Decentralized Direction of Arrival Estimation

Vom Fachbereich 18
Elektrotechnik und Informationstechnik
der Technischen Universität Darmstadt
zur Erlangung der Würde eines
Doktor-Ingenieurs (Dr.-Ing.)
genehmigte Dissertation

von
M.Sc. Wassim Suleiman
geboren am 05.02.1981 in Hama (Syrien)

Referent:	Prof. Dr.-Ing. Marius Pesavento
Korreferent:	Prof. Dr.-Ing. Abdelhak M. Zoubir
Tag der Einreichung:	05.05.2017
Tag der mündlichen Prüfung:	07.09.2017

D 17
Darmstadt 2017

Acknowledgments

I would like to thank all the people that supported me during my doctoral study and contributed to this thesis in various ways.

First, I would like to thank Prof. Dr.-Ing. Mrius Pesavento and Prof. Dr.-Ing. Abdelhak Zoubir for their support, trust, and academic guidance. I thank Prof. Heinz Koeppl, Prof. Jürgen Adamy, and Prof. Udo Schwalke, for being on of my PhD committee.

I thank Dr. Pouyan Parvazi for the many inspiring conversations and for his patience and help.

I would like to thank my colleagues of the Communication Systems and Signal Processing Groups: Florian Bahlke, Ganapati Hegde, Minh Hoang, Gerta Kushe, Fabio Nikolay, Dima Taleb, Christian Steffens, Alexander Sorg, Dr. Oscar Dario Ramos Cantor, Dr. Xin Wen, Dr. Xin Zhang, Marlis Gorecki, Klaus Schmidt, Sahar Khawatmi, Lala Khadidja Hamaidi, Freweyni Kidane Teklehaymanot, Di Jin, Mark Ryan Leonard, Patricia Binder, Dominik Reinhard, Tim Schäck, Ann-Kathrin Seifert, Adrian Šošić, Dr. Michael Muma, Dr. Michael Fauß, Dr. Nevine Demitri, Dr. Sara Al-Sayed, Dr. Mouhammad Alhumaidi, Dr. Phillip Heidenreich, Renate Koschella, and Hauke Fath.

My deep and sincere gratitude to my mother, my brothers, my sister, and my late father for their continuous and unparalleled love.

Last but not least, my wife Randa and my son Julian, thank you for your love, understanding, support, and joy.

Frankfurt, 16.09.2017

Kurzfassung

Richtungsschätzung (englisch: Direction-of-arrival, kurz DOA, estimation), mittels teilkalibrierter Sensorgruppen (Arrays) bestehend aus mehreren Untergruppen (Subarrays), ist ein Schätzproblem in verschiedenen praktischen Anwendungen, wie Radar, Sonar und Reflexionsseismik. Aktuelle DOA Schätzalgorithmen benötigen die Messdaten aller Sensoren an einem Bearbeitungszentrum (englisch: processing center, kurz PC). Die Rechenleistung des PCs und die Kommunikationsbandbreite der Untergruppen nehmen mit der Anzahl der Sensoren zu. Solche zentralisierten Algorithmen skalieren eher schlecht mit der Anzahl der Sensoren. In dieser Arbeit werden dezentralisierte DOA Schätzalgorithmen für teilweise kalibrierte Sensorgruppen vorgestellt, mit dem Ziel die Nachteile der zentralisierten Algorithmen zu vermeiden. In der dezentralisierten DOA Schätzung wird davon ausgegangen, dass jede Untergruppe eine mäßige Rechenleistung besitzt und mit den Untergruppen in ihrer Nähe kommunizieren kann. Anstatt die Rohdaten an das PC zu senden, verarbeiten die Untergruppen ihre Messungen und kommunizieren untereinander, um das Schätzproblem zu bewältigen. In dieser Dissertation wird eine dezentralisierte DOA Schätzung aus der Kovarianz der Messungen in zwei Verarbeitungsschemata, nämlich eine kohärente und nicht-kohärente Verarbeitung, berücksichtigt. Bei der kohärenten Verarbeitung ist die gesamte Array-Kovarianzmatrix einschließlich der Zwischengruppen-Kovarianzmatrizen verfügbar, während nur die Untergruppen-Kovarianzmatrizen in der nicht-kohärenten Verarbeitung verfügbar sind. Die Genauigkeit der DOA Schätzung bei der kohärenten Verarbeitung ist besser als die der nicht-kohärenten Verarbeitung, da mehr Daten in der kohärenten Verarbeitung verfügbar sind, nämlich die Zwischengruppen-Kovarianzmatrizen. Die kohärente Verarbeitung ist jedoch restriktiver als die nicht-kohärente Verarbeitung, insbesondere müssen die Untergruppen für die kohärente Verarbeitung zeitlich synchronisiert sein.

Bei der kohärenten Verarbeitung lässt sich mithilfe der dezentralisierten Power Methode (DPM) bei der Eigenwertzerlegung der Stichprobenvarianz-Matrix eine dezentral DOA Schätzung realisieren. Die Leistungsanalyse der DPM wird durchgeführt. Ein analytischer Ausdruck der Varianz der Eigenvektoren und Eigenwerte wird bestimmt, der für die Berechnung des mittleren quadratischen Fehlers (englisch: mean square error, kurz MSE) der unterraumbasierten Schätzer benötigt wird. Weiterhin wird der dezentralisierte ESPRIT Algorithmus eingeführt, der vollständig dezentralisierte DOA Schätzungen mittels der DPM liefert. Ein asymptotischer, analytischer Ausdruck des MSE von DOA Schätzern mit dem dezentralisierten ESPRIT Algorithmus wird abgeleitet. Ähnlich wie bei dem herkömmlichen ESPRIT Algorithmus benötigt der dezentralisierte ESPRIT Algorithmus eine verschiebungsinvariante Sensorgruppenanordnung. Unter

Verwendung der Interpolation wird der dezentralisierte ESPRIT Algorithmus auf beliebige Array-Geometrien verallgemeinert. Der dezentralisierte ESPRIT Algorithmus hat folgende Nachteile, die durch die DPM verursacht werden: 1.) der große Kommunikationsaufwand, der von der DPM benötigt wird, um jeden Eigenvektor zu berechnen, 2.) die Power Methode ist ein Batch-Verarbeitungs Algorithmus, während z.B. bei Tracking-Anwendungen Online-Algorithmen bevorzugt werden. Um diese Nachteile zu vermeiden werden zwei neue dezentralisierte Eigenwertzerlegungs-algorithmen präsentiert, die einen niedrigeren Kommunikationsaufwand und eine Online-Verarbeitung der Eigenvektoren und Eigenwerte der Kovarianzmatrix ermöglichen. Der dezentralisierte ESPRIT Algorithmus erfordert, dass die Anzahl der Quellen im Voraus verfügbar ist. Es wird ein dezentraler Quell-Detektionsalgorithmus eingeführt, der im Gegensatz zu den herkömmlichen Quell-Detektionsalgorithmen nicht die Berechnung aller Eigenwerte der Kovarianzmatrix erfordert. Als Alternative wird für vollständig kalibrierte Arrays der dezentralisierte Root-MUSIC Algorithmus eingeführt, der die Struktur des Arrays ausnutzt. Ein asymptotischer, analytischer Ausdruck der MSE von DOA Schätzungen, die aus dem dezentralisierten Root-MUSIC Algorithmus erhalten werden, wird abgeleitet.

Für die nicht-kohärente Verarbeitung werden zwei DOA Schätzer vorgestellt, nämlich der Maximum Likelihood Schätzer (englisch: Maximum Likelihood estimator, kurz MLE) und ein rechnerisch einfacher Ansatz, der auf der spärlichen Signaldarstellung (englisch: sparse signal representation, kurz SSR) basiert. Eine hinreichende Bedingung für die eindeutige Identifizierbarkeit der Quellen in dem nicht-kohärenten Verarbeitungsschema wird hergeleitet. Unter schwachen Bedingungen wird bewiesen, dass mit dem nicht-kohärenten System von Untergruppen mehr Quellen identifiziert werden können als mit jedem Untergruppe alleine. Diese Eigenschaft der nicht-kohärenten Verarbeitung wurde bisher nicht untersucht. Darüber hinaus wird die Cramér-Rao Schranke (englisch: Cramér-Rao Bound, kurz CRB) für das nicht-kohärente Messmodell abgeleitet, die zur Bewertung der Leistung der entwickelten DOA Schätzer dient. Das Verhalten des CRB bei hohem Signal-Rausch-Verhältnis (englisch: signal-to-noise ratio, kurz SNR) wird analysiert. Im Gegensatz zur kohärenten Verarbeitung wird bewiesen, dass bei hohem SNR die CRB sich nur dann gegen Null konvergiert, wenn mindestens eine einzelne Untergruppe die Quellen identifizieren kann. Schließlich wird das herkömmliche nicht-kohärente DOA Schätzungsszenario betrachtet, bei dem die Sensoren alle Untergruppen lineare und äquidistant angeordnet sind und die Quellen alleine identifizieren können. Zwei DOA Schätzalgorithmen, die die herkömmlichen nicht-kohärenten DOA Schätzer in ihrer Leistungsfähigkeit übertreffen, werden vorgestellt.

Abstract

Direction-of-arrival (DOA) estimation using partly calibrated arrays composed of multiple subarrays is employed in various practical applications, such as radar, sonar, and seismic exploration. The state-of-the-art DOA estimation algorithms require the measurements of all sensors to be available at a processing center (PC). The processing power of the PC and the communication bandwidth of the subarrays increase with the number of sensors. Thus, such centralized algorithms do not scale well with the number of sensors. In this thesis, decentralized DOA estimation algorithms for partly calibrated arrays are introduced to avoid the drawbacks of the centralized algorithms. In decentralized DOA estimation, each subarray is assumed to possess modest processing power and to be able to communicate with the subarrays in its vicinity. Rather than sending the raw measurement to the PC, the subarrays process their measurements and communicate among each other to achieve the estimation task. In this dissertation, decentralized DOA estimation from the second order statistics of the measurements in two processing schemes, namely, coherent and non-coherent processing is considered. In coherent processing, the whole array covariance matrix including the inter-subarray covariance matrices is available, whereas only the subarray covariance matrices are available in non-coherent processing. The DOA estimation performance of coherent processing is superior to that of non-coherent processing, since more data is available in coherent processing, that is the inter-subarray covariance matrices. However, coherent processing is more restrictive than non-coherent processing, e.g., for coherent processing the subarrays must be synchronized in time.

For coherent processing, decentralized DOA estimation is achieved based on the recently introduced decentralized power method for the eigendecomposition of the sample covariance matrix. Performance analysis of the decentralized power method is presented. An analytical expression of the second order statistics of the eigenvectors and eigenvalues obtained from the decentralized power method, which is required for computing the mean square error (MSE) of subspace-based estimators, is derived. Further, the decentralized ESPRIT algorithm, which yields fully decentralized DOA estimates based on the decentralized power method, is introduced. An asymptotic analytical expression of the MSE of DOA estimators using the decentralized ESPRIT algorithm is derived. Similar to the conventional ESPRIT algorithm, the decentralized ESPRIT algorithm requires a shift-invariant array structure. Using interpolation, the decentralized ESPRIT algorithm is generalized to arbitrary array geometries. The decentralized ESPRIT algorithm inherits the following shortcomings of the decentralized power method: (1) the large communication cost required by the power method to compute each eigenvector, (2) the power method is a batch-based algorithm, whereas in

tracking applications, online algorithms are favored. To mitigate the aforementioned shortcomings, two decentralized eigendecomposition algorithms are proposed, which achieve lower communication cost and online update of the eigenvectors and eigenvalues of the measurement covariance matrix. The decentralized ESPRIT algorithm requires the number of sources to be available beforehand. A decentralized source enumeration algorithm is introduced, which in contrast to the conventional source enumeration algorithms, does not require the computation of all the eigenvalues of the measurement covariance matrix. As an alternative, for fully calibrated arrays, the decentralized Root-MUSIC algorithm is introduced, which exploit the structure of the array. An asymptotic analytical expression of the MSE of DOA estimates obtained from the decentralized Root-MUSIC algorithm is derived.

