. For example, the SPICE estimator combined with subsequent gradient steps can be used to this end, and will be used as a benchmark for the NN-based approaches presented in the next section.

8.3 Machine Learning Approach

8.3.1 End-to-End Classification

Aiming to select the true model order out of a finite set of possible model orders is a multiclass classification problem. In this section, we discuss a NN-based solution to this classification problem for systems with subarray sampling, which has been published under the name *CovNet* in [3].

Similar to the NNs that have been presented in Chapter 5, we utilize a fully connected, feedforward NN with N_{h} hidden layers with a ReLU activation function. As input data $\boldsymbol{x}$, we again use the real parameters of artificially generated subarray sample covariance matrices with random DoAs, transmit powers and SNR (see Section 5.1). In contrast to the data generation for the DoA estimation task, we vary the number of sources in our input data between 0 and L_{max} and label each input data sample with the correct model order of the respective data-generating model. Thereby, we generate an equal amount of data samples for each model order in $\mathcal{L}$. The output layer consists of $L_{\text{max}} + 1$ neurons followed by a softmax activation. For the training, we use the cross-entropy loss function defined in (5.4), such that the output values of the neural network $z(\ell|\boldsymbol{x};\boldsymbol{w})$ can be interpreted as estimates of the posterior density $p(\mathcal{H}_\ell|\boldsymbol{Y})$ [72]. In that sense, the NN forms a data-based approximation of the MAP estimator (8.2) rather than being based on some information criterion.

In addition to the aforementioned training for a perfectly known, fixed array manifold, we are also interested in how we can adapt the NN to array imperfections. This problem has already been discussed for the fully sampled case in [15] for model order selection, and in [100] for DoA estimation. There, the idea is to train the NN based on a theoretic model for the array manifold with a large amount of artificial data. Then, after deployment, we use an online learning procedure to adapt the NN to the array imperfections by utilizing only a limited number of calibration measurements[2]. This method showed promising results and can be easily applied to the subarray sampling case as well. The results for this scenario can be found in the following chapter and show that this approach might be a viable alternative to training from scratch with the knowledge of the calibrated array manifold, if enough labeled data for the online training is available.

Another interesting question is if it is possible to utilize a single network over a broad frequency range for a frequency scanning direction finder, which are used in frequency monitoring and surveillance applications, for example. Although the array geometry remains fixed, the frequency of interest, and therefore, the array manifold changes rapidly. Here, utilizing a NN-based approach is particularly interesting, due to the very short time required for a network inference in comparison to an eigenvalue decomposition or DoA estimation for all model orders. However, storing an individual network for each frequency is infeasible, as it would require an immense amount of memory. Instead, we propose to train a single network for multiple frequencies. To that end, we draw the frequency for the data-generating received signal model uniformly from a range $[f_{\text{min}}, f_{\text{max}}]$ during the training. Again, the results for this approach that are presented in the next chapter show that this is indeed a feasible solution.

[2] For details on array calibration we refer the reader to [33].

8.3.2 Covariance Matrix Reconstruction

An alternative method to the direct solution of the classification problem by a NN, uses the covariance reconstruction technique presented in Section 5.4. There, we trained a NN that estimates the full covariance matrix from the subarray sample covariance matrices for a fixed model order L. Now, we train a similar NN, however, for the training data, we use data samples with varying model order as described for the classification approach above. The respective label per data sample is again the true data-generating covariance matrix for the fully sampled array. Apart from the training data, no changes compared to the case with fixed model order are necessary, as the architecture of the GramNet NNs does not directly exploit the knowledge about the number of sources for the case of a fixed model order.

From the reconstructed full covariance matrix, we estimate the model order with some information criterion. To that end, we directly utilize the expression for the log-likelihood function (8.5) that depends only on the eigenvalues of the reconstructed full covariance matrix. Note that this procedure is a heuristic. In fact, we cannot give an expression for the likelihood function of the observations parameterized by the eigenvalues of the estimated full covariance matrix. Therefore, we do not even know how the log-likelihood scales in the number of observations. To obtain the best accuracy, should we use the factor N as above or KN or something else completely? Due to the lack of a fitting stochastic model for the full covariance estimates[3], this question cannot be readily answered by a theoretical argument. Instead, a cross-validation approach can be used to determine the best prefactor in (8.5) and penalty term.

In terms of complexity, the GramNet approach adds an eigenvalue decomposition compared to the end-to-end approach presented above. However, for a subsequent DoA estimation by means of a MUSIC estimator based on the reconstructed covariance matrix, the eigenvalue decomposition is necessary anyway.

[3] At first thought one might suggest to approximate the PDF by a Wishart distribution. Then, however, the problem still remains that it is unclear how many degrees of freedom we should choose for said Wishart distribution.

9 Simulation Results – Model Order Selection

Here, we present simulation results of the model order selection problem for systems with subarray sampling. The chapter is again split into two parts. First, we explain the simulation parameters that we used in our simulations. Then, we discuss the results of the Monte Carlo simulations.

9.1 Simulation Setting and Neural Network Parameters

As for the DoA simulations, we consider a 9-element UCA with steering vectors given in (7.1). The subarray sampling scheme with the $K = 4$ constellations is given in Table 7.1, i.e., the system uses $W = 3$ RF chains. The maximum number of sources that are transmitting simultaneously is $L_{\text{max}} = 3$. Hence, we are operating in the critical region, where the number of sources is not always smaller than the number of RF chains.

The test sets, on which we evaluate the different model order selection methods, consist of an equal amount of data samples from each model order. The distribution of the DoAs, transmit powers, and SNR follow that of the training set described at the beginning of Section 5.1. Thereby, the respective parameters of the distributions can be found in Table 9.1, which are also used for the training of the CovNet NNs. Additionally, Table 9.1 summarizes the architecture used for CovNet. Compared to the NNs that are used for the DoA estimation, the CovNet NNs are smaller[1]. In contrast, the GramNet network trained with a varying model order, which we use for the full covariance matrix reconstruction, uses the same parameters as the other DoA estimation networks (cf. Table 7.2). For the cost function of the GramNet, we used the affine invariant distance. The information criterion, which we employ in combination with the GramNet reconstruction, is the Minimum Description Length

[1] We use the same architecture as for the CovNet NNs that have been proposed for the fully sampled case in [15].

Parameter	Value
$\sigma_{s,\text{min}}$	$-9\,\text{dB}$
$\sigma_{\eta,\text{min}}$	$-10\,\text{dB}$
$\sigma_{\eta,\text{max}}$	$30\,\text{dB}$
N_h	3
N_u	1024
Weight Initialization	Glorot [91]
Batch Size	64
Optimizer	Adam [92]
Learning Rate	10^{-4}
Samples per Training Set	$64 \cdot 10^6$

Table 9.1: CovNet Parameters

(MDL) criterion[2]. Its respective penalty term in (8.4) is given as

$$c(\ell) = \frac{2\ell + 1}{2} \ln\left(KN\right). \tag{9.1}$$

As the prefactor for the log-likelihood we choose N, like in (8.5), since this choice provided better results than KN. However, we want to emphasize again that the choice of the prefactor and penalty term does not come naturally, due to the heuristic character of GramNet model order selection approach. These parameters have not been optimized by a cross-validation approach, for example, but we selected a combination that provides acceptable results for our simulations. In that sense, the presented results for the GramNet approach should be understood more as a proof of concept.

As a reference for the NN-based approaches, we use an MDL estimator that obtains its parameter estimates for the individual model orders by evaluating a hybrid SPICE estimator. The oversampling factor for this SPICE estimator has been chosen as $Q = 32$, as for the DoA estimation simulations in Chapter 7.

9.2 Simulation Results

In Fig. 9.1, we look at the accuracy of the different model order selection techniques for $N = 10$ snapshots at different SNR. To this end, we evaluated the different methods on test sets with a fixed SNR, each consisting of $4 \cdot 10^3$ data samples. Note that in this case

[2] The MDL criterion is the same as the Bayesian Information Criterion (BIC). Depending on which rationale leads to the common penalty term, one or the other name is preferred.

fixed SNR means that the noise power σ_η^2, and therefore, the ratio between the transmit power of the strongest source and noise, is fixed. However, the transmit power of the other sources still varies. From Fig. 9.1, we can see that the CovNet approach clearly outperforms the MDL criterion, which uses the hybrid SPICE estimates, over the whole SNR range. This can be explained by the insufficiency of the hybrid SPICE estimator to produce good estimates for all realizations (cf. Chapter 7), which heavily impacts the performance of any information criterion. The GramNet approach trained with varying L, denoted by "GramNet $L \in \mathcal{L}$", is able to surpass the accuracy of CovNet in the high SNR regime. However, we observe a substantial performance degradation at low SNR. There, the GramNet approach is prone to underfitting the model order[3]. In this region, the estimates of the eigenvalues are dominated by noise, which leads to a poor selection accuracy. A similar performance at low SNR can be observed if the GramNet NN has been trained on data with the maximum model order, referred to by "GramNet $L = 3$". This approach, however, sees a performance degradation in the high SNR regime, due to some overfitting of the model order. Intuitively, this behavior can be explained by the fact that this GramNet implementation has been trained to match the covariance matrices with a signal subspace of dimension $L = 3$. For high SNR scenarios, its training data always featured three eigenvalues that are well above the noise power, and the network tries to replicate this even for input data stemming from fewer sources.