For non-coherent processing, two DOA estimators are presented, namely, the Maximum Likelihood estimator (MLE) and a computationally simpler approach based on sparse signal representation (SSR). A sufficient condition for the unique identifiability of the sources in the non-coherent processing scheme is presented, which shows that under mild conditions, the number of sources identifiable by the system of subarrays is larger than the number identifiable by each individual subarray. This property of non-coherent processing has not been investigated before, where the state-of-the-art non-coherent DOA estimation algorithms fail if the individual subarrays can not identify the sources. Moreover, the Cramér-Rao Bound (CRB) for the non-coherent measurement model is derived and is used to assess the performance of the proposed DOA estimators. The behaviour of the CRB at high signal-to-noise ratio (SNR) is analyzed. In contrast to coherent processing, the in this case CRB approaches zero at high SNR only if at least one subarray can identify the sources individually. Finally, the conventional non-coherent DOA estimation scenario, where all the subarrays are uniform linear and can identify the sources, is considered. Two DOA estimation algorithms, which outperform the state-of-the-art non-coherent DOA estimators, are presented.

Contents

Chapter 1

Introduction

The problem of determining the direction-of-arrival (DOA), i.e., the azimuth and the elevation angles, of signals emitted by some sources using sensor arrays appears in a variety of important applications. For example, in many radar applications, estimating the direction of objects, such as, airplanes, is of interest. This is achieved by finding the DOAs of the electromagnetic-signals reflected from these objects at an antenna array. Other application areas for DOA estimation are sonar, seismic exploration, and astronomy [VT02]. In DOA estimation applications, high angular resolution and the capability of identifying large number of sources are required. The angular resolution of a sensor array is limited by its aperture size, which is related to the largest distance between any two sensors at this array. Thus, the wider the area occupied by the sensors of the array, the higher is its angular resolution. Generally, using the second order statistics of the array output, a number of sources which is smaller than the number of the sensors in the array by one can be identified. Consequently, using an array composed of large number of sensors, which are distributed over a wide area, both DOA estimation requirements, namely, high angular resolution and the ability to identify large number of sources, can be accomplished. In conventional DOA estimation algorithms, raw measurements of all the sensors are collected at a processing center (PC), on which the DOA estimation algorithms are implemented. This processing scheme is referred to as centralized processing. Such centralized scheme requires a powerful PC, which is capable of processing the raw measurements of all sensors. Further, all sensors of the array must be able to transfer their measurements to the PC using wired or wireless communication. Note that such a centralized processing scheme does not scale well with the number of sensors, since expanding the array, i.e., adding new sensors to the array, yields increased communication cost and requires a more powerful PC. Further, in large arrays, calibrating the sensor positions, i.e., finding the exact distances between the sensors of the array, is more difficult than in small arrays, which can be calibrated in laboratories. Moreover, in large arrays, calibration errors due to temperature, synchronization clock distribution, etc are more severe than in small arrays.

Decentralized DOA estimation mitigates the aforementioned drawbacks of conventional DOA estimation algorithms. In decentralized DOA estimation, partly calibrated arrays composed of multiple subarrays are considered. Each subarray is assumed to be fully calibrated, whereas the displacements between the subarrays are assumed to be

unknown. Thus, the deployment of partly calibrated arrays over a wide area is easier than that of fully calibrated arrays. Further, the subarrays are assumed to possess modest processing power which can be used to perform DOA estimation without the need for a PC or with a less powerful (processing power-wise) PC. Moreover, each subarray can communicate with its neighboring subarrays to achieve the estimation task while reducing the required communication cost compared to the centralized processing scheme. The aforementioned decentralized processing scheme scales well with the number of sensors since each subarray added to the network brings new local processing power and communication opportunities with its neighboring subarrays.
Wireless sensor networks (WSNs) provide an ideal infrastructure for executing the decentralized DOA estimation algorithms. A WSN consists of multiple sensing nodes (SNs), where each SN is equipped with one or more sensors. Further, the SNs possess processing power and can communicate among each other using a wireless link [Sto05, SCN12]. WSNs are increasingly being employed in a variety of applications. For example, monitoring and sensing in harsh environments, such as factories, forests, underwater, and offshore platforms, where they provide fast detection of events, such as fire in a wood, earthquakes, and machine failure. In many of the aforementioned applications, localization, i.e., determining the position of the SNs, is crucial, since data and information are useless if the SNs are not aware of their positions. Decentralized DOA estimation can be used to estimate the location of the SNs in a WSN. In [Sto05], DOA estimation is employed for localization in WSNs where each SN achieves individually the DOA estimation task. Decentralized DOA estimation can extend the precision of this localization algorithm by cooperation between multiple neighboring SNs.

1.1 State-of-the-Art

Before the literature on DOA estimation is reviewed, the main categories of sensing networks and processing schemes are introduced. Typically sensing networks are classified based on their processing topologies into [Mit12, TM03]:

1. *Centralized*: where all SNs forward their raw measurements to a PC which carries out the estimation task.

2. *Decentralized (without PC)*: where the SNs communicate among each other to perform the estimation task and no PC is required.

3. *Decentralized with PC*[1]: which is a mix of both aforementioned schemes, where the SNs do not communicate the raw data to the PC but sufficient statistics, which are of less size than the raw data, are computed at the SNs and sent to the PC.

Based on the covariance of the measurements from different SNs the processing scheme can be described as:

1. *Coherent*: where the covariance of the measurements from different SNs can be computed and it is non-zero. In other words, the measurement covariance matrix of the whole array can be estimated from the measurements. This usually requires that the SNs are time synchronized.

2. *Non-coherent*: the covariance of the measurements from different SNs can not be computed or it is zero, e.g., the SNs are not synchronized in time[2]. In other words, only the covariance matrices of the SNs can be estimated from the measurements.

In the following, the literature of DOA estimation, is briefly reviewed. The existing DOA estimation algorithms are classified as coherent or non-coherent algorithms. Firstly, we review the literature on coherent DOA estimation. Coherent algorithms, which are all assume a centralized network topology, are revised. Secondly, non-coherent DOA estimation algorithms are shortly discussed. These algorithms perform decentralized DOA estimation with PC.

Coherent DOA estimation algorithms dates more than half a century back [KV96]. The beamformer, which is an application of Fourier-based spectral analysis, is one of the early used DOA estimation algorithms. Later, adaptive bearmformers, e.g., Capon beamformer [Cap69], were applied to enhance the ability to resolve closely spaced sources. The performance of the beamformers is directly dependent upon the array aperture, regardless of the available sample size and the SNR [KV96]. To increase the resolution of spectral-based DOA estimation methods beyond the classical Fourier limit, subspace-based DOA estimation algorithms were developed. Starting from the work of Pisarenko in the seventies [Pis73], several high resolution subspace-based algorithms have been proposed in literature, e.g., MUSIC [Sch86], Root-MUSIC [Bar83],

[1] In [Mit12, TM03], this scheme is referred to as hierarchical, where in [SA89, SS92] it is called decentralized, from which the name decentralized with PC is derived.

[2] Note that in this scheme, the data of each SN is coherently processed. Thus, this processing type is partly coherent. Nevertheless, in this dissertation (similar to [RF04]), such a scheme is referred to as non-coherent.

MODE [SNS95], ESPRIT [RK89], WSF [VOK91], RARE [PGW02], and the algorithms proposed in [PPG11] and [PP11]. Subspace-based algorithms are computationally attractive and achieve higher resolution compared to the beamformers. However, from a statistical viewpoint, they may be very inefficient [SS90]. The MLE [BM86,SA89,SS90], on the other hand, is efficient, however, it requires multi-dimensional search over the directions of the sources. Thus, it can not be applied if the number of sources is large. Recently, sparse signal representation (SSR) [Tib96, DET06] has attracted much attention in DOA estimation applications. One important advantage of SSR is that it performs well in the low sample size regime. Furthermore, using the ℓ_1 norm relaxation, the SSR approach can be casted as a convex optimization problem [BV04, CR12]. SSR approaches for DOA estimation are based on either the direct data, e.g., [MÇW05, GR97], or the sample covariance matrix, e.g., [SPP16, SBL11, SPP14].

Non-coherent DOA estimation algorithms available in the literature are generalizations of subspace-based ones. In [WK85a,RF04], the MUSIC algorithm is generalized to non-coherent processing where the subarrays are assumed to estimate their signal subspaces locally and send them to the PC. In [SS92], another version of the MUSIC algorithm for non-coherent processing is analyzed. In this algorithm, the subarrays send the locally estimated DOAs and their estimated variances to the PC. A similar method which is robust against uncertainties in the statistical distribution of the noise is presented in [LKM90]. In [SNS95], DOA estimation is perform using the MODE algorithm individually in each subarray. At the PC, the DOA estimates are optimally combined as in [SS92]. Although the algorithms proposed in [WK85a, LKM90, SS92, SNS95, RF04] are designed for non-coherent processing and can be used for decentralized DOA estimation with PC, they all assume that each subarray can locally identify all the sources. Thus, in these algorithms, using multiple subarrays does not increase the number of identifiable sources, but it results in a more accurate DOA estimates.

Detecting the number of the sources is closely related to the DOA estimation problem. Most DOA estimation algorithms require the number of sources to be known beforehand. Source enumeration algorithms available in the literature, e.g., the MDL [WK85b] and the bootstrap-based detector [BZP02], are centralized and require the computation of all the eigenvalues of the measurement covariance matrix.

1.2 Aims, Contributions, and Overview

In this dissertation, decentralized DOA estimation using partly calibrated arrays composed of multiple fully calibrated subarrays is considered. The subarrays are assumed

to possess some limited processing capabilities which are used for decentralized DOA estimation with or without PC. The focus of this thesis is on the scenarios in which none of the subarrays can achieve the DOA estimation task individually, i.e., when the number of the sensors at the subarrays is smaller than the number of the sources. Decentralized DOA estimation in such scenarios is challenging and many existing algorithms fail. Further, decentralized DOA estimation in such scenarios is practically important since subarrays may only have a few sensors, which limits the number of identifiable sources otherwise. The DOA estimation algorithms introduced in this thesis can be classified into two main groups:

1. Algorithms which are based on coherent processing in a decentralized (without PC) network topology.

2. Non-coherent processing based algorithms in a decentralized with PC network topology.

This thesis is organized in six chapters and four appendices. Chapters 1 and 2 contain overview of the thesis and state-of-the-art DOA estimation algorithms, respectively. The contributions of this dissertation are introduced in chapters 3–5. Coherent decentralized DOA estimation and eigendecomposition algorithms are presented in Chapter 3 and Chapter 4, respectively. Non-coherent DOA estimation algorithms are presented in Chapter 5. Chapter 6 is summary and outlook. Mathematical proofs are intruded in the appendices. In the following, an overview of chapters 2–5 is provided.

Chapter 2 contains the signal model and overview of the conventional DOA estimation algorithms. Further, the decentralized eigendecomposition, which can be used for subspace-based decentralized DOA estimation, is revised.

In **Chapter 3**, a performance analysis of the decentralized eigendecomposition based on the decentralized power method (DPM) is presented. An asymptotic analytical expression of the second order statistics of the eigenvalues and eigenvectors obtained from the DPM is introduced. Two decentralized DOA estimation algorithms are proposed, namely, the decentralized ESPRIT (DESPRIT) and the decentralized Root-MUSIC algorithms. Based on the analysis of the decentralized eigendecomposition using the DPM, an asymptotic analytical expression of the MSE of DOA estimation using the aforementioned algorithms is derived. Further, the interpolated DESPRIT algorithm is introduced, which performs decentralized DOA estimation for arbitrary array geometries, whereas the DESPRIT algorithm requires a shift-invariant array. Moreover, a decentralized source enumeration algorithm which does not require the computation of all eigenvalues of the sample covariance matrix is presented. Therefore, this algorithm

requires lower computational cost as compared to conventional source enumeration algorithms, e.g., the MDL [WK85b], which require the computation of all eigenvalues of the sample covariance matrix.

In **Chapter 4**, the decentralized eigendecomposition is considered. Based on the Lanczos method, a decentralized eigendecomposition algorithm is proposed, which requires lower communication cost as compared to the DPM. Moreover, an online decentralized generalized eigendecomposition algorithm is introduced which updates the sample eigenvectors adaptively with the arrival of new snapshots. The online DOA estimation is implemented using the decentralized generalized eigendecomposition. In this implementation, estimates of the DOAs are updated at each time instant, whereas using the DPM, batch-based DOA estimation is achieved.

The second group of contributions presented in **Chapter 5** corresponds to decentralized non-coherent DOA estimation. A bound on the maximum number of identifiable sources is introduced. This bound shows that in certain array geometries, it is possible to identify more sources than each subarray can identify locally. Furthermore, two DOA estimation approaches are proposed: 1) the MLE and 2) a computationally simpler DOA estimation approach based on SSR. Moreover, the CRB for DOA estimation using non-coherent processing is derived and analyzed. In contrast to the state-of-the-art algorithms in [WK85a,LKM90,SS92,SNS95,RF04], the assumption that each subarray can identify the sources is not made during the previous contributions. For the special case when all the subarrays are uniform linear and can identify the sources two DOA estimation algorithms which outperform the ones in [WK85a,LKM90,SS92,SNS95,RF04] are presented.

Publications

The following publications have been produced during the period of the doctoral candidacy.

Internationally Refereed Journal Articles

- W. Suleiman, M. Pesavento, and A. M. Zoubir "Performance Analysis of the Decentralized Eigendecomposition and ESPRIT Algorithm," *IEEE Transactions on Signal Processing*, vol. 64, no. 9, pp. 2375-2386, May 2016.
- W. Suleiman, P. Parvazi, M. Pesavento, and A. M. Zoubir, "Non-Coherent Direction-of-Arrival Estimation Using Partly Calibrated Arrays," *arXiv:1704.06000 [stat.AP]*, Apr. 2017.

Internationally Refereed Conference Papers

- W. Suleiman, M. Pesavento, and A. M. Zoubir, "Decentralized Direction Finding Using Partly Calibrated Arrays," in *European signal processing conference (EUSIPCO)*, Sep. 2013, pp. 1-5.
- W. Suleiman and P. Parvazi, "Search-Free Decentralized Direction-of-Arrival Estimation Using Common Roots for Non-Coherent Partly Calibrated Arrays," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014, pp. 2292-2296.
- W. Suleiman, P. Parvazi, M. Pesavento, and A. M. Zoubir, "Decentralized Direction Finding Using Lanczos Method," in *IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM)*, June 2014, pp. 9-12.
- W. Suleiman, M. Pesavento, and A. M. Zoubir, "Decentralized Cooperative DOA Tracking Using Non-Hermitian Generalized Eigendecomposition," in *European signal processing conference (EUSIPCO)*, Jun 2015.
- W. Suleiman, M. Pesavento, and A. M. Zoubir, "Performance Analysis of Direction-of-Arrival Estimation Using the Decentralized Root-MUSIC," in *IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, Dec. 2015.
- W. Suleiman, M. Pesavento, and A. M. Zoubir, "Decentralized Cooperative Detection Based on Averaging Consensus," in *IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM)*, July 2016, pp. 1-5.
- W. Suleiman, A. A. Vaheed, M. Pesavento, and A. M. Zoubir, "Decentralized Direction-of-Arrival Estimation for Arbitrary Array Geometries," in *European Signal Processing Conference (EUSIPCO)*, Aug. 2016, pp. 1921-1925.
- W. Suleiman, M. Pesavento, and A. M. Zoubir, "Performance Analysis of Eigenvalue Based Distributed Spectrum Sensing," in *International Symposium on Wireless Communication Systems (ISWCS)*, Sep. 2016, pp. 481-485.

Other Contributions

- C. Steffens, W. Suleiman, A. Sorg, and M. Pesavento, "Gridless Compressed Sensing Under Shift-Invariant Sampling.," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Mar. 2017.

- W. Suleiman, C. Steffens, A. Sorg, and M. Pesavento, "Gridless Compressed Sensing for Fully Augmentable Arrays," *European signal processing conference (EUSIPCO)*, Aug. 2017, (Submitted).

Chapter 2

Signal Model and the State-of-the-Art

In this chapter, the signal model which is used throughout the dissertation is introduced. Conventional subspace-based DOA estimation algorithms are reviewed, in particular the MUSIC, Root-MUSIC, and ESPRIT algorithms are revised. Further, DOA estimation using the MLE is discussed. DOA estimation algorithms based on SSR that operate directly on the measurements or on the corresponding sample covariance matrix are briefly reviewed. Finally, the decentralized eigendecomposition based on the PM algorithm and the AC protocol is revised.

2.1 Signal Model

A planar array composed of K subarrays, where the kth subarray comprises of M_k sensors, is considered. The total number of sensors in the array is denoted by

$$M = \sum_{k=1}^{K} M_k. \tag{2.1}$$

The assignment of the sensors to the individual subarrays is characterized by the $M \times K$ sensor selection matrix $\boldsymbol{T}$, whose entries are defined as

$$[\boldsymbol{T}]_{i,j} = \begin{cases} 1, & \text{if the } i\text{th sensor belongs to the } j\text{th subarray} \\ 0, & \text{otherwise,} \end{cases} \tag{2.2}$$

where $i = 1, \ldots, M$ and $j = 1, \ldots, K$. Without loss of generality, the first sensor of the kth subarray is considered as the reference sensor of this subarray. The planar location of the ith sensor of the kth subarray with respect to its first sensor, denoted by $\boldsymbol{\zeta}'_{k,i} \in \mathbb{R}^{2\times 1}$, for $i = 2, \ldots, M_k$, $\boldsymbol{\zeta}'_{k,1} = [0, 0]^T$, is assumed to be known. In other words, the subarrays are fully calibrated. The displacement of the first sensor of the kth subarray with respect to the first sensor in the first subarray, denoted by $\boldsymbol{\zeta}_k \in \mathbb{R}^{2\times 1}$, for $k = 2, \ldots, K$, $\boldsymbol{\zeta}_1 = [0, 0]^T$, is assumed to be unknown. Fig. 2.1 demonstrates the topology of the array and sensor displacements.

Signals of L narrow-band far-field sources impinge onto the array from directions $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_L]^T$, where, without loss of generality, the signals are assumed to be in the

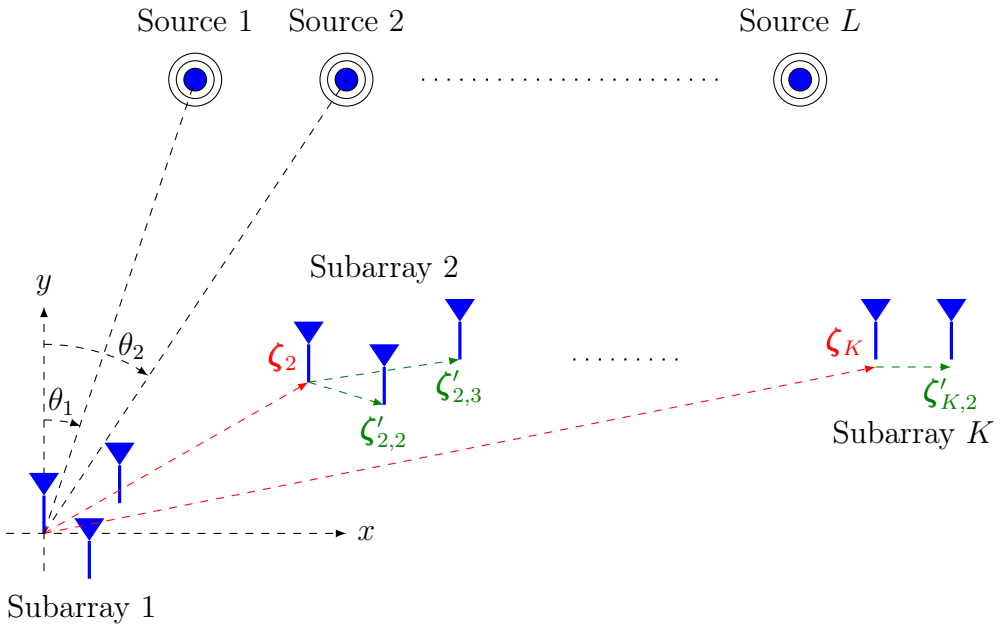

Fig. 2.1. Partly calibrated array composed of K fully calibrated subarrays.

same plane as the sensors, i.e., elevation is zero. The response of the kth subarray corresponding to a source at direction θ is given by

$$\boldsymbol{a}_k(\theta) = \boldsymbol{v}_k(\theta)\phi(\theta, \boldsymbol{\zeta}_k), \tag{2.3}$$

where $\phi(\theta, \boldsymbol{\zeta}_k) = \exp(\jmath \frac{2\pi}{\lambda_c} \boldsymbol{\zeta}_k^T \boldsymbol{\nu}(\theta))$ is an unknown phase shift, λ_c is the wavelength corresponding to the signal carrier frequency, and $\boldsymbol{\nu}(\theta) = [\sin(\theta), \cos(\theta)]^T$. The vector $\boldsymbol{v}_k(\theta)$ is defined as

$$\boldsymbol{v}_k(\theta) = [1, \exp(\jmath \frac{2\pi}{\lambda_c} \boldsymbol{\zeta}'^T_{k,2} \boldsymbol{\nu}(\theta)), \ldots, \exp(\jmath \frac{2\pi}{\lambda_c} \boldsymbol{\zeta}'^T_{k,M_k} \boldsymbol{\nu}(\theta))]^T, \tag{2.4}$$

where, in contrast to the phase shift $\phi(\theta, \boldsymbol{\zeta}_k)$, the manifold vector $\boldsymbol{v}_k(\theta)$ is fully known as a function of θ. In [SG04], it has been shown that the array response of (2.3) models not only unknown inter-subarray displacements but it also models other inter-subarray calibration errors, such as imperfect inter-subarry time synchronization, unknown propagation channel mismatches between subarrays, or a combination of the aforementioned effects.