The results of the CovNet and the MDL criterion with SPICE estimates for different N is shown in Fig. 9.2[4]. For each N, the test sets consist of $4 \cdot 10^3$ realizations, where the SNR for each realization has been drawn from a uniform distribution between $-10\,\text{dB}$ and $30\,\text{dB}$. Again, we see that the CovNet approach is superior to the MDL estimator that is based on SPICE, for the same reasons as described above. Since we use sample covariance matrix information as the input data, we are able to pass data from a system model with $N \neq N_{\text{train}}$ to a network, which has been trained on data consisting of N_{train} snapshots. In Fig. 9.2, we added the results for such a scenario, where we use a network trained with $N_{\text{train}} = 10$ to classify the input data with varying $N \in [3, 20]$. This approach is able to achieve almost the same performance as the CovNet, which has been trained with data matching the number of snapshots in the test set, i.e., $N = N_{\text{train}}$. A possible explanation for this kind of robustness towards

[3] Again, note that the choice of the prefactor and penalty term has not been heavily optimized. To alleviate this problem a different information criterion, such as the Akaike Information Criterion (AIC) with a smaller penalty term, can be used. However, in general, this may impact the performance at a higher number of snapshots.

[4] Throughout the remainder of this section, we will focus on these two methods, since their respective results are not subject to uncertainties in the choice of parameters, as is the case for the GramNet approach.

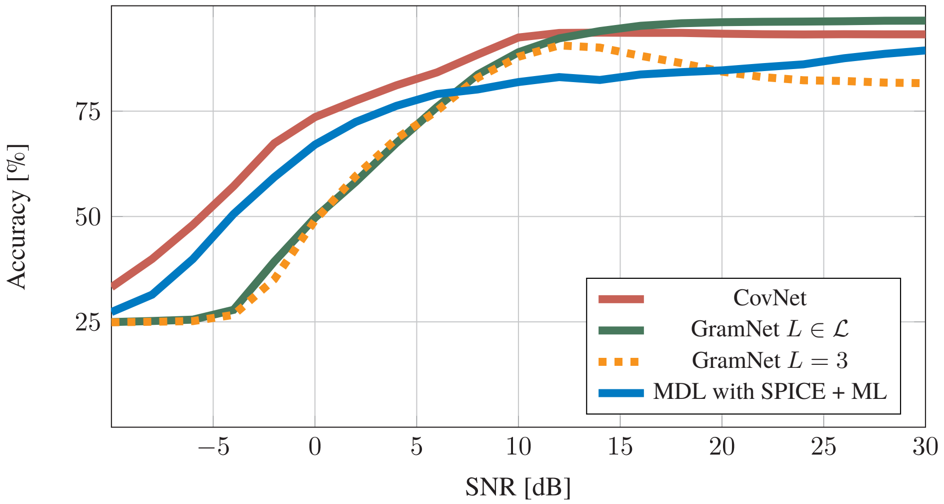


Figure 9.1: Model Order Selection Accuracy vs. SNR, $N = 10$

the number of snapshots lies in the effect of the number of snapshots on the input data. For the subarray sample covariance matrices a change in the number of snapshots translates to a change in the effective SNR, for which we have chosen a broad range during training. In fact, this robustness property is pretty important for a deployment in a direction finder that has to be able to cope with a differing number of snapshots. For such a system, we do not have to store an individual network for each N, but a single NN can cover multiple N.

Similarly to the robustness with respect to N, we are also interested in the performance of a single network for covering a certain frequency range. To this end, we provide simulation results in Fig. 9.3. There, we investigate the performance of a CovNet NN that has been trained on a single frequency $R/\lambda = 1$ and a CovNet approach trained on data samples with a randomly drawn frequency per sample. We see the accuracy for the different CovNet variants over the frequency, which is given by the ratio of radius R and the frequency dependent wavelength λ[5]. For each frequency point, we evaluated the different methods on a test set containing $4 \cdot 10^3$ data samples with the respective frequency. As a reference, we added the performance of several CovNet NNs that have been trained on the same frequency as is used in the respective

[5] This ratio is a common design parameter for direction finders. Note that this ratio of aperture and wavelength is found in the UCA steering vectors in (7.1) and needs to lie in a certain range to ensure identifiability of the DoAs. The considered range in R/λ can be easily translated to a frequency band, by fixing the radius R, e.g., the investigated range from 0.5 to 2 reflects a frequency band between 150 MHz and 600 MHz for a radius of $R = 1$ m.

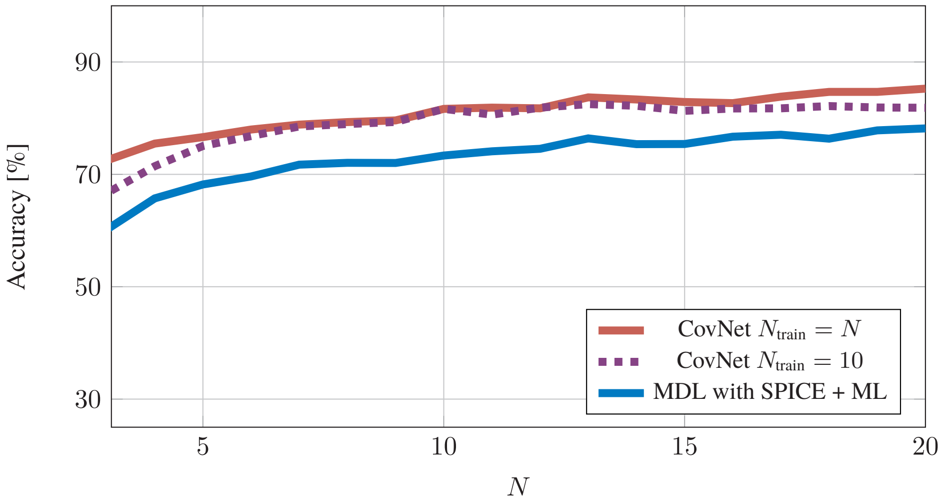


Figure 9.2: Model Order Selection Accuracy vs. N

test set in red. Fig. 9.3 shows that a CovNet approach that is trained over a whole range of frequencies is able to achieve almost the same accuracy at each frequency as the CovNet NNs, which have seen data from the respective frequency during the training process. In contrast, the CovNet NN that has been trained only on $R/\lambda = 1$ is not able to generalize well to other frequencies.

Next, we investigate the performance of the online learning procedure discussed in Section 8.3.1. For this purpose, let us consider a deployment of the previously considered UCA in the field. Due to some imperfections and effects that have not been accounted for in our model for the steering vectors, the true array manifold is different from the ideal UCA manifold in (7.1). We model these imperfections by a global calibration matrix $\boldsymbol{F}$, which is multiplied to the ideal UCA manifold $\boldsymbol{A}_{\text{UCA}}$ to form the calibrated manifold

$$\boldsymbol{A}_{\text{cal}} = \boldsymbol{F}\boldsymbol{A}_{\text{UCA}}. \tag{9.2}$$

Such a global calibration matrix is often used for modeling the effects of mutual coupling among the antenna elements [33, Ch. 3]. For our simulations, we assume a tridiagonal calibration matrix, which reads as

$$\boldsymbol{F} = \begin{bmatrix} 1 & 0.25 & 0 & \dots & 0 \\ 0.25 & 1 & 0.25 & \dots & 0 \\ 0 & 0.25 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix}. \tag{9.3}$$

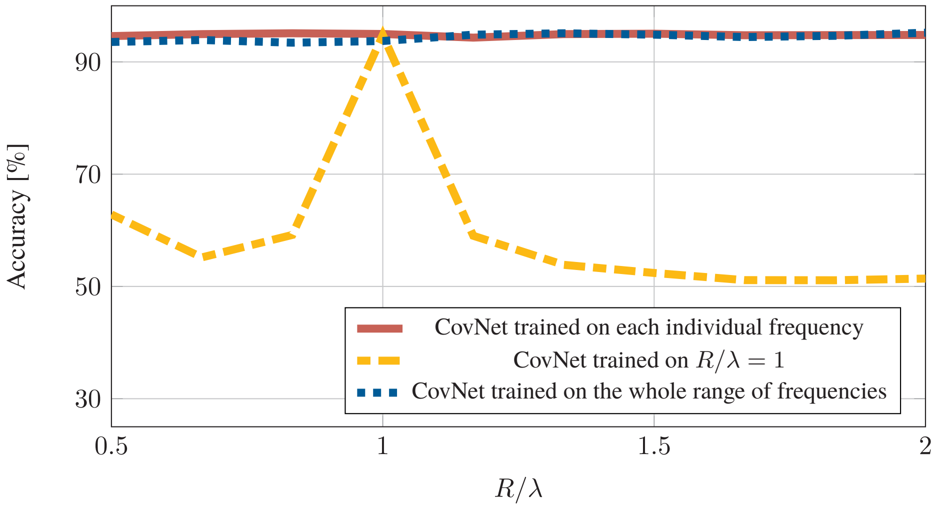


Figure 9.3: Model Order Selection Accuracy for Different Frequencies

Now, Fig. 9.4 shows the achieved accuracy of the online learning procedure starting from a random initialization and the proposed initialization by training on artificial data with the ideal UCA manifold $\boldsymbol{A}_{\text{UCA}}$. Thereby, the abscissa denotes the amount of available data in the number of batches, each with 64 data samples from the true signal model that uses the calibrated manifold $\boldsymbol{A}_{\text{cal}}$, which can be used for the online learning. As a reference, we included the achievable accuracy of the CovNet NN trained on 64 million data samples from the calibrated model. With an initialization by artificial data from the ideal UCA model, the online training is able to achieve an accuracy of over 70% after only one batch of data. In contrast, the random initialization needs about 500 batches to do so. This amount of data leads to a performance close to the reference NN if the proper initialization is chosen.