The vector of the baseband signals received at the kth subarray $\boldsymbol{x}_k(t)$ is given by

$$\boldsymbol{x}_k(t) = \boldsymbol{A}_k(\boldsymbol{\theta}, \boldsymbol{\zeta}_k)\boldsymbol{s}(t) + \boldsymbol{n}_k(t) \tag{2.5}$$

where $\boldsymbol{n}_k(t)$ is the vector containing the complex circular Gaussian sensor noise with zero-mean and variance σ^2. The vector $\boldsymbol{s}(t)$ contains the complex circular Gaussian

source signals with zero-mean and covariance

$$\boldsymbol{P} = \mathbb{E}\big(\boldsymbol{s}(t)\boldsymbol{s}^H(t)\big). \tag{2.6}$$

The array manifold matrix $\boldsymbol{A}_k(\boldsymbol{\theta}, \boldsymbol{\zeta}_k) = [\boldsymbol{a}_k(\theta_1, \boldsymbol{\zeta}_k), \ldots, \boldsymbol{a}_k(\theta_L, \boldsymbol{\zeta}_k)]^T$ is written as

$$\boldsymbol{A}_k(\boldsymbol{\theta}, \boldsymbol{\zeta}_k) = \boldsymbol{V}_k(\boldsymbol{\theta})\boldsymbol{\Phi}_k(\boldsymbol{\theta}, \boldsymbol{\zeta}_k), \tag{2.7}$$

where the matrix

$$\boldsymbol{V}_k(\boldsymbol{\theta}) = [\boldsymbol{v}_k(\theta_1), \ldots, \boldsymbol{v}_k(\theta_L)] \tag{2.8}$$

depends only on the DOAs, whereas the diagonal matrix

$$\boldsymbol{\Phi}_k(\boldsymbol{\theta}, \boldsymbol{\zeta}_k) = \operatorname{diag}\big(\phi_{k,1}, \ldots, \phi_{k,L}\big), \tag{2.9}$$

where $\phi_{k,l} = \phi(\theta_l, \boldsymbol{\zeta}_k)$, depends on the DOAs and the unknown displacement $\boldsymbol{\zeta}_k$. The measurement vector of the array is

$$\boldsymbol{x}(t) = \boldsymbol{A}(\boldsymbol{\theta}, \boldsymbol{\zeta})\boldsymbol{s}(t) + \boldsymbol{n}(t), \tag{2.10}$$

where $\boldsymbol{x}(t) = [\boldsymbol{x}_1^T(t), \ldots, \boldsymbol{x}_K^T(t)]^T$, $\boldsymbol{A}(\boldsymbol{\theta}, \boldsymbol{\zeta}) = [\boldsymbol{A}_1^T(\boldsymbol{\theta}, \boldsymbol{\zeta}_1), \ldots, \boldsymbol{A}_K^T(\boldsymbol{\theta}, \boldsymbol{\zeta}_K)]^T$, $\boldsymbol{n}(t) = [\boldsymbol{n}_1^T(t), \ldots, \boldsymbol{n}_K^T(t)]^T$, and $\boldsymbol{\zeta} = [\boldsymbol{\zeta}_1^T, \ldots, \boldsymbol{\zeta}_K^T]^T$. In the following, the dependency on $\boldsymbol{\theta}$ and $\boldsymbol{\zeta}$ are dropped for notation convenience. Assuming that the source and the noise signals are independent, the covariance matrix of the array measurements $\boldsymbol{R} = \mathbb{E}\big(\boldsymbol{x}(t)\boldsymbol{x}^H(t)\big)$ is reduced to

$$\boldsymbol{R} = \boldsymbol{A}\boldsymbol{P}\boldsymbol{A}^H + \sigma^2\boldsymbol{I}_M. \tag{2.11}$$

The eigendecomposition of the measurement covariance matrix $\boldsymbol{R}$ is written as

$$\boldsymbol{R} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^H, \tag{2.12}$$

where $\boldsymbol{\Lambda} = \operatorname{diag}\big(\lambda_1, \ldots, \lambda_M\big)$ is a diagonal matrix that contains the eigenvalues of the matrix $\boldsymbol{R}$ ordered in descending order, i.e., $\lambda_1 \geq \lambda_2 \cdots \geq \lambda_L > \lambda_{L+1} = \cdots = \lambda_M = \sigma^2$, $\boldsymbol{U} = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_M]$, and $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_M$ are the eigenvectors of the matrix $\boldsymbol{R}$ corresponding to the eigenvalues $\lambda_1, \ldots, \lambda_M$. For the later use, the matrix

$$\boldsymbol{\Xi}_i = \boldsymbol{U}_{-i}\boldsymbol{\Gamma}_i^{-1}\boldsymbol{U}_{-i}^H, \tag{2.13}$$

is defined, where $\boldsymbol{U}_{-i} = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_{i-1}, \boldsymbol{u}_{i+1}, \ldots, \boldsymbol{u}_M]$ and $\boldsymbol{\Gamma}_i = \operatorname{diag}\big(\lambda_1 - \lambda_i, \ldots, \lambda_{i-1} - \lambda_i, \lambda_{i+1} - \lambda_i, \ldots, \lambda_M - \lambda_i\big)$. The eigendecomposition of (2.12) is partitioned as

$$\boldsymbol{R} = \boldsymbol{U}_{\mathrm{s}}\boldsymbol{\Lambda}_{\mathrm{s}}\boldsymbol{U}_{\mathrm{s}}^H + \boldsymbol{U}_{\mathrm{n}}\boldsymbol{\Lambda}_{\mathrm{n}}\boldsymbol{U}_{\mathrm{n}}^H, \tag{2.14}$$

where $\boldsymbol{U}_{\mathrm{s}} = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_L]$ and $\boldsymbol{U}_{\mathrm{n}} = [\boldsymbol{u}_{L+1}, \ldots, \boldsymbol{u}_M]$ are the signal and noise eigenvector matrices, respectively. The matrices $\boldsymbol{\Lambda}_{\mathrm{s}} = \operatorname{diag}\big(\lambda_1, \ldots, \lambda_L\big)$ and $\boldsymbol{\Lambda}_{\mathrm{n}} = \operatorname{diag}\big(\lambda_{L+1}, \ldots, \lambda_M\big)$ are the signal and noise eigenvalue matrices, respectively.

In practice, the true covariance matrix $\boldsymbol{R}$ is not available, and can only be estimated from N observations of $\boldsymbol{x}(t)$, for $t = 1, \ldots, N$, as

$$\hat{\boldsymbol{R}} = \frac{1}{N} \sum_{t=1}^{N} \boldsymbol{x}(t)\boldsymbol{x}^H(t). \tag{2.15}$$

The conventional estimator of the sample covariance matrix in (2.15) is referred to as the centralized estimator, since it requires that all measurements (from every subarray) are available at a PC. Similar to (2.14), the eigendecomposition of the sample covariance matrix is written as

$$\hat{\boldsymbol{R}} = \hat{\boldsymbol{U}}_{\mathrm{s}}\hat{\boldsymbol{\Lambda}}_{\mathrm{s}}\hat{\boldsymbol{U}}_{\mathrm{s}}^H + \hat{\boldsymbol{U}}_{\mathrm{n}}\hat{\boldsymbol{\Lambda}}_{\mathrm{n}}\hat{\boldsymbol{U}}_{\mathrm{n}}^H, \tag{2.16}$$

where $\hat{\boldsymbol{U}}_{\mathrm{s}}$, $\hat{\boldsymbol{\Lambda}}_{\mathrm{s}}$, $\hat{\boldsymbol{U}}_{\mathrm{n}}$, and $\hat{\boldsymbol{\Lambda}}_{\mathrm{n}}$ are the estimates of $\boldsymbol{U}_{\mathrm{s}}$, $\boldsymbol{\Lambda}_{\mathrm{s}}$, $\boldsymbol{U}_{\mathrm{n}}$, and $\boldsymbol{\Lambda}_{\mathrm{n}}$, respectively. Moreover, the sample estimates of $\boldsymbol{\Lambda}$, $\boldsymbol{U}$, λ_i, and $\boldsymbol{u}_i$ in (2.12), for $i = 1, \ldots, M$, obtained form the eigendecomposition of the sample covariance matrix are denoted as $\hat{\boldsymbol{\Lambda}}$, $\hat{\boldsymbol{U}}$, $\hat{\lambda}_i$, and $\hat{\boldsymbol{u}}_i$, respectively.

2.2 DOA Estimation for Fully Calibrated Arrays

In *fully calibrated* arrays, the sensor displacements $\boldsymbol{\zeta}_k$, for $k = 1, \ldots, K$, are considered to be known, i.e., the array manifold matrix $\boldsymbol{A}$ is analytically described as a function of the DOAs $\boldsymbol{\theta}$. In the following, DOA estimation algorithms for fully calibrated arrays are reviewed.

2.2.1 Subspace-Based Algorithms

Subspace-based DOA estimation algorithms are computationally efficient and possess the high resolution property. In the sequel, The MUSIC [Sch86] and Root-MUSIC [Bar83] algorithms, which represent this type of DOA estimation methods are revised.

2.2.1.1 The MUSIC Algorithm

The basic idea behind the MUSIC algorithm [Sch86] is the following. The signal eigenvector matrix $\boldsymbol{U}_{\mathrm{s}}$ spans the same subspace as the matrix $\boldsymbol{A}$ and the noise eigenvector matrix $\boldsymbol{U}_{\mathrm{n}}$ spans the subspace orthogonal to that spanned by the matrices $\boldsymbol{A}$ and $\boldsymbol{U}_{\mathrm{s}}$,

i.e., $\boldsymbol{U}_{\mathrm{n}}^H \boldsymbol{A} = \boldsymbol{0}$. This can be seen from (2.11) and (2.14). Thus, in [Sch86], the MUSIC spatial spectrum at direction θ is defined as

$$\mathcal{S}_{\mathrm{MUSIC}}(\theta) = \frac{\boldsymbol{a}^H(\theta)\boldsymbol{a}(\theta)}{\boldsymbol{a}^H(\theta)\hat{\boldsymbol{\Pi}}_{\mathrm{n}}\boldsymbol{a}(\theta)}, \tag{2.17}$$

where the orthogonal projection onto the noise subspace is estimated as

$$\hat{\boldsymbol{\Pi}}_{\mathrm{n}} = \hat{\boldsymbol{U}}_{\mathrm{n}}\hat{\boldsymbol{U}}_{\mathrm{n}}^H = \boldsymbol{I}_M - \hat{\boldsymbol{U}}_{\mathrm{s}}\hat{\boldsymbol{U}}_{\mathrm{s}}^H. \tag{2.18}$$

Note that when θ corresponds to a source direction, the corresponding array response vector $\boldsymbol{a}(\theta)$, which belongs to the signal subspace, is orthogonal to the noise subspace. Thus, the denominator of (2.17) is very small leading to a large value of the MUSIC spectrum $\mathcal{S}_{\mathrm{MUSIC}}(\theta)$. In the MUSIC algorithm, the field of view is sampled at a fine grid and the spectrum of (2.17) is computed over the grid points. The directions corresponding to the L largest peaks in $\mathcal{S}_{\mathrm{MUSIC}}(\theta)$ are the estimates of the DOAs. The MUSIC algorithm is a search-based algorithm, since it requires searching over the field of view, which results in a high computational cost. In the following subsection, the Root-MUSIC algorithm which is a search-free DOA estimation algorithm is reviewed.