Finally, we conclude this chapter by a brief discussion of the computational complexity behind the different model order selection approaches. For 1000 realizations with varying SNR evaluated on the same simulation server as discussed in Chapter 7, the evaluation of the CovNet estimator takes about 2.6 seconds. The GramNet evaluation lies in the order of 15 seconds (see Table 7.3). However, the MDL estimator based on SPICE takes 505.2 seconds for estimating the model order. In contrast to the other two methods, for the MDL information criterion, DoA estimates for all possible model orders—including the computationally expensive high model orders—have to be performed, which explains this enormous difference by a factor of 200. Although the MDL approach automatically yields the respective DoA estimate, its execution

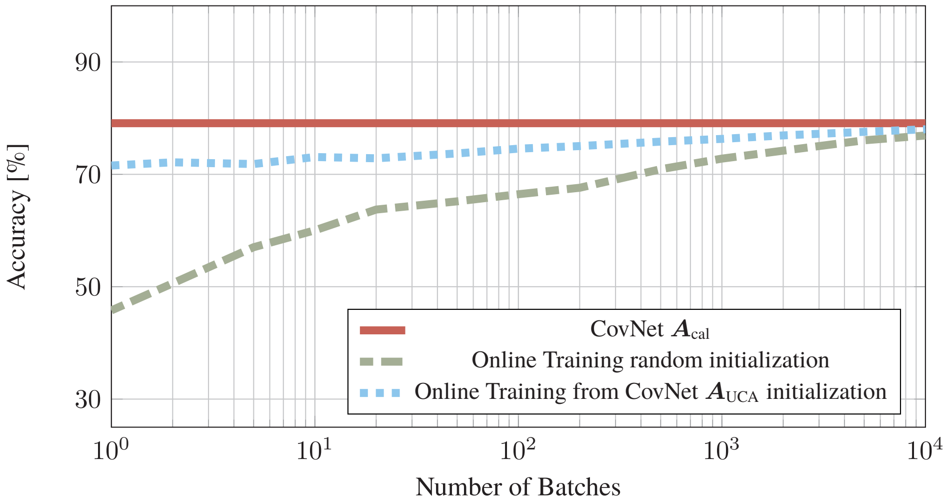


Figure 9.4: Model Order Selection Accuracy for Online Learning Procedure

time is still larger than a CovNet evaluation and a consecutive DoA estimation for the estimated model order (cf. Table 7.3 for $L = 3$).

Conclusion and Outlook 10

In this work, we have investigated DoA estimation techniques and model order selection approaches for systems that employ subarray sampling. For the DoA estimation problem, we have revisited and augmented classical estimators that are based on the underlying stochastic model. In particular, we have discussed ML techniques, the GLS estimator, and sparse recovery methods. Furthermore, we have proposed multiple, new data-based estimators that tackle the DoA estimation problem from three different directions: an end-to-end regression formulation that directly optimizes on the cost function of interest, a formulation of the DoA estimation as a multilabel, multiclass classification problem, which utilizes a sectorization of the field of view, and an estimation of the fully sampled covariance matrix from the subarray sample covariance information.

In our simulations, we have observed that these NN-based estimators are able to clearly outperform classical solutions in terms of estimation accuracy and computational complexity. Especially in the low snapshot domain and for the critical case with as much sources as RF chains, the classical estimators like GLS and SPICE do not provide a competitive performance, due to their covariance matching objective. In contrast, the proposed NN-based techniques show a superior precision because of a higher robustness against outliers. Additionally, we have shown that these machine learning techniques can be combined with a subsequent gradient approach to benefit from the knowledge of the stochastic model such that in most cases the same estimation error as an ML estimator is achieved.

Regarding the model order selection problem, we have discussed the extension of information criteria to the subarray sampling case. Here, the problem is that the ML estimators cannot be directly evaluated for potential model orders above or equal to the number of RF chains. Instead, a replacement of these estimates by the GLS estimator (as has been proposed in [14]), or better, one of the superior hybrid estimators, can be used. Alternatively, the problem can be treated as a standard multiclass classification problem and solved by a NN that has been trained on a cross-entropy objective. Finally, we have discussed the possibility to reuse the estimator of the fully sampled covariance matrix and derive the model order from the eigenvalues of the reconstructed covariance matrix.

Again, we have seen that the classification-based NN approach has been able to significantly outperform the classical information criteria. The NN achieves a higher selection accuracy at a fraction of the computational cost. The model order selection based on the reconstructed covariance matrix turned out to yield good results at high SNR, but failed to provide reasonable estimates at lower SNR. For the eigenvalue based parameterization of the information criteria used in this method, the correct choice for the penalty terms and the number of observations is still unknown. For future work, a more in-depth investigation of this method and optimization of these parameters, e.g., by a cross-validation approach, might be interesting, as the full covariance estimate can be used for DoA estimation and model order selection.

In general, an extension of the investigated DoA estimation and model order selection methods based on machine learning techniques to other scenarios, where either no satisfying performance is obtained by known estimators, or the evaluation of well performing estimators is very expensive, may be worthwhile. Especially, the field of wideband DoA estimation comes to mind. There, we have a similar system model as for the subarray sampling case. Instead of working with different subarrays, in wideband processing, we obtain received signals at different frequencies. As has been discussed in the introduction, these parallels have already been explored in [5] by employing focusing matrices that have been originally proposed in the context of wideband signals [6, 7]. In particular, using the principle behind the full covariance matrix reconstruction described in Section 5.4 may provide a promising starting point for estimating a narrowband covariance matrix at a center frequency based on the sample covariance information from multiple frequencies. Another interesting application are sparse linear arrays, such as minimum redundancy arrays, nested arrays, and co-prime arrays, where the construction of the co-array manifold allows the DoA estimation for more sources than antenna elements (see, e.g., [20, 21, 22]).

More Simulation Results for DoA Estimation A

In this chapter, we provide additional simulation results and compare the classifier based DoA estimators, and the covariance reconstruction techniques. For a more in depth discussion and the underlying simulation parameters see Chapter 7.

A.1 Classification Based Estimators

In Fig. A.1 and Fig. A.2, we depict the simulation results for the classifier-based DoA estimators, for $L = 2$ and $L = 3$, respectively. In contrast to Chapter 7, we include the hybrid CEClaNet and ProjNet methods in these plots. The hybrid CEClaNet performs very similar to the hybrid BRClaNet method, which is also based on a peak detection in the output spectrum, but uses a different loss function and activation in the output layer. Comparing the classifier chain approaches, we can observe that the hybrid ProjNet estimator, which in between its stages removes the signal portions that can be explained by DoA estimates from the previous stages, shows a similar behavior as its ChainNet counterpart for $L = 2$. However, for $L = 3$, ProjNet fails to provide meaningful estimates for the third DoA, and therefore, it shows a poor overall estimation performance in this case.

To confirm that the poor performance of the ProjNet for $L = 3$ indeed stems from the estimate of the last stage, let us first look at the CDFs of the non-hybrid estimators for $L = 3$ in A.3. We can clearly see that ChainProjNet provides poor estimates for all realizations. Now, we look at the estimation performance of the first two stages of the classifier chains in the CDF plot in Fig. A.4. As a reference, we added the results of the first two stages of ChainNet, and for the other estimators, the estimates of the two sources with the smallest estimation error. We see that the performance of the ProjNet is reasonable and not dominated by outliers, i.e., the large estimation errors for $L = 3$ stem from the last stage. Apparently, the projection steps eliminate too much of the received signals to be able to infer the DoA of the last source. Furthermore, we can identify in Fig. A.4 that the BRClaNet and CEClaNet approach are able to attain

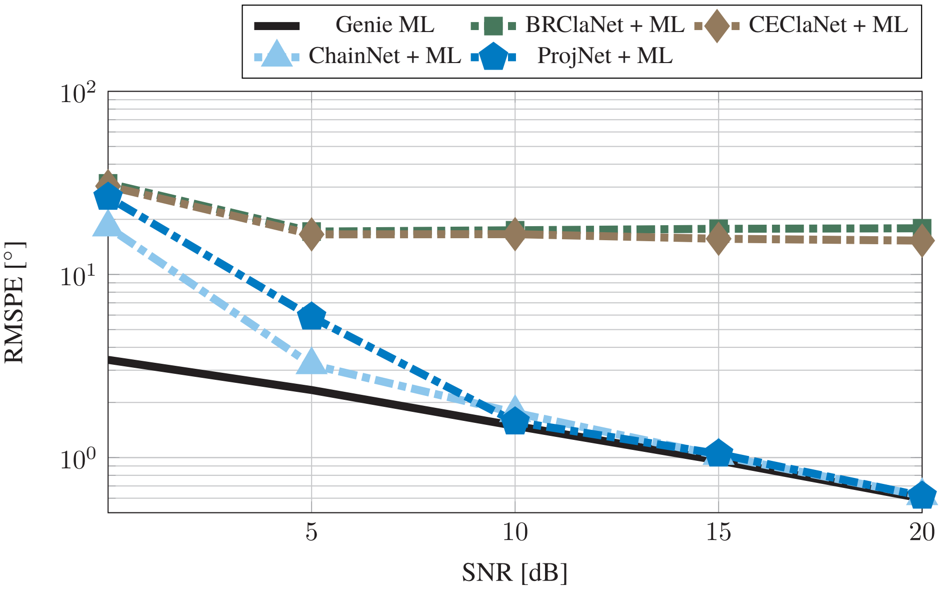


Figure A.1: Performance of Classifiers, $L = 2$, $N = 10$.

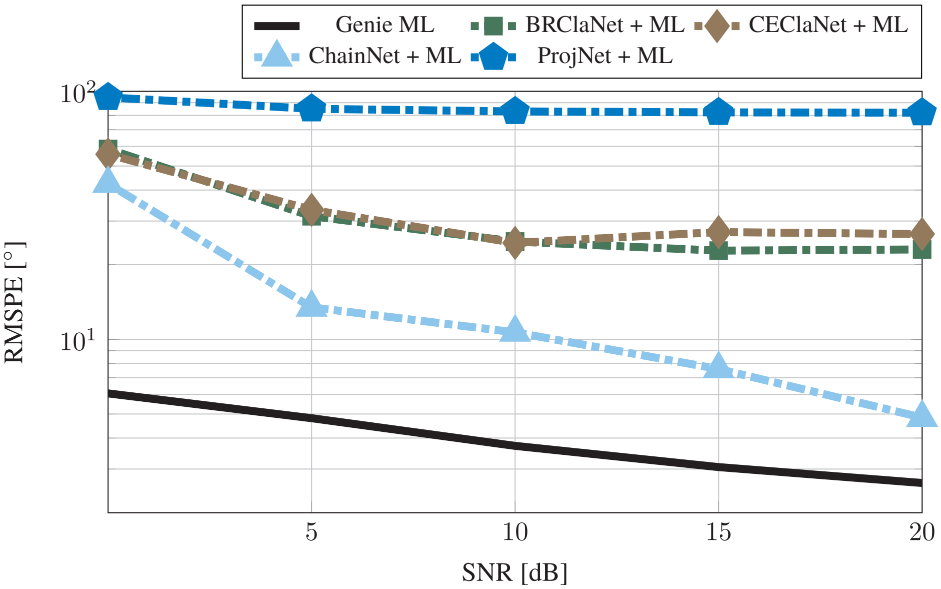


Figure A.2: Performance of Classifiers, $L = 3$, $N = 10$.

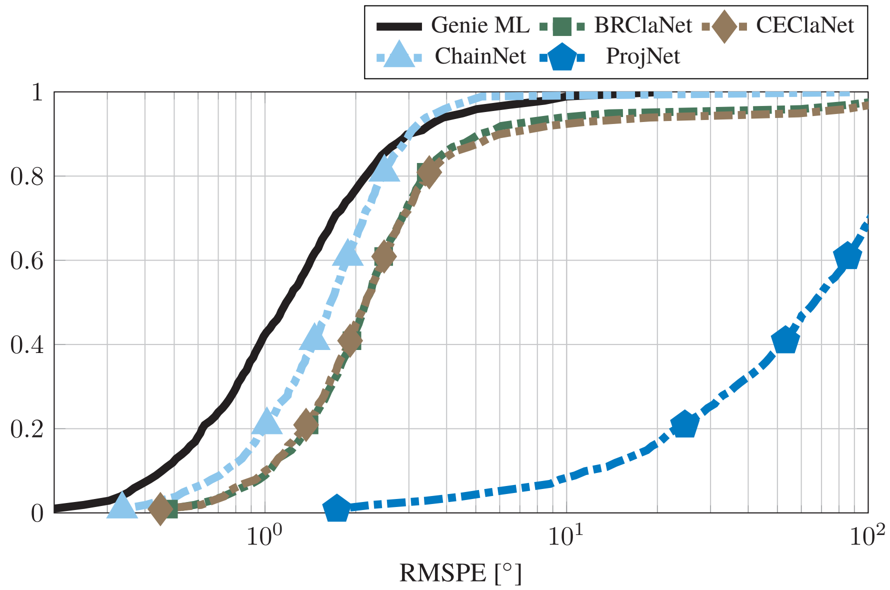


Figure A.3: Empirical Cumulative Density Function of the Hybrid Estimators, $L = 3$, SNR$= 20\,$dB, $N = 10$.

a similar estimation performance as the ChainNet approach for two of the sources, which means that for their respective outlier realizations the estimate of one source must be far from the true value.

A.2 ReconNet and GramNet Results

In this section, we compare the performance of the different covariance reconstruction techniques that have been presented in Section 5.4. To this end, Fig. A.5 shows the simulation results of ReconNet and the different GramNet versions[1] for $L = 3$ sources. There, we can see that ReconNet and GramNet trained on the Frobenius norm have the worst performance. The best performing GramNet NNs have been trained on the Jensen-Bregman LogDet divergence and the affine invariant distance between two positive definite matrices. Similar results can also be seen if a gap of 10° between two sources is introduced. In that case, the distance between the NNs that have been trained on the Frobenius norm, i.e., ReconNet and "GramNet Fro", and the other GramNet

[1] To differentiate between the different GramNet versions, we indicate which distance measure has been used during the training process. Thereby, we use the following abbreviation: Frobenius norm (Fro), Affine Invariant distance (AffInv), Log-Frobenius (LogFro), Cholesky-Euclidean (Chol), J-Divergence (J-Div), and Jensen-Bregman LogDet Divergence (JBD).

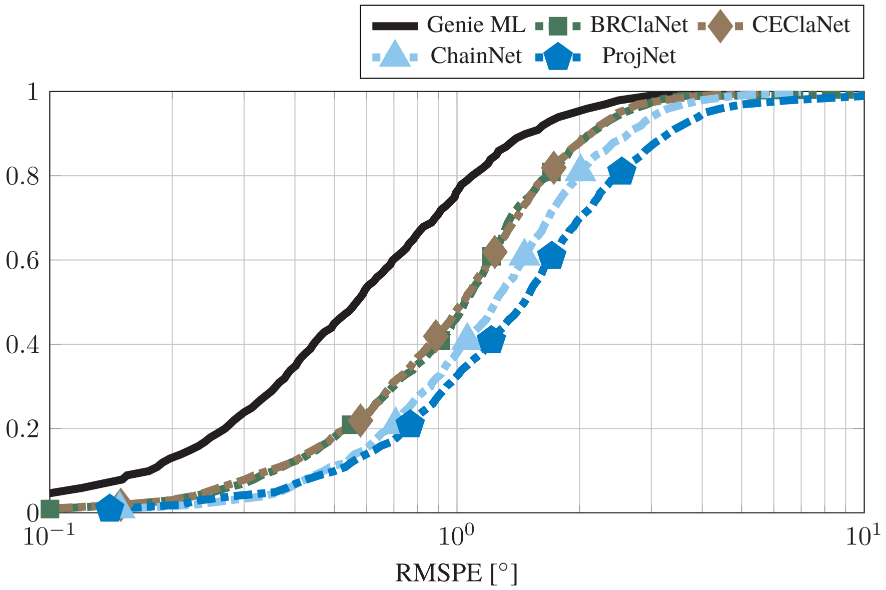


Figure A.4: Empirical CDF for the two First Stages of the Classifier Chains, $L = 3$, SNR$=$ 20 dB, $N = 10$.

NNs increases. Note that we do not include the results for the GramNet trained on the Bures-Wasserstein distance, as the resulting NN did not yield meaningful estimates.

In Table A.1, we show the squared distance between the reconstructed covariance matrix by the different GramNet implementations and the true covariance matrix. To obtain these results, we performed 10^6 Monte Carlo simulations. For each Monte Carlo run, we create a full covariance matrix with $L = 3$ uniformly distributed DoAs, random SNR between 0 dB and 20 dB, and random transmit powers between 0 dB and -9 dB, as for the training data in Section 5.1. For each full covariance matrix, we draw a $N = 10$ snapshots per subarray, and feed the respective subarray sample covariance matrices to the GramNet. Each column contains the results for one of the distances described in Table 5.1. Interestingly, the NN that is trained on a certain distance measure does not always achieve the best result in the respective distance[2]. Nevertheless, we can see that the best performing NNs in terms of RMSPE, i.e., “GramNet AffInv” and “GramNet JBD”, perform among the worst with regard to the Frobenius Norm, however, they are among the best for all other distances, especially for the geodesic ones.

[2] Note that we are dealing with NNs that all employ the same architecture, but are trained on different objectives. Naturally, the objective has a huge impact on the local minimum to which the NN converges.