2.2.1.2 The Root-MUSIC Algorithm

The Root-MUSIC algorithm for *fully calibrated uniform linear* arrays (ULAs) is introduced in [Bar83]. In such arrays, the response vector, defined in (2.3), is reduced to the Vandermonde vector

$$\boldsymbol{a}(z) = [1, z, z^2, \ldots, z^{M-1}]^T, \tag{2.19}$$

where $z = \exp(\jmath\frac{2\pi}{\lambda_c}\overline{d}\sin\theta)$ and $\overline{d}$ is the distance between adjacent sensors. Thus, the denominator of the right-hand-side of (2.17) can be written as a polynomial $\hat{\mathcal{F}}(z)$ of degree $2(M-1)$ as

$$\hat{\mathcal{F}}(z) = z^{M-1}\boldsymbol{a}^T(1/z)\hat{\boldsymbol{\Pi}}_{\mathrm{n}}\boldsymbol{a}(z), \tag{2.20}$$

where $\hat{\boldsymbol{\Pi}}_{\mathrm{n}}$ is defined in (2.18). The polynomial $\hat{\mathcal{F}}(z)$ is referred to as the Root-MUSIC polynomial. Let $\hat{z}_1, \ldots, \hat{z}_L$ be the L roots of the polynomial $\hat{\mathcal{F}}(z)$ which are inside the unit circle and have the largest magnitude. Then, the DOAs can be estimated as

$$\hat{\theta}_l = \sin^{-1}(\arg(\hat{z}_l)\lambda_c/(2\pi\overline{d})), \tag{2.21}$$

for $l = 1, \ldots, L$ [Bar83]. Note that the Root-MUSIC algorithm does not require search over the field of view, thus, it is a search-free algorithm. Consequently, the computational cost of the Root-MUSIC is much less than that of the MUSIC algorithm. However it is only applicable for ULA geometries.

2.2.2 The Maximum Likelihood Estimator

Under the model (2.10) and the assumptions in Section 2.1 of Gaussian source and noise signals, the MLE for *fully calibrated* arrays can be derived as in [SS90, SA89, BM86]. The concentrated (with respect to the DOAs $\boldsymbol{\theta}$) negative log-likelihood function is given by

$$\mathcal{L}(\boldsymbol{\theta}) = \mathrm{tr}\big((\boldsymbol{I}_M - \boldsymbol{A}(\boldsymbol{A}^H\boldsymbol{A})^{-1}\boldsymbol{A}^H)\hat{\boldsymbol{R}}\big), \tag{2.22}$$

where the array manifold $\boldsymbol{A}$ depends only on $\boldsymbol{\theta}$. The estimate of $\boldsymbol{\theta}$ is obtained as the minimizer of $\mathcal{L}(\boldsymbol{\theta})$ in (2.22), i.e., by solving the minimization problem

$$\min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}). \tag{2.23}$$

The function $\mathcal{L}(\boldsymbol{\theta})$ is highly nonlinear [SS90]. Consequently, finding the global minimizer of (2.23) is an NP-hard problem.

2.2.3 Sparse Signal Representation-Based Algorithms

Sparse signal Representation (SSR) [Tib96, DET06, DH01, DT08, CW08] has recently attracted much attention in DOA estimation applications, see [MÇW05, HM10, SPP14, YC11, AK12]. One important advantage of SSR is that it performs well in the low sample size regime. Furthermore, using the ℓ_1 norm relaxation, the SSR can be cast as a convex optimization problem [BV04, CR12]. Whereas, the aforementioned algorithms, such as the MLE, operate with the non-linearity introduced by the array manifold matrix. SSR approaches for DOA estimation are based on either the direct data, e.g., [MÇW05, GR97, SPP14], or the sample covariance matrix, e.g., [SPP16, SBL11]. In this section, assuming a *fully-calibrated* array, both aforementioned SSR-based DOA estimation approaches are reviewed.

2.2.3.1 Data-Based DOA Estimation

Let $\tilde{\boldsymbol{\theta}}$ be the vector of length G obtained by sampling the field-of-view in $G \gg L$ angular directions

$$\tilde{\boldsymbol{\theta}} = [\tilde{\theta}_1, \ldots, \tilde{\theta}_G]^T. \tag{2.24}$$

Further, let $\tilde{\boldsymbol{A}}$ denote the overcomplete dictionary defined based on the array manifold matrix $\boldsymbol{A}$, i.e.,

$$\tilde{\boldsymbol{A}} = \boldsymbol{A}(\tilde{\boldsymbol{\theta}}) = [\boldsymbol{a}(\tilde{\theta}_1), \ldots, \boldsymbol{a}(\tilde{\theta}_G)]. \tag{2.25}$$

The grid $\tilde{\boldsymbol{\theta}}$ is assumed to be sufficiently fine, such that the true DOAs $\boldsymbol{\theta}$ are on the grid[1] $\tilde{\boldsymbol{\theta}}$. Thus, the overcomplete representation of (2.10) can be written as

$$\boldsymbol{x}(t) = \tilde{\boldsymbol{A}}\,\tilde{\boldsymbol{s}}(t) + \boldsymbol{n}(t), \tag{2.26}$$

where $\tilde{\boldsymbol{s}}(t)$ is the sparse signal vector of size G. The entries of the vector $\tilde{\boldsymbol{s}}(t)$ correspond to the directions $\tilde{\theta}_1, \ldots, \tilde{\theta}_G$, i.e., to the entries of the vector $\tilde{\boldsymbol{\theta}}$. Thus, the ith entry of the sparse signal vector $[\tilde{\boldsymbol{s}}(t)]_i$ is zero unless $\tilde{\theta}_i$ corresponds to a true source direction. Note that the overcomplete dictionary $\tilde{\boldsymbol{A}}$ of (2.26) is completely known, since the array is considered to be fully-calibrated. Assuming that N samples of the array output are available, (2.26) is written in a matrix form as

$$\boldsymbol{X} = \tilde{\boldsymbol{A}}\,\tilde{\boldsymbol{S}} + \boldsymbol{N}, \tag{2.27}$$

where $\boldsymbol{X} = [\boldsymbol{x}(1), \ldots, \boldsymbol{x}(N)]$, $\tilde{\boldsymbol{S}} = [\tilde{\boldsymbol{s}}(1), \ldots, \tilde{\boldsymbol{s}}(N)]$, and $\boldsymbol{N} = [\boldsymbol{n}(1), \ldots, \boldsymbol{n}(N)]$. Since the sparse vectors $\tilde{\boldsymbol{s}}(1), \ldots, \tilde{\boldsymbol{s}}(N)$ have the same sparsity pattern, i.e., the non-zero entries of these vectors have the same indices, the matrix $\tilde{\boldsymbol{S}}$ exhibits a row-sparse structure. In other words, the elements in a row of the matrix $\tilde{\boldsymbol{S}}$ are either jointly zero or mostly non-zero. Fig. 2.2 illustrates the row-sparsity structure of (2.27) regardless of the additive white Gaussian noise. The row-sparse structure in the matrix $\tilde{\boldsymbol{S}}$ can be exploited for DOA estimation by using the mixed norm formulation [Tro06, Kow09, YL06, HM10]

$$\min_{\tilde{\boldsymbol{S}}} \quad \|\boldsymbol{X} - \tilde{\boldsymbol{A}}\,\tilde{\boldsymbol{S}}\|_{\mathrm{F}}^2 + \tau \|\tilde{\boldsymbol{S}}\|_{p,q}, \tag{2.28}$$

where $\tau > 0$ is a regularization parameter, $\|\cdot\|_{\mathrm{F}}$ is the Frobenius matrix norm, and

$$\|\tilde{\boldsymbol{S}}\|_{p,q} = \Big(\sum_{i=1}^{G} \big(\sum_{j=1}^{N} [\tilde{\boldsymbol{S}}]_{i,j}^{p} \big)^{q/p} \Big)^{1/q} \tag{2.29}$$

is the $\ell_{p,q}$ mixed norm. The elements of each row of the matrix $\tilde{\boldsymbol{S}}$ are coupled through the ℓ_p norm, whereas the ℓ_q norm is a sparsity inducing norm. Ideally, the ℓ_0 pseudo-norm can be used as an ideal sparsity inducing norm, however, this resulting problem is NP-hard. Therefore, to obtain a computationally tractable problem, a convex relaxation of $\ell_{p,1}$ mixed norm is considered in practice. A common choice of mixed norm is the $\ell_{2,1}$ norm [YL06]. Note that using the SSR approach of (2.28), the DOA estimation problem is reduced to finding the peaks of the directional spectrum

$$\mathcal{S}_{\ell_{p,q}}(\tilde{\theta}_i) = \big(\sum_{j=1}^{N} [\tilde{\boldsymbol{S}}]_{i,j}^{p} \big)^{1/p}. \tag{2.30}$$

[1] If the sources are not on the grid then (2.25) in not accurate and the performance of the SSR approach is deteriorated [CSPC11].

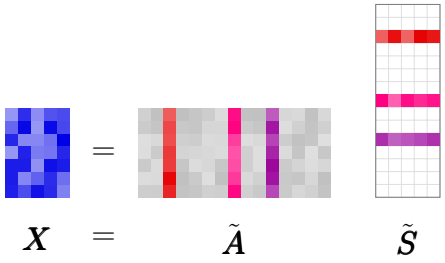

Fig. 2.2. Sparse representation of (2.27) (neglecting the additive noise) for $N = 5$ snapshots, $M = 7$ sensors, $L = 3$ source signals and $G = 15$ grid points.

2.2.3.2 Covariance-Based DOA Estimation

For coherent processing using *fully calibrated* arrays, the covariance-based SSR approaches for DOA estimation in the deterministic and stochastic source models are introduced in [SPP16] and [SBL11], respectively. Since a stochastic source model is assumed in this dissertation, the approach of [SBL11], referred to as SParse Iterative Covariance-based approach (SPICE), is reviewed.

In the SPICE method, uncorrelated sources are assumed, and a sparse representation, which matches the sample covariance matrix (2.15), is sought as follows. Considering the sparse representation of the measurements in (2.26), the measurement covariance matrix (2.11) can be written as

$$\tilde{\boldsymbol{R}} = \tilde{\boldsymbol{A}}\tilde{\boldsymbol{P}}\tilde{\boldsymbol{A}}^H + \sigma^2 \boldsymbol{I}_M, \tag{2.31}$$

where the overcomplete dictionary $\tilde{\boldsymbol{A}}$ is defined in (2.25). The diagonal entries of the sparse diagonal matrix

$$\tilde{\boldsymbol{P}} = \operatorname{diag}\big(\tilde{p}_1, \ldots, \tilde{p}_G\big) \tag{2.32}$$

correspond to power of the sources whose directions are the entries of the vector $\tilde{\boldsymbol{\theta}}$. Fig. 2.3 illustrates the sparsity structure of (2.31) regardless of the noise effects, i.e., neglecting the term $\sigma^2 \boldsymbol{I}_M$. The SPICE optimization problem [SBL11, Equation (20)] is written as

$$\begin{aligned} \min_{\tilde{\boldsymbol{P}},\sigma} \quad & \operatorname{tr}\big(\tilde{\boldsymbol{R}}^{-1}\hat{\boldsymbol{R}}\big) && \text{(2.33a)} \\ \text{subject to} \quad & \tilde{p}_i \geq 0, i = 1, \ldots, G && \text{(2.33b)} \\ & \sum_{i=1}^{G} \omega_i \tilde{p}_g + \bar{\omega}\sigma^2 = 1 && \text{(2.33c)} \end{aligned}$$

where

$$\omega_g = \boldsymbol{a}^H(\tilde{\theta}_g)\hat{\boldsymbol{R}}^{-1}\boldsymbol{a}(\tilde{\theta}_g) \tag{2.34}$$

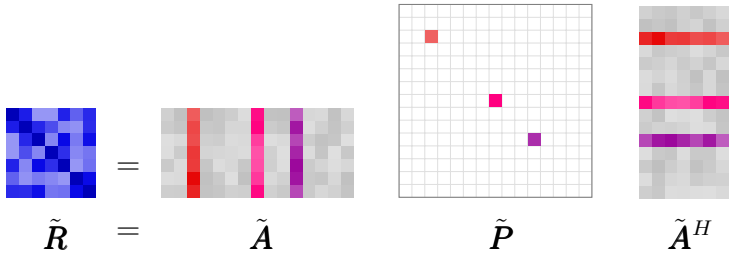

Fig. 2.3. Sparse representation of (2.31) (neglecting the additive noise) for $M = 7$ sensors, $L = 3$ source signals, and $G = 15$ grid points.

and

$$\bar{\omega} = \frac{1}{M}\mathrm{tr}\big(\hat{\boldsymbol{R}}^{-1}\big) \tag{2.35}$$

are the weights corresponding to the powers of the sources and the noise variance, respectively. In [SBL11], it has been pointed out that the constraint (2.33c) is a weighted ℓ_1 norm and thus is "expected" to induce sparsity. Note that in contrast to other ℓ_1 norm-based DOA estimation approaches, the SPICE approach does not require the configuration of a sparsity regularization parameter. Problem (2.33) is semi-definite [SBL11] and thus can be solved using conventional convex optimization frameworks, e.g., cvx [CR12]. Although the SPICE algorithm is derive in [SBL11] under the assumption of uncorrelated sources, it has been shown in [SBL11] that the SPICE algorithm is robust against this assumption.