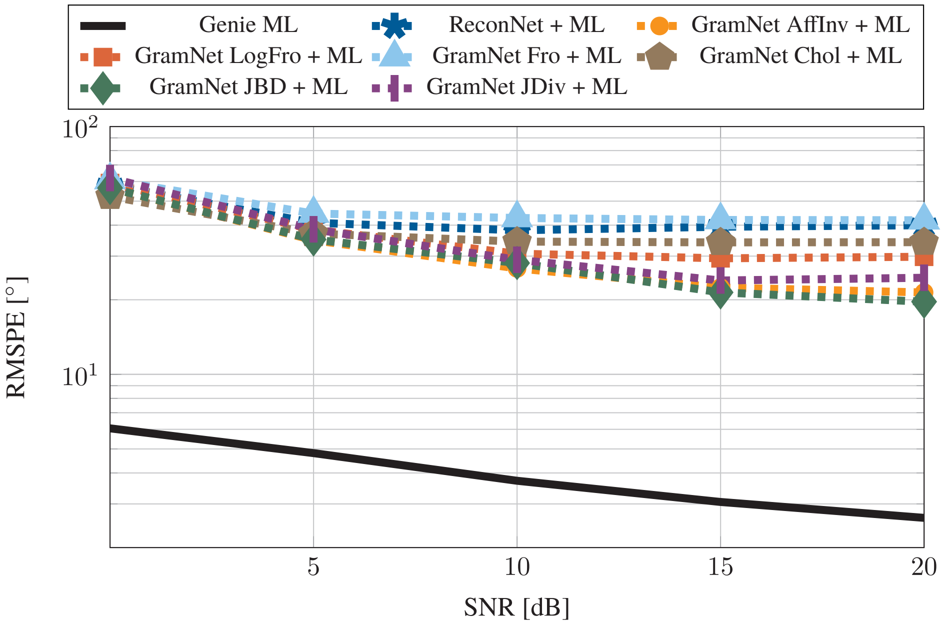


Figure A.5: Performance of MUSIC, RMSPE vs. SNR, $L = 3$, $N = 10$.

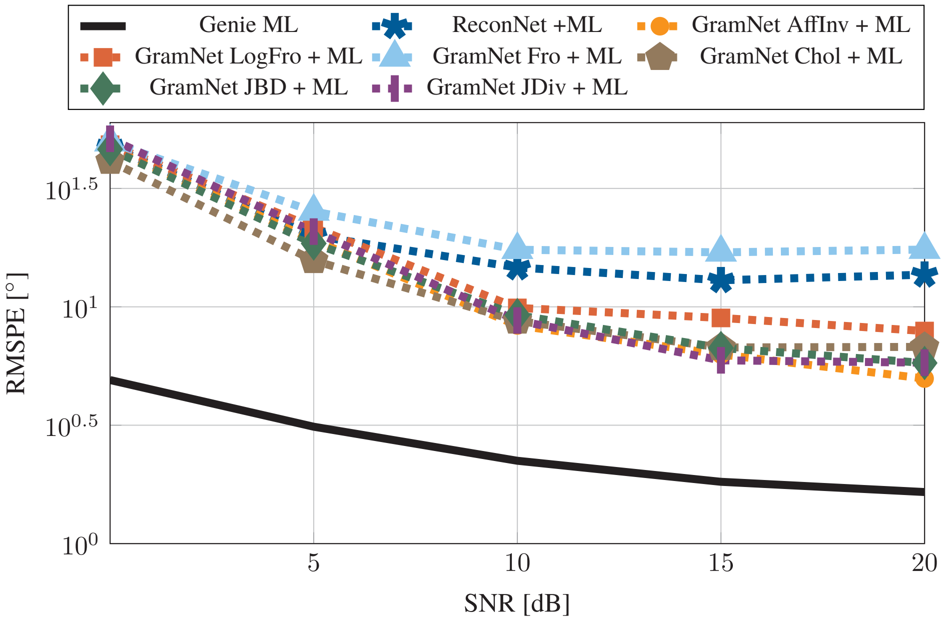


Figure A.6: Performance of MUSIC, RMSPE vs. SNR, $L = 3$, 10° gap between two sources, $N = 10$.

Distance	Fro	AffInv	LogFro	Chol	JBD	JDiv	Bu.-Wa.
GramNet Fro	0.1745	15.34	13.39	0.2609	1.594	7.500	1.636
GramNet AffInv	0.2550	7.985	6.033	0.2005	0.9077	2.563	1.360
GramNet LogFro	0.1916	9.466	6.487	0.1901	1.055	3.259	1.554
GramNet Chol	0.1580	8.99	6.136	0.1603	1.000	4.548	1.540
GramNet JBD	0.2233	7.902	5.948	0.1910	0.8979	2.542	1.403
GramNet JDiv	0.4401	9.275	7.346	0.2730	1.045	3.039	1.188

Table A.1: Distance Between True Covariance Matrices and Reconstructed Covariance Matrices

Finally, we want to compare the GramNet approach to the trivial covariance reconstruction techniques for ULAs discussed in Section 5.4. To that end, we present simulation results for an 8-element ULA with $\frac{\lambda}{2}$ spacing and $L = 2$ equally powered sources. The subarray sampling scheme follows the description above (5.9), i.e., we always sample the first antenna, while the second RF chain consecutively samples the other antennas. The resulting Root Mean Squared Error (RMSE) for 10^4 samples per SNR with a 10° gap between the sources can be found in Fig. A.7, and the CDF of all realizations at 20 dB is depicted in Fig. A.8. We can see that the GramNet clearly outperforms the trivial estimators “ULA Recon (5.13)” and “ULA Recon (5.14)”, which are heavily affected by outliers.

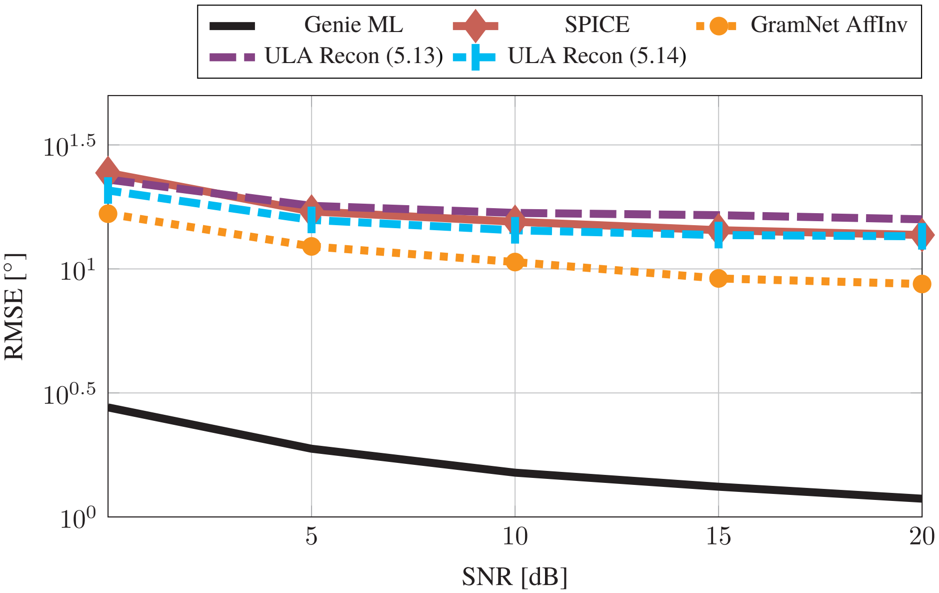


Figure A.7: ULA Reconstruction Techniques, RMSE vs. SNR, $L = 2$, 10° gap between two sources, $N = 10$.

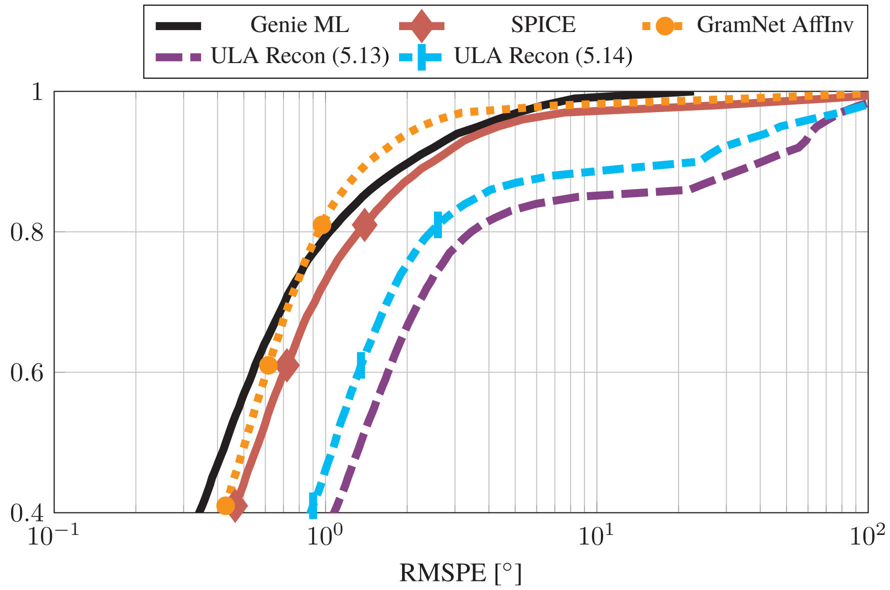


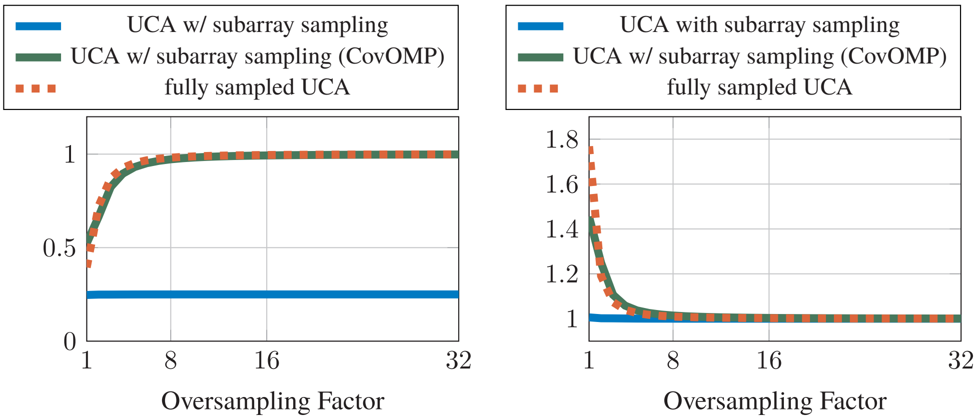
Figure A.8: Empirical CDF for ULA Reconstruction Techniques, $L = 2$, SNR= 20 dB, $N = 10$.

B Outlier Susceptibility of OMP-Based Estimators

As we can see in the simulation results presented in Chapter 7, the OMP estimators provide only poor estimation results. However, in many other applications, OMP algorithms have been shown to be quite potent. Here, we take a closer look on the reasons why the OMP based estimators cannot cope with the DoA estimation problem.

As we have discussed in 4.2.1, in theory, there exist nice recovery guarantees for the OMP and BOMP methods that are based on the mutual coherence of the dictionary. Unfortunately, for the considered systems with subarray sampling and even for fully sampled arrays none of these theoretical guarantees hold. This is confirmed in Fig. B.1, where we show on the left hand side the mutual coherence of the dictionaries for the UCA considered in Chapter 7 for a single snapshot $N = 1$. Here, we depict the mutual coherence for the sparse model of the received signals (4.8) in blue, the covariance formulation (4.9) in green and the mutual coherence for the fully sampled case as a dashed red line. For the fully sampled array and the covariance-based formulation of the subarray sampling case, the mutual coherence quickly approaches 1 for a growing oversampling factor Q, and for the received signal formulation with subarray sampling, which uses the BOMP algorithm, the coherence is almost at its maximum value of $K^{-1} = 0.25$ even for $Q = 1$. Hence, the upper bound for the reconstructible number of sources in (4.12) and (4.14) compute to the numbers shown on the right hand side of Fig. B.1. Since these are below two, for every oversampling factor, we cannot guarantee reconstruction for two sources even in the noiseless case.

In the following, we want to briefly discuss a case, where the OMP systematically fails to resolve both sources. To that end, let us look at an example. We consider $L = 2$ sources located at $\boldsymbol{\theta} = [0°, 105°]^{\mathrm{T}}$ with transmit signals $\boldsymbol{s}^{(k)} = \mathbf{1}_L, k = 1, \ldots, K$. The UCA with the parameters from Chapter 7 collects a single, noise free snapshot, i.e., $\boldsymbol{\eta}^{(k)} = \mathbf{0}, k = 1, \ldots, K$. The corresponding beam pattern for the source at 0° is depicted at the top of Fig. B.2. Note that at $105°$, the second source lies directly at a sidelobe peak of the beam pattern of the first source. In the middle of Fig. B.2, we



(a) Mutual Coherence

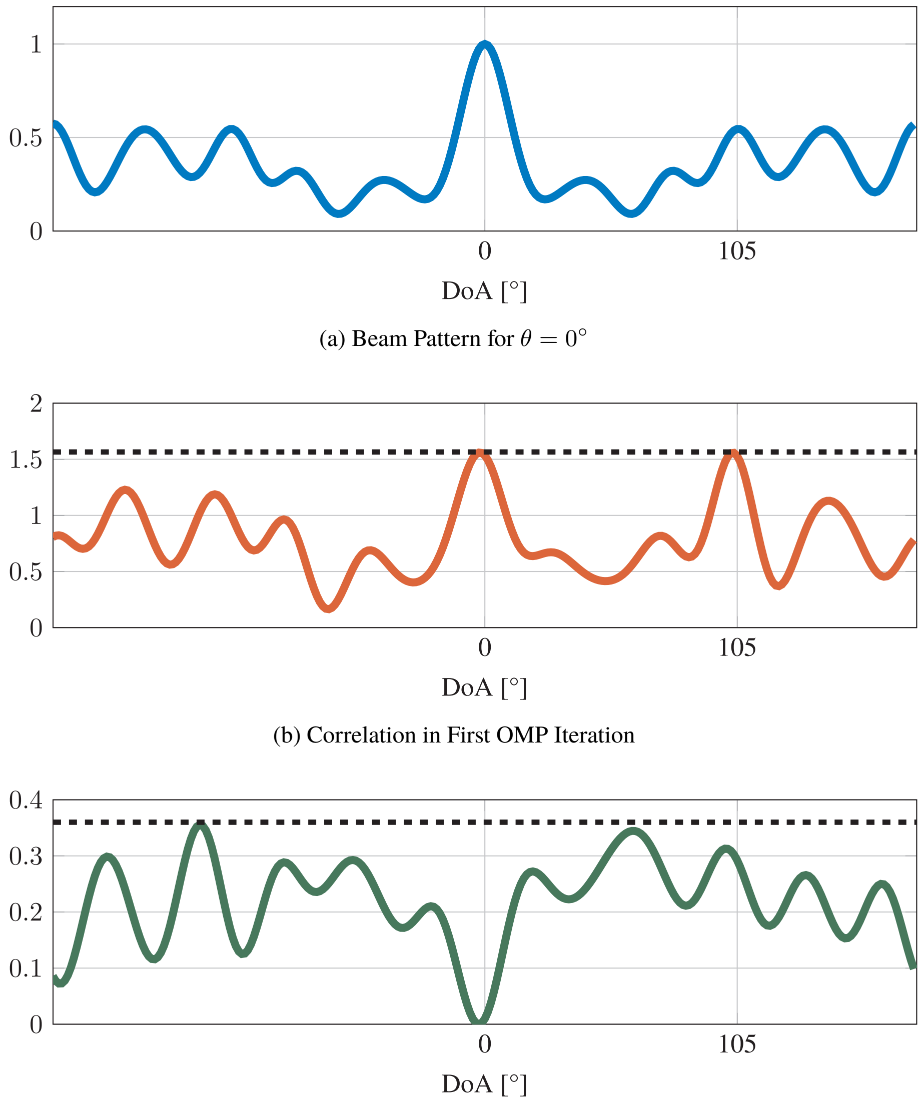
(b) Number of Reconstructible Sources

Figure B.1: Mutual Coherence and Reconstruction Guarantee for Different Oversampling Factors.

show the correlation of the received signal with the dictionary, i.e., the correlation function that is formed in the first stage of the OMP algorithm. There, we can clearly recognize two peaks with equal height close to the true DoAs. However, when we look at the correlation function in the second stage, shown at the bottom of Fig. B.2, we now see that after the projection, the maximum correlation with the residuum is not obtained at the DoA of the second source, but at about $-120°$. This means that in the second stage, the DoA estimate of the OMP algorithm is about $-120°$ and thus very far away from the true DoA of the second source. Hence, this realization would lead to an outlier, although we considered no noise. The problem is that as the second source lies in a sidelobe of the first source, the projection of the elimination of the signal portion of the first detected DoA does not only remove the signal of the first source, but a significant portion of the second source gets eliminated by the projection[1]. Therefore, the correlation with the residuum in the second stage of OMP has its maximum not in the vicinity of the true DoA of the second source.

The example above illustrates that the OMP based estimators may experience systematic outliers in the case where two sources lie in each others sidelobe. A way to reduce this effect lies in utilizing antenna arrays with lower sidelobe levels. However, in general, for such arrays the curvature of the mainlobe is smaller, i.e., the overall estimation accuracy decreases.

[1] The same happens for closely spaced sources.



(a) Beam Pattern for $\theta = 0^\circ$

(b) Correlation in First OMP Iteration

(c) Correlation in Second OMP Iteration

Figure B.2: Systematic Outlier of OMP

Bibliography

[1] H. Krim and M. Viberg, “Two decades of array signal processing research: The parametric approach,” *IEEE Signal Process. Mag.*, vol. 13, no. 4, pp. 67–94, Jul. 1996.

[2] H. L. V. Trees, *Optimum Array Processing: Part IV of Detection, Estimation, and Modulation Theory*. Wiley, 2002.

[3] A. Barthelme and W. Utschick, “A machine learning approach to DoA estimation and model order selection for antenna arrays with subarray sampling,” *IEEE Trans. Sig. Process.*, vol. 69, pp. 3075–3087, May 2021.

[4] A. Zeira and B. Friedlander, “Direction finding with time-varying arrays,” *IEEE Trans. Signal Process.*, vol. 43, no. 4, pp. 927–937, Apr. 1995.

[5] B. Friedlander and A. Zeira, “Eigenstructure-based algorithms for direction finding with time-varying arrays,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 32, no. 2, pp. 689–701, Apr. 1996.