2.3 Subspace-Based DOA Estimation for Partly Calibrated Arrays

The problem of DOA estimation using partly-calibrated arrays with arbitrary sensor positions has been considered in [PPG11]. The algorithm of [PPG11] is search-based, thus, requires high computational cost. Search-free DOA estimation algorithms are applicable for specific array geometries, e.g., ESPRIT [RK89] and Root-RARE [PGW02]. In this section, the so-called shift-invariant array structure is reviewed. Moreover, the ESPRIT algorithm [RK89] which exploits the shift-invariant array structure to achieve search-free DOA estimation is revised.

2.3.1 Shift-Invariant Arrays

In a shift-invariant array, the sensors of the array can be partitioned into two identical groups which are separated by a common displacement d. These two sensor groups are

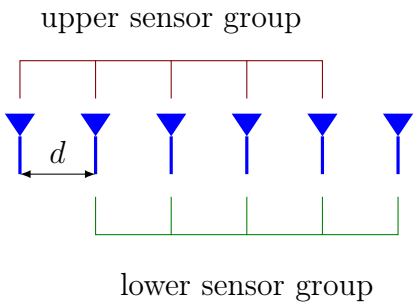

Fig. 2.4. Example of a shift-invariant array of $M = 6$ sensors. The sensors are arranged in upper and lower groups each of 5 sensors. In this example, the upper and lower groups are overlapping.

referred to as the upper and lower sensor groups. For example, Fig. 2.4 demonstrate a ULA of $M = 6$ sensors where the distance between two successive sensors is denoted by d. The sensors in this array configuration are arranged in upper and lower groups where each group consists of 5 sensors. The four sensors in the center of the array belongs to both groups, i.e., the groups are overlapping. In Fig. 2.4, by shifting the sensors of the upper group by d the position of the sensors of the upper group will coincide with that of the lower group, hence the name shift-invariant array.

For partly-calibrated array, if all subarrays admit the shift-invariant structure with the same displacement d, the array is referred to as shift-invariant. An example of such an array is demonstrated in Fig. 2.5, where the partly-calibrated array consists of $K = 3$ subarrays. The upper and lower sensor groups are shown for three subarray. Note that where the upper and lower sensor groups of the first subarray are overlapping, these groups are completely separated in the second and third subarrays. Nevertheless, the whole array is shift-invariant since the displacement between the upper and lower sensor groups is identical at all subarrays. The upper and lower sensor groups of the whole array are the union of the upper and lower sensor groups, respectively, of the individual subarrays.

The subarray response matrices corresponding to the upper and lower sensor groups of the kth subarray are denoted as

$$\overline{\boldsymbol{A}}_k = \overline{\boldsymbol{J}}_k \boldsymbol{A}_k \tag{2.36}$$

and

$$\underline{\boldsymbol{A}}_k = \underline{\boldsymbol{J}}_k \boldsymbol{A}_k, \tag{2.37}$$

respectively, where $\overline{\boldsymbol{J}}_k$ and $\underline{\boldsymbol{J}}_k$ are the upper and lower selection matrices. The structure of $\overline{\boldsymbol{J}}_k$ and $\underline{\boldsymbol{J}}_k$ depends on the shift-invariant structure of the kth subarray. For example,

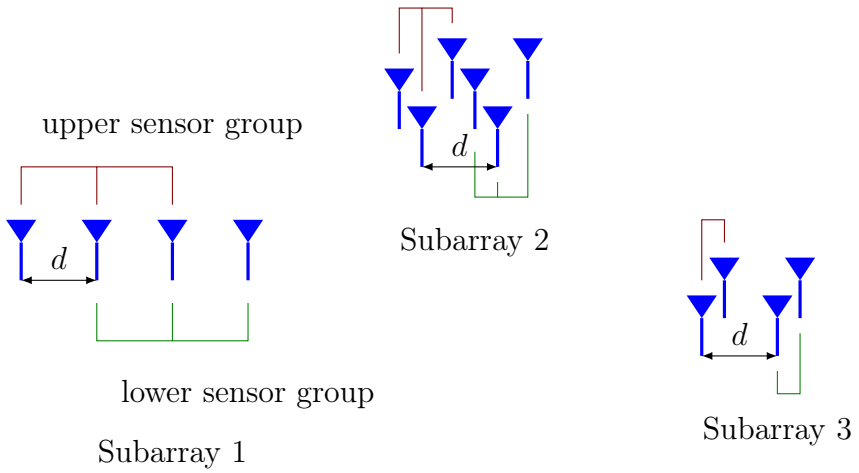

Fig. 2.5. Example of a shift-invariant partly-calibrated array of $M = 14$ sensors and $K = 3$ subarrays. The subarrays are partitioned into upper and lower groups with identical separation d.

the upper and lower selection matrices of the first subarray in Fig. 2.5 are $\overline{\boldsymbol{J}}_1 = [\boldsymbol{I}_3, \boldsymbol{0}_3] \in \mathbb{R}^{3\times 4}$ and $\underline{\boldsymbol{J}}_1 = [\boldsymbol{0}_3, \boldsymbol{I}_3] \in \mathbb{R}^{3\times 4}$, respectively, where $\boldsymbol{I}_3$ is the 3×3 identity matrix and $\boldsymbol{0}_3$ is the all zeros vector of length 3. Whereas for the second subarray, these matrices are $\overline{\boldsymbol{J}}_2 = [\boldsymbol{I}_3, \boldsymbol{0}_3\boldsymbol{0}_3^T] \in \mathbb{R}^{3\times 6}$ and $\underline{\boldsymbol{J}}_2 = [\boldsymbol{0}_3\boldsymbol{0}_3^T, \boldsymbol{I}_3] \in \mathbb{R}^{3\times 6}$, respectively. Let

$$\overline{\boldsymbol{J}} = \text{blkdiag}\big(\overline{\boldsymbol{J}}_1, \ldots, \overline{\boldsymbol{J}}_K\big) \tag{2.38}$$

and

$$\underline{\boldsymbol{J}} = \text{blkdiag}\big(\underline{\boldsymbol{J}}_1, \ldots, \underline{\boldsymbol{J}}_K\big), \tag{2.39}$$

where blkdiag$(\cdot)$ constructs block diagonal matrix from its operands, denote the upper and lower selection matrices of the whole array, respectively. Then, it follows from (2.36) and (2.37) that the array response matrices corresponding to the upper and lower sensor groups are

$$\overline{\boldsymbol{A}} = \overline{\boldsymbol{J}}\boldsymbol{A} \tag{2.40}$$

and

$$\underline{\boldsymbol{A}} = \underline{\boldsymbol{J}}\boldsymbol{A}, \tag{2.41}$$

where $\overline{\boldsymbol{A}} = [\overline{\boldsymbol{A}}_1^T, \ldots, \overline{\boldsymbol{A}}_K^T]^T$, $\underline{\boldsymbol{A}} = [\underline{\boldsymbol{A}}_1^T, \ldots, \underline{\boldsymbol{A}}_K^T]^T$, and the overall array response $\boldsymbol{A}$ is defined in (2.10). It can be easily seen that the upper and lower response matrices are related by [RK89]

$$\overline{\boldsymbol{A}}\boldsymbol{\Omega} = \underline{\boldsymbol{A}}, \tag{2.42}$$

where the $L \times L$ diagonal matrix $\boldsymbol{\Omega}$ is defined as

$$\boldsymbol{\Omega} = \text{diag}\Big(\exp\big(\jmath\frac{2\pi}{\lambda_c} d \sin(\theta_1)\big), \ldots, \exp\big(\jmath\frac{2\pi}{\lambda_c} d \sin(\theta_L)\big)\Big). \tag{2.43}$$

The property (2.42) is the basis of the ESPRIT algorithm presented in the next subsection.

2.3.2 The ESPRIT Algorithm

The ESPRIT algorithm [RK89] is a search-free DOA estimation algorithm which is applicable for *shift-invariant partly calibrated* arrays. In the ESPRIT algorithm, the upper and lower parts of the signal eigenvectors matrix $\boldsymbol{U}_{\mathrm{s}}$, defined in (2.14), are denoted as

$$\overline{\boldsymbol{U}}_{\mathrm{s}} = \overline{\boldsymbol{J}}\boldsymbol{U}_{\mathrm{s}} \tag{2.44}$$

and

$$\underline{\boldsymbol{U}}_{\mathrm{s}} = \underline{\boldsymbol{J}}\boldsymbol{U}_{\mathrm{s}}, \tag{2.45}$$

respectively. These two matrices correspond to the upper and lower sensor groups of the shift-invariant array. The matrices $\overline{\boldsymbol{U}}_{\mathrm{s}}$ and $\underline{\boldsymbol{U}}_{\mathrm{s}}$ span the same subspace as the matrices $\overline{\boldsymbol{A}}$ and $\underline{\boldsymbol{A}}$, respectively. Thus, the following equation holds [RK89]

$$\begin{bmatrix} \overline{\boldsymbol{U}}_{\mathrm{s}} \\ \underline{\boldsymbol{U}}_{\mathrm{s}} \end{bmatrix} = \begin{bmatrix} \overline{\boldsymbol{A}} \\ \underline{\boldsymbol{A}}\boldsymbol{\Omega} \end{bmatrix} \boldsymbol{\mathcal{H}}, \tag{2.46}$$

where $\boldsymbol{\mathcal{H}}$ is a nonsingular $L \times L$ matrix. By simple calculation, it is proved in [RK89] that the matrices $\boldsymbol{\Omega}$ and

$$\boldsymbol{\Psi} = \left(\overline{\boldsymbol{U}}_{\mathrm{s}}^{H}\overline{\boldsymbol{U}}_{\mathrm{s}}\right)^{-1}\overline{\boldsymbol{U}}_{\mathrm{s}}^{H}\underline{\boldsymbol{U}}_{\mathrm{s}} \tag{2.47}$$

are similar matrices in the sense that they possess the same eigenvalues. Thus, the diagonal entries of the diagonal matrix $\boldsymbol{\Omega}$ can be computed from the eigenvalues of the matrix $\boldsymbol{\Psi}$. Consequently, the DOAs are computed using

$$\theta_l = \sin^{-1}(\arg(\psi_l)\lambda_c/(2\pi d)), \tag{2.48}$$

where ψ_l for $l = 1, \ldots, L$ are the eigenvalues of the matrix $\boldsymbol{\Psi}$. In practice, a sample estimate of ψ_l, for $l = 1, \ldots, L$, can be computed based on the sample estimate of the signal eigenvectors matrix $\hat{\boldsymbol{U}}_{\mathrm{s}}$ as follows. Firstly, the estimates of the upper and lower signal eigenvector matrices are computed based on (2.44) and (2.45), denote these estimates as $\hat{\overline{\boldsymbol{U}}}_{\mathrm{s}}$ and $\hat{\underline{\boldsymbol{U}}}_{\mathrm{s}}$, respectively. Secondly, the matrix $\hat{\boldsymbol{\Psi}}$ which is the sample estimate of $\boldsymbol{\Psi}$ is computed from (2.47). Finally, the eigenvalues of $\hat{\boldsymbol{\Psi}}$, denoted as $\hat{\psi}_1, \ldots, \hat{\psi}_L$, are computed and used in (2.48) to estimate the DOAs. The estimated DOAs are denoted by $\hat{\boldsymbol{\theta}} = [\hat{\theta}_1, \ldots, \hat{\theta}_L]^T$. In the following section, the problem of estimating the matrix $\boldsymbol{U}_{\mathrm{s}}$ in a decentralized fashion is considered.