[6] H. Hung and M. Kaveh, “Focussing matrices for coherent signal-subspace processing,” *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 36, no. 8, pp. 1272–1281, Aug. 1988.

[7] M. A. Doron and A. J. Weiss, “On focusing matrices for wide-band array processing,” *IEEE Trans. Signal Process.*, vol. 40, no. 6, pp. 1295–1302, Jun. 1992.

[8] J. Worms, “RF direction finding with a reduced number of receivers by sequential sampling,” in *Proc. Int. Conf. on Phased Array Syst. and Technol.*, May 2000, pp. 165–168.

[9] D. Rieken and D. Fuhrmann, “Generalizing MUSIC and MVDR for multiple noncoherent arrays,” *IEEE Trans. Signal Process.*, vol. 52, no. 9, pp. 2396–2406, Sep. 2004.

[10] J. Sheinvald and M. Wax, “Direction finding with fewer receivers via time-varying preprocessing,” *IEEE Trans. Signal Process.*, vol. 47, no. 1, pp. 2–9, Jan. 1999.

[11] E. Fishler and H. Messer, “Multiple source direction finding with an array of M sensors using two receivers,” in *Proc. Workshop on Statistical Signal and Array Process.*, Aug. 2000, pp. 86–89.

[12] R. Schmidt, “Multiple emitter location and signal parameter estimation,” *IEEE Trans. Antennas Propag.*, vol. 34, no. 3, pp. 276–280, Mar. 1986.

[13] W. Suleiman, P. Parvazi, M. Pesavento, and A. M. Zoubir, “Non-coherent direction-of-arrival estimation using partly calibrated arrays,” *IEEE Trans. Signal Process.*, vol. 66, no. 21, pp. 5776–5788, Nov. 2018.

[14] J. Sheinvald and M. Wax, “Detection and localization of multiple signals using subarrays data,” in *Advances in Spectr. Anal. and Array Process. (vol. III)*, S. Haykin, Ed. Prentice-Hall, Inc., 1995, pp. 324–351.

[15] A. Barthelme, R. Wiesmayr, and W. Utschick, “Model order selection in DoA scenarios via cross-entropy based machine learning techniques,” in *Proc. ICASSP*, May 2020, pp. 4622–4626.

[16] R. Bowden, “The theory of parametric identification,” *Econometrica*, vol. 41, no. 6, pp. 1069–1074, Nov. 1973.

[17] M. Wax and I. Ziskind, “On unique localization of multiple sources by passive sensor arrays,” *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 37, no. 7, pp. 996–1000, Jul. 1989.

[18] M. Wax and T. Kailath, “Detection of signals by information theoretic criteria,” *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 33, no. 2, pp. 387–392, Apr. 1985.

[19] B. Hochwald and A. Nehorai, “Identifiability in array processing models with vector-sensor applications,” *IEEE Trans. Signal Process.*, vol. 44, no. 1, pp. 83–95, Jan. 1996.

[20] W. Ma, T. Hsieh, and C. Chi, “DOA estimation of quasi-stationary signals with less sensors than sources and unknown spatial noise covariance: A Khatri–Rao subspace approach,” *IEEE Trans. Signal Process.*, vol. 58, no. 4, pp. 2168–2180, Apr. 2010.

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

[22] P. P. Vaidyanathan and P. Pal, "Sparse sensing with co-prime samplers and arrays," *IEEE Trans. Signal Process.*, vol. 59, no. 2, pp. 573–586, Feb. 2011.

[23] H. Cramér, *Mathematical Methods of Statistics*. Princeton University Press, 1946.

[24] C. R. Rao, "Information and the accuracy attainable in the estimation of statistical parameters," *Bulletin of Calcutta Mathematical Society*, vol. 37, pp. 81–89, 1945.

[25] P. Stoica and R. L. Moses, *Introduction to Spectral Analysis*. Prentice Hall, 1997.

[26] J. Sheinvald, M. Wax, and A. Weiss, "On the achievable localization accuracy of multiple sources at high SNR," *IEEE Trans. Signal Process.*, vol. 45, no. 7, pp. 1795–1799, Jul. 1997.

[27] A. Barthelme, J. Kunz, and W. Utschick, "An MSE approximation for grid-based maximum likelihood direction-of-arrival estimators," in *Proc. WSA*, Feb. 2020, pp. 1–6.

[28] F. Athley, "Threshold region performance of maximum likelihood direction of arrival estimators," *IEEE Trans. Signal Process.*, vol. 53, no. 4, pp. 1359–1373, Apr. 2005.

[29] E. Barankin, "Locally best unbiased estimates," *Ann. Math. Statist.*, vol. 20, no. 4, pp. 477–501, Dec. 1949.

[30] F. Glave, "A new look at the Barankin lower bound," *IEEE Trans. Inf. Theory*, vol. 18, no. 3, pp. 349–356, May 1972.

[31] K. Bell, Y. Steinberg, Y. Ephraim, and H. V. Trees, "Extended Ziv-Zakai lower bound for vector parameter estimation," *IEEE Trans. Inf. Theory*, vol. 43, no. 2, pp. 624–637, Mar. 1997.

[32] E. Weinstein and A. Weiss, "A general class of lower bounds in parameter estimation," *IEEE Trans. Inf. Theory*, vol. 34, no. 2, pp. 338–342, Mar. 1988.

[33] T. E. Tuncer and B. Friedlander, *Classical and modern direction-of-arrival estimation*. Academic Press, 2009.

[34] J. G. Proakis and M. Salehi, *Digital Communications*, 5th ed. McGraw-Hill Higher Education, 2008.

[35] D. Kuonen, “Saddlepoint approximations for distributions of quadratic forms in normal variables,” *Biometrika*, vol. 86, no. 4, pp. 929–935, Dec. 1999.

[36] H. E. Daniels, “Saddlepoint approximations in statistics,” *Annals of Mathematical Statistics*, vol. 25, no. 4, pp. 631–650, Dec. 1954.

[37] P. Stoica and A. Nehorai, “MUSIC, maximum likelihood, and cramer-rao bound,” *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 37, no. 5, pp. 720–741, May 1989.

[38] A. Genz, “Numerical computation of multivariate normal probabilities,” *J. Comput. and Graphical Statist.*, vol. 1, no. 2, p. 141, Jun. 1992.

[39] B. Ottersten, P. Stoica, and R. Roy, “Covariance matching estimation techniques for array signal processing applications,” *Digit. Signal Process.*, vol. 8, no. 3, pp. 185–210, Jul. 1998.

[40] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[41] D. L. Donoho, “Compressed sensing,” *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[42] H. Boche, R. Calderbank, G. Kutyniok, and J. Vybíral, “A survey of compressed sensing,” in *Compressed Sensing and its Applications*, H. Boche, R. Calderbank, G. Kutyniok, and J. Vybíral, Eds. Springer, 2015, pp. 1–39.

[43] Z. Yang, J. Li, P. Stoica, and L. Xie, “Sparse methods for direction-of-arrival estimation,” in *Academic Press Library in Signal Processing*. Elsevier, 2018, vol. 7, pp. 509–581.

[44] Y. C. Eldar and M. Mishali, “Robust recovery of signals from a structured union of subspaces,” *IEEE Trans. Inf. Theory*, vol. 55, no. 11, pp. 5302–5316, Nov. 2009.

[45] R. G. Baraniuk, V. Cevher, M. F. Duarte, and C. Hegde, "Model-based compressive sensing," *IEEE Trans. Inf. Theory*, vol. 56, no. 4, pp. 1982–2001, Apr. 2010.

[46] Y. C. Eldar, P. Kuppinger, and H. Bolcskei, "Block-sparse signals: Uncertainty relations and efficient recovery," *IEEE Trans. Signal Process.*, vol. 58, no. 6, pp. 3042–3054, Jun. 2010.

[47] P. Stoica, P. Babu, and J. Li, "SPICE: A sparse covariance-based estimation method for array processing," *IEEE Trans. Signal Process.*, vol. 59, no. 2, pp. 629–638, Feb. 2011.

[48] Y. D. Zhang, M. G. Amin, and B. Himed, "Sparsity-based DOA estimation using co-prime arrays," in *Proc. ICASSP*, May 2013, pp. 3967–3971.

[49] Y. Pati, R. Rezaiifar, and P. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," in *Proc. Asilomar Conf. Sig. Syst. Comp.*, Nov. 1993, pp. 40–44 vol.1.

[50] J. Tropp, "Greed is good: Algorithmic results for sparse approximation," *IEEE Trans. Inf. Theory*, vol. 50, no. 10, pp. 2231–2242, Oct. 2004.

[51] D. Donoho, M. Elad, and V. Temlyakov, "Stable recovery of sparse overcomplete representations in the presence of noise," *IEEE Trans. Inf. Theory*, vol. 52, no. 1, pp. 6–18, Jan. 2006.

[52] Z. Ben-Haim and Y. C. Eldar, "Near-oracle performance of greedy block-sparse estimation techniques from noisy measurements," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 5, pp. 1032–1047, Sep. 2011.

[53] R. Rastogi, P. Gupta, and R. Kumaresan, "Array signal processing with interconnected neuron-like elements," in *Proc. ICASSP*, vol. 12, Apr. 1987, pp. 2328–2331.