2.4 The Decentralized Eigendecomposition

The sample estimate of the signal eigenvector matrix $\hat{\boldsymbol{U}}_{\mathrm{s}}$ in (2.16) is required for many DOA estimation algorithms as shown for MUSIC, Root-MUSIC, and ESPRIT algorithms. Computing all the eigenvalues and eigenvectors of a matrix in a centralized fashion is well established, see, e.g., [ABB+99]. Iterative algorithms for computing few eigenvectors, thus, saving computation time, are also available, e.g., the power method (PM) and the Lanczos method [GVL13]. In this section, decentralized eigendecomposition using the PM, which has been introduced in [SPK08], is reviewed.

2.4.1 The Conventional Power Method

The conventional PM is an iterative algorithm, which can be used to compute the principal eigenvector of the sample covariance matrix $\hat{\boldsymbol{R}}$. In the qth PM iteration the following update is used:

$$\hat{\boldsymbol{u}}_1^{(q)} = \hat{\boldsymbol{R}}\,\hat{\boldsymbol{u}}_1^{(q-1)}, \tag{2.49}$$

where $\hat{\boldsymbol{u}}_1^{(q-1)}$ is the sample estimate of the principal eigenvector of the matrix $\hat{\boldsymbol{R}}$ at the qth PM iteration. The vector $\hat{\boldsymbol{u}}_l^{(0)}$ is a random initial value. After a sufficient number of iterations Q, the vector $\hat{\boldsymbol{u}}_1^{(Q)}$ is normalized[2] yielding $\hat{\boldsymbol{u}}_1$, i.e.,

$$\hat{\boldsymbol{u}}_1 = \hat{\boldsymbol{u}}_1^{(Q)} / \|\hat{\boldsymbol{u}}_1^{(Q)}\|. \tag{2.50}$$

Equation (2.50) requires $Q \to \infty$, i.e., Q approaches infinity[3]. In this dissertation, Q is considered to be large such that (2.50) holds, i.e., it is assumed that after Q PM iterations, the normalized vector $\hat{\boldsymbol{u}}_1^{(Q)}/\|\hat{\boldsymbol{u}}_1^{(Q)}\|$ converges to the principal eigenvector of the matrix $\hat{\boldsymbol{R}}$ denoted by $\hat{\boldsymbol{u}}_1$. Assume that $(l-1)$ principal eigenvectors of the matrix $\hat{\boldsymbol{R}}$ have been computed using the PM, then, the lth eigenvector is computed using the iteration

$$\hat{\boldsymbol{u}}_l^{(q)} = (\boldsymbol{I}_M - \hat{\boldsymbol{U}}_{l-1}\hat{\boldsymbol{U}}_{l-1}^H)\hat{\boldsymbol{R}}\,\hat{\boldsymbol{u}}_l^{(q-1)}, \tag{2.51}$$

where $\hat{\boldsymbol{u}}_l^{(q)}$ denotes the lth eigenvector of $\hat{\boldsymbol{R}}$ at the qth PM iteration, $\boldsymbol{I}_M$ is the $M \times M$ identity matrix and $\hat{\boldsymbol{U}}_{l-1} = [\hat{\boldsymbol{u}}_1, \ldots, \hat{\boldsymbol{u}}_{l-1}]$ is the concatenation of the $l-1$ previously computed eigenvectors of $\hat{\boldsymbol{R}}$ using the PM. Note that the matrix $(\hat{\boldsymbol{U}}_{l-1}\hat{\boldsymbol{U}}_{l-1}^H)$ is an orthogonal projection onto the computed eigenvectors, thus, the principal eigenvalue of the matrix $(\boldsymbol{I}_M - \hat{\boldsymbol{U}}_{l-1}\hat{\boldsymbol{U}}_{l-1}^H)\hat{\boldsymbol{R}}$ is the lth eigenvalue of the matrix $\hat{\boldsymbol{R}}$. After a sufficiently

[2] Note that if Q is large then the normalization step should also be carried out periodically for $q < Q$ to avoid numerical issues in the vector $\hat{\boldsymbol{u}}_l^{(q)}$.

[3] The convergence rate of the PM depends on the ratio of the first and second largest eigenvalues of the matrix $\hat{\boldsymbol{R}}$ [GVL13].

large number of PM iterations Q, the vector $\hat{\boldsymbol{u}}_l^{(Q)}$ is normalized as in (2.50) yielding the lth eigenvector of the matrix $\hat{\boldsymbol{R}}$, i.e.,

$$\hat{\boldsymbol{u}}_l = \hat{\boldsymbol{u}}_l^{(Q)} / \|\hat{\boldsymbol{u}}_l^{(Q)}\|. \tag{2.52}$$

Algorithm 1 summaries the steps of the conventional PM algorithm. The eigenvalues of the matrix $\hat{\boldsymbol{R}}$ can be computed from the eigenvectors as

$$\hat{\boldsymbol{\Lambda}} = \hat{\boldsymbol{U}}^H \hat{\boldsymbol{R}} \hat{\boldsymbol{U}}, \tag{2.53}$$

where the diagonal matrix $\hat{\boldsymbol{\Lambda}}$, defined in (2.16), represents the eigenvalues of $\hat{\boldsymbol{R}}$. In the following subsection, the AC protocol which is used for the decentralized implementation of the PM is introduced.

Algorithm 1 The Conventional Power Method (PM)

Step 0: Set $\hat{\boldsymbol{U}}_0$ to zero.
for $l = 1, 2, \ldots, M$ **do**
  Step 0: $\hat{\boldsymbol{u}}_l^{(0)} \leftarrow M \times 1$ random vector
  for $q = 1, 2, \ldots, Q$ **do**
    Step 1: Compute $\hat{\boldsymbol{u}}_l^{(q)}$ as in (2.51).
  end for
  Step 2: Normalize $\hat{\boldsymbol{u}}_l^{(Q)}$ as in (2.52) and append the normalized eigenvector to $\hat{\boldsymbol{U}}_{l-1}$.
end for

2.4.2 The Averaging Consensus Protocol

The AC protocol is used to iteratively compute the average of scalar numbers, which are distributed among K network nodes using only local communications between neighboring nodes. The term network node is used in wireless sensor networks, however, in this dissertation, since DOA estimation is considered, these nodes are subarrays. Fig. 2.6 demonstrates the topology of a network composed of $K = 6$ nodes. The red lines indicate communication links between the nodes, i.e., nodes (or subarrays) only communicate with their adjacent nodes. Let $\chi_1, \ldots, \chi_K$ denote K scalars, which are locally available at K distinct nodes in the network, where the kth node stores only the kth scalar. These scalar are either the sensor measurements or quantities computed from the measurements. The conventional average of these scalars is denoted as $\chi = \frac{1}{K} \sum_{k=1}^{K} \chi_k$. In the AC protocol [DeG74, OSM04, XB04, OSFM07, XBK07], χ is computed iteratively, where at the pth AC iteration, the kth node sends its current local estimate of the average $\chi_k^{(p-1)}$ to its neighboring nodes, denoted as the set $\mathcal{N}_k$,

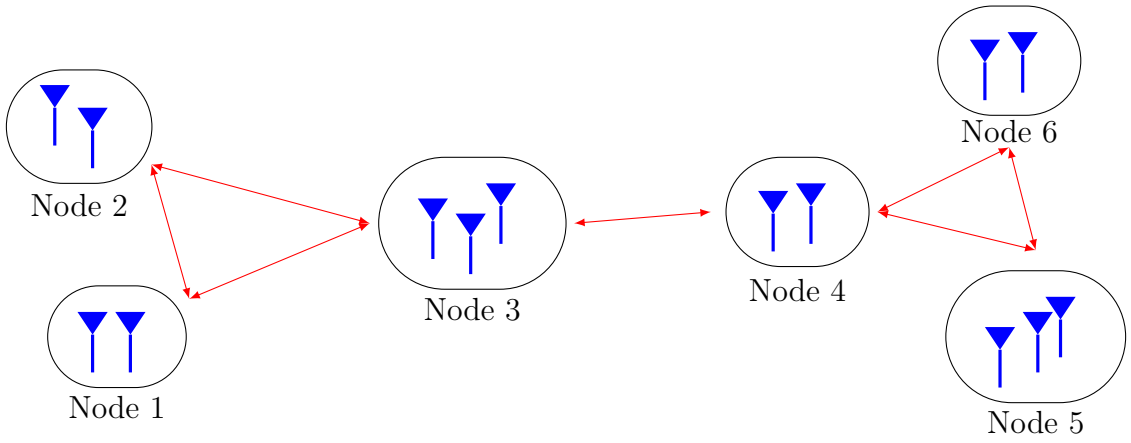

Fig. 2.6. The topology of a wireless sensor network of $K = 6$ nodes. Each node communicates with its neighboring nodes. The red lines indicate the communication links.

and receives the corresponding average estimates of the respective neighboring nodes. Then, the kth node updates its local estimate of the average as follows:

$$\chi_k^{(p)} = w_{k,k}\, \chi_k^{(p-1)} + \sum_{i \in \mathcal{N}_k} w_{i,k}\, \chi_i^{(p-1)}, \tag{2.54}$$

where $w_{i,k}$ is the weighting factor associated with the communication link between node i and node k, which satisfies $w_{i,k} = 0$ when $i \notin \mathcal{N}_k$ [XB04]. The AC iteration in (2.54) is initialized with $\chi_k^{(0)} = \chi_k$, for $k = 1, \ldots, K$. For more details, see [XB04].

Let $\boldsymbol{\chi}^{(p)} = [\chi_1^{(p)}, \ldots, \chi_K^{(p)}]^T$, then the update iteration in (2.54) can be expressed as

$$\boldsymbol{\chi}^{(p)} = \boldsymbol{W}\boldsymbol{\chi}^{(p-1)} = \boldsymbol{W}^2\boldsymbol{\chi}^{(p-2)} = \cdots = \boldsymbol{W}^p\boldsymbol{\chi}^{(0)}, \tag{2.55}$$

where the entries of the $K \times K$ AC weighting matrix $\boldsymbol{W}$ are $[\boldsymbol{W}]_{i,j} = w_{i,j}$ for $i, j = 1, \ldots, K$. Iteration (2.55) converges asymptotically (for $p \to \infty$) to the vector of averages $\chi \mathbf{1}_K$ if and only if

$$\lim_{p \to \infty} \boldsymbol{W}^p = \frac{\mathbf{1}_K \mathbf{1}_K^T}{K}, \tag{2.56}$$

where $\mathbf{1}_K \mathbf{1}_K^T$ is the all one matrix of size $K \times K$ and $\lim_{p\to\infty} \boldsymbol{W}^p$ denote the limit of $\boldsymbol{W}^p$ when p approaches infinity. Let the eigendecomposition of the matrix $\boldsymbol{W}$ be

$$\boldsymbol{W}\,[\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_K] = [\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_K]\,\mathrm{diag}\big(\alpha_1, \ldots, \alpha_K\big), \tag{2.57}$$

where $\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_K$ are the eigenvectors of the matrix $\boldsymbol{W}$ corresponding to the ordered eigenvalues $\alpha_1 > \alpha_2 \geq \cdots \geq \alpha_K$. According to [XB04], the matrix $\boldsymbol{W}$ which satisfies the asymptotic convergence condition (2.56) possesses the following properties:

P1: The principal eigenvalue of the matrix $\boldsymbol{W}$ is unique (single multiplicity) and equals to one, i.e., $\alpha_1 = 1$. The corresponding normalized principal eigenvector of the matrix $\boldsymbol{W}$ is given by $\boldsymbol{\beta}_1 = \frac{1}{\sqrt{K}}\mathbf{1}_K$.