[54] K. L. Du, A. K. Y. Lai, K. K. M. Cheng, and M. N. S. Swamy, "Neural methods for antenna array signal processing: A review," *Signal Process.*, vol. 82, no. 4, pp. 547–561, Apr. 2002.

[55] Y. LeCun, Y. Bengio, and G. E. Hinton, "Deep learning," *Nat.*, vol. 521, no. 7553, pp. 436–444, May 2015.

[56] S. Chakrabarty and E. A. P. Habets, “Broadband DOA estimation using convolutional neural networks trained with noise signals,” *Proc. WASPAA*, pp. 136–140, Oct. 2017.

[57] Z.-M. Liu, C. Zhang, and P. S. Yu, “Direction-of-arrival estimation based on deep neural networks with robustness to array imperfections,” *IEEE Trans. Antennas Propag.*, vol. 66, no. 12, pp. 7315–7327, Dec. 2018.

[58] S. Chakrabarty and E. A. P. Habets, “Multi-speaker DOA estimation using deep convolutional networks trained with noise signals,” *IEEE J. Sel. Topics Signal Process.*, vol. 13, no. 1, pp. 8–21, Mar. 2019.

[59] E. Ozanich, P. Gerstoft, and H. Niu, “A feedforward neural network for direction-of-arrival estimation,” *J. Acoustical Soc. Amer.*, vol. 147, no. 3, pp. 2035–2048, Mar. 2020.

[60] Y. Yao, H. Lei, and W. He, “A-CRNN-based method for coherent DOA estimation with unknown source number,” *Sensors*, vol. 20, no. 8, p. 2296, Jan. 2020.

[61] G. K. Papageorgiou, M. Sellathurai, and Y. C. Eldar, “Deep networks for direction-of-arrival estimation in low SNR,” *arXiv:2011.08848 [cs, eess]*, Nov. 2020. [Online]. Available: http://arxiv.org/abs/2011.08848

[62] L. Wu, Z.-M. Liu, and Z.-T. Huang, “Deep convolution network for direction of arrival estimation with sparse prior,” *IEEE Signal Process. Lett.*, vol. 26, no. 11, pp. 1688–1692, Nov. 2019.

[63] A. M. Elbir, “DeepMUSIC: Multiple signal classification via deep learning,” *IEEE Sensors Lett.*, vol. 4, no. 4, pp. 1–4, Apr. 2020.

[64] Y. Guo, Z. Zhang, Y. Huang, and P. Zhang, “DOA estimation method based on cascaded neural network for two closely spaced sources,” *IEEE Signal Process. Lett.*, vol. 27, pp. 570–574, Apr. 2020.

[65] O. Bialer, N. Garnett, and T. Tirer, “Performance advantages of deep neural networks for angle of arrival estimation,” in *Proc. ICASSP*, May 2019, pp. 3907–3911.

[66] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving,

M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org.

[67] P. Costa, J. Grouffaud, P. Larzabal, and H. Clergeot, "Estimation of the number of signals from features of the covariance matrix: A supervised approach," *IEEE Trans. Signal Process.*, vol. 47, no. 11, pp. 3108–3115, Nov. 1999.

[68] R. Reggiannini, "A fundamental lower bound to the performance of phase estimators over Rician-fading channels," *IEEE Trans. Commun.*, vol. 45, no. 7, pp. 775–778, Jul. 1997.

[69] T. Routtenberg and J. Tabrikian, "Non-Bayesian periodic Cramér-Rao bound," *IEEE Trans. Signal Process.*, vol. 61, no. 4, pp. 1019–1032, Feb. 2013.

[70] Q. Li, X. Zhang, and H. Li, "Online direction of arrival estimation based on deep learning," in *Proc. ICASSP*, Apr. 2018, pp. 2616–2620.

[71] J. S. Bridle, "Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters," in *Advances Neural Inf. Process. Syst.*, Nov. 1989, pp. 211–217.

[72] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, pp. 2278–2324, Nov. 1998.

[73] M. Zhang and Z. Zhou, "A review on multi-label learning algorithms," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 8, pp. 1819–1837, Aug. 2014.

[74] F. Herrera, F. Charte, A. J. Rivera, and M. J. d. Jesus, *Multilabel Classification: Problem Analysis, Metrics and Techniques*. Springer International Publishing, 2016.

[75] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown, "Learning multi-label scene classification," *Pattern Recognit.*, vol. 37, no. 9, pp. 1757–1771, Sep. 2004.

[76] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains for multi-label classification," *Mach. Learn.*, vol. 85, no. 3, p. 333, Jun. 2011.

[77] ——, “Classifier chains: A review and perspectives,” *arXiv:1912.13405 [cs, stat]*, Apr. 2020, arXiv: 1912.13405.

[78] A. Barthelme and W. Utschick, “ChainNet: neural network-based successive spectral analysis,” *arXiv:2105.03742 [eess]*, pp. 1–23, May 2021. [Online]. Available: https://arxiv.org/abs/2105.03742

[79] ——, “DoA estimation using neural network-based covariance matrix reconstruction,” *IEEE Signal Processing Letters*, vol. 28, pp. 783–787, 2021.

[80] G. Cybenko, “Approximation by superpositions of a sigmoidal function,” *Math. of Control, Signals and Syst.*, vol. 2, no. 4, pp. 303–314, Dec. 1989.

[81] K. Hornik, “Approximation capabilities of multilayer feedforward networks,” *Neural Netw.*, vol. 4, no. 2, pp. 251–257, Jan. 1991.

[82] C. Liu and P. P. Vaidyanathan, “Remarks on the spatial smoothing step in coarray MUSIC,” *IEEE Signal Processing Letters*, vol. 22, no. 9, pp. 1438–1442, Sep. 2015.

[83] P. Soille, *Morphological Image Analysis*. Springer, 1999.

[84] R. Bhatia, *Positive Definite Matrices*. Princeton University Press, Jan. 2009.

[85] V. Arsigny, P. Fillard, X. Pennec, and N. Ayache, “Log-Euclidean metrics for fast and simple calculus on diffusion tensors,” *Magn. Resonance Medicine*, vol. 56, no. 2, pp. 411–421, Jun. 2006.

[86] R. Vemulapalli and D. W. Jacobs, “Riemannian metric learning for symmetric positive definite matrices,” *arXiv:1501.02393 [cs]*, Jan. 2015. [Online]. Available: http://arxiv.org/abs/1501.02393

[87] I. L. Dryden, A. Koloydenko, and D. Zhou, “Non-Euclidean statistics for covariance matrices, with applications to diffusion tensor imaging,” *Ann. Appl. Statist.*, vol. 3, no. 3, pp. 1102–1123, Sep. 2009.

[88] Z. Wang and B. Vemuri, “An affine invariant tensor dissimilarity measure and its applications to tensor-valued image segmentation,” in *Proc. CVPR*, Jun. 2004, pp. 1–6.

[89] A. Cherian, S. Sra, A. Banerjee, and N. Papanikolopoulos, “Jensen-Bregman logdet divergence with application to efficient similarity search for covariance

matrices," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 9, pp. 2161–2174, Sep. 2013.

[90] R. Bhatia, T. Jain, and Y. Lim, "On the Bures-Wasserstein distance between positive definite matrices," *Expositiones Mathematicae*, vol. 37, no. 2, pp. 165–191, Jun. 2019.

[91] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. Int. Conf. Artif. Intell. Statist.*, vol. 9, May 2010, pp. 249–256.

[92] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv:1412.6980 [cs.LG]*, Dec. 2014. [Online]. Available: https://arxiv.org/abs/1412.6980

[93] J. Lofberg, "YALMIP : A toolbox for modeling and optimization in MATLAB," in *Proc. ICRA*, Sep. 2004, pp. 284–289.

[94] MOSEK ApS, *The MOSEK optimization toolbox for MATLAB manual. Version 9.0.*, 2019. [Online]. Available: http://docs.mosek.com/9.0/toolbox/index.html

[95] P. Stoica and Y. Selen, "Model-order selection: A review of information criterion rules," *IEEE Signal Process. Mag.*, vol. 21, no. 4, pp. 36–47, Jul. 2004.

[96] J. Ding, V. Tarokh, and Y. Yang, "Model selection techniques: An overview," *IEEE Signal Process. Mag.*, vol. 35, no. 6, pp. 16–34, Nov. 2018.

[97] E. Fishler, M. Grosmann, and H. Messer, "Detection of signals by information theoretic criteria: General asymptotic performance analysis," *IEEE Trans. Signal Process.*, vol. 50, no. 5, pp. 1027–1036, May 2002.

[98] P. Costa-Hirschauer, J. Grouffaud, P. Larzabal, and H. Clergeot, "A neural network for model order selection in signal processing," in *Proc. ICNN*, vol. 6, Nov. 1995, pp. 3057–3061.

[99] Y. Yang, F. Gao, C. Qian, and G. Liao, "Model-aided deep neural network for source number detection," *IEEE Signal Process. Lett.*, vol. 27, pp. 91–95, Jan. 2020.

[100] R. Takeda and K. Komatani, "Sound source localization based on deep neural networks with directional activate function exploiting phase information," in *Proc. ICASSP*, Mar. 2016, pp. 405–409.