P2: The remaining eigenvalues of $\boldsymbol{W}$ are strictly less than α_1 in magnitude.

The decentralized value of the average χ computed at the kth node using p AC iterations is expressed as

$$\check{\chi}_{[k]} = [\boldsymbol{W}^p \boldsymbol{\chi}^{(0)}]_k, \tag{2.58}$$

where $[\boldsymbol{W}^p \boldsymbol{\chi}^{(0)}]_k$ denotes the kth entry of the vector $\boldsymbol{W}^p \boldsymbol{\chi}^{(0)}$. The notation $\check{\chi}$ in (2.58) indicates that the average is computed using the AC protocol. The sub-index $[k]$ in (2.58) means that all the nodes in the network possess local instance of the average. Note that if the value of the average is only available at the kth node, then, it would be denoted as $\check{\chi}_k$, i.e., without the index brackets surrounding the node index k.

2.4.3 The Decentralized Power Method

The DPM [SPK08] achieves a decentralized eigendecomposition of the sample covariance matrix without explicitly computing the sample covariance matrix $\hat{\boldsymbol{R}}$. In the DPM, the computations in (2.51) and (2.52) are performed in a fully decentralized fashion using the AC protocol. In this aspect in the DPM, the lth eigenvector is partitioned at the qth PM iteration as

$$\check{\boldsymbol{u}}_l^{(q)} = [\check{\boldsymbol{u}}_{l,1}^{(q)T}, \ldots, \check{\boldsymbol{u}}_{l,K}^{(q)T}]^T, \tag{2.59}$$

where the kth node stores and updates only the kth part, denoted as $\check{\boldsymbol{u}}_{l,k}^{(q)} \in \mathbb{C}^{M_k \times 1}$, of the vector $\check{\boldsymbol{u}}_l^{(q)}$. The notation $\check{\boldsymbol{u}}_{l,k}$ (and not $\hat{\boldsymbol{u}}_{l,k}$) is used in (2.59) to indicate that in the DPM, the vector $\check{\boldsymbol{u}}_{l,k}$ is computed using the AC protocol and stored only at the kth node, refer to (2.58) for details about the notation. Towards explaining the DPM, we perform iteration (2.51) in two steps. In the first step, the intermediate vector

$$\check{\boldsymbol{u}}_l'^{(q)} = \hat{\boldsymbol{R}}\, \check{\boldsymbol{u}}_l^{(q-1)} \tag{2.60}$$

is calculated. In the second step, the vector $\check{\boldsymbol{u}}_l^{(q)}$ is updated as

$$\check{\boldsymbol{u}}_l^{(q)} = \check{\boldsymbol{u}}_l'^{(q)} - \check{\boldsymbol{U}}_{l-1} \check{\boldsymbol{U}}_{l-1}^H \check{\boldsymbol{u}}_l'^{(q)}, \tag{2.61}$$

where $\check{\boldsymbol{U}}_{l-1} = [\check{\boldsymbol{u}}_1, \ldots, \check{\boldsymbol{u}}_{l-1}]$ is the concatenation of the $l-1$ vectors already computed using the DPM. In the following, the fully decentralized computation of both steps

(2.60) and (2.61), and the normalization Step (2.52) are reviewed, for details refer to [SPK08].

Substituting (2.15) into (2.60) yields

$$\check{\boldsymbol{u}}_l'^{(q)} = \frac{1}{N}\sum_{t=1}^{N} \boldsymbol{x}(t)\check{x}_{t,l}^{(q)}, \tag{2.62}$$

where $\check{x}_{t,l}^{(q)} = \boldsymbol{x}^H(t)\check{\boldsymbol{u}}_l^{(q-1)}$. Note that

$$\check{x}_{t,l}^{(q)} = K\frac{1}{K}\sum_{k=1}^{K} \boldsymbol{x}_k^H(t)\check{\boldsymbol{u}}_{l,k}^{(q-1)}, \tag{2.63}$$

where $\boldsymbol{x}_k^H(t)\check{\boldsymbol{u}}_{l,k}$ is computed and stored locally at the kth node. In analogy to (2.58), the estimate of $\check{x}_{t,l}^{(q)}$ at the kth node computed using the AC protocol is

$$\check{x}_{t,l,[k]}^{(q)} = K\left[\boldsymbol{W}^P \begin{bmatrix} \boldsymbol{x}_1^H(t)\check{\boldsymbol{u}}_{l,1}^{(q-1)} \\ \vdots \\ \boldsymbol{x}_K^H(t)\check{\boldsymbol{u}}_{l,K}^{(q-1)}] \end{bmatrix}\right]_k, \tag{2.64}$$

where P is the number of AC iterations used in this AC protocol [SPK08]. Using N parallel instances of the AC protocol, the kth node will locally maintain the scalars $\{\check{x}_{t,l,[k]}^{(q)}\}_{t=1}^{N}$. Thus, each node k can locally compute one part of the vector $\check{\boldsymbol{u}}_l'^{(q)}$ as

$$\check{\boldsymbol{u}}_{l,k}'^{(q)} = \frac{1}{N}\sum_{t=1}^{N} \boldsymbol{x}_k(t)\check{x}_{t,l,[k]}^{(q)}, \tag{2.65}$$

that in turn perform the first step of the DPM iteration described in (2.60).

Note that, in the second step of the DPM iteration, only the second term of (2.61) has to be computed in a decentralized fashion [SPK08]. This term can be written as

$$\check{\boldsymbol{U}}_{l-1}\check{\boldsymbol{U}}_{l-1}^H\check{\boldsymbol{u}}_l'^{(q)} = \sum_{i=1}^{l-1} \check{\boldsymbol{u}}_i^H\check{u}_{i,l}'^{(q)}, \tag{2.66}$$

where $\check{u}_{i,l}'^{(q)} = \check{\boldsymbol{u}}_i^H\check{\boldsymbol{u}}_l'^{(q)}$. In analogy to (2.62), each node can locally compute its corresponding part of $\check{\boldsymbol{U}}_{l-1}\check{\boldsymbol{U}}_{l-1}^H\check{\boldsymbol{u}}_l'^{(q)}$ once the scalars $\{\check{u}_{i,l}'^{(q)}\}_{i=1}^{l-1}$ are available at every node. This can be achieved using $l-1$ parallel instances of the AC protocol as

$$\check{u}_{i,l,[k]}'^{(q)} = K\left[\boldsymbol{W}^{P_1} \begin{bmatrix} \check{\boldsymbol{u}}_{i,1}^H\,\check{\boldsymbol{u}}_{l,1}'^{(q)} \\ \vdots \\ \check{\boldsymbol{u}}_{i,K}^H\,\check{\boldsymbol{u}}_{l,K}'^{(q)} \end{bmatrix}\right]_k, \tag{2.67}$$

where $\check{u}_{i,l,[k]}^{\prime(q)}$ is the ith scalar computed at the kth node and P_1 is the number of AC iterations used in these $l-1$ AC protocol instances. Thus achieving the second step of the DPM iteration.

After a sufficiently large number of PM iterations Q, the vector $\check{\boldsymbol{u}}_l^{(Q)}$ is normalized as in (2.52). This normalization can be carried out locally once the norm $\|\check{\boldsymbol{u}}_l^{(Q)}\|$ is available at each node which is achieved using the AC protocol as

$$\|\check{\boldsymbol{u}}_l^{(Q)}\|_{[k]}^2 = K \left[\boldsymbol{W}^{P_2} \begin{bmatrix} \check{\boldsymbol{u}}_{l,1}^{(Q)H} \check{\boldsymbol{u}}_{l,1}^{(Q)} \\ \vdots \\ \check{\boldsymbol{u}}_{l,K}^{(Q)H} \check{\boldsymbol{u}}_{l,K}^{(Q)} \end{bmatrix} \right]_k , \tag{2.68}$$

where P_2 is the number of iterations used in the AC protocol instance. Thus, using the DPM, the eigenvectors of the sample covariance matrix can be calculated without PC. Algorithm 2 summaries the steps of the DPM and shows where the AC protocol

Algorithm 2 The DPM

Step 0: Set $\hat{\boldsymbol{U}}_0$ to empty matrix.
for $l = 1, 2, \ldots, M$ **do**
  Step 0: $\hat{\boldsymbol{u}}_{l,k}^{(0)} \leftarrow M_k \times 1$ random vector for $k = 1, \ldots, K$
  for $q = 1, 2, \ldots, Q$ **do**
    Step 1: Compute $\check{\boldsymbol{u}}_l^{\prime(q)}$ as described in (2.62)-(2.65) using N parallel instances of the AC protocol each with P iterations.
    Step 2: Compute $\check{\boldsymbol{u}}_l^{(q)}$ as described in (2.61), (2.66), and (2.67) using $l-1$ parallel instances of the AC protocol each with P_1 iterations.
  end for
  Step 3: Normalize $\check{\boldsymbol{u}}_l^{(Q)}$ as in (2.68) using one instance of the AC protocol with P_2 iterations and append the normalized eigenvector to $\check{\boldsymbol{U}}_{l-1}$.
end for

is used.

The eigenvalues of $\hat{\boldsymbol{R}}$ can be computed from the estimated eigenvector matrix $\check{\boldsymbol{U}}$ as in the conventional PM which is described in (2.53). Consider the decentralized computation of the lth eigenvalue which can be written as

$$\check{\lambda}_l = \check{\boldsymbol{u}}_l^H \hat{\boldsymbol{R}} \check{\boldsymbol{u}}_l. \tag{2.69}$$

This computation can be decomposed into two steps. In the first step, the vector

$$\check{\boldsymbol{u}}_l^{\prime} = \hat{\boldsymbol{R}} \check{\boldsymbol{u}}_l \tag{2.70}$$

is computed using N parallel instances of the AC protocol with P iterations each as described in (2.62)-(2.65). In the second step, the scalar product

$$\check{\lambda}_l = \check{\boldsymbol{u}}_l^H \check{\boldsymbol{u}}_l^{\prime} \tag{2.71}$$

is computed using one instance of the AC protocol with P_3 iterations as described in (2.68).

2.4.4 Communication Cost Analysis of the DPM

In the DPM, communication between the nodes is required to compute the scalars in steps 1, 2, and 3 of Algorithm 2. From a signaling perspective, the first and most expensive computation is that of Step 1 where N parallel instances of the AC protocol are required, i.e., NP AC iterations, are carried out in Step 1. The second most expensive computation lies in Step 2 of Algorithm 2, which requires $l-1$ AC protocol instances, i.e., $(l-1)P_1$ AC iterations. The third and least expensive computation is the normalization of the eigenvectors which requires only one AC protocol instance with P_2. Consequently, computing each eigenvector using the DPM, requires a communication cost per node of order QNP complex numbers.

Computing the eigenvalues in the DPM is performed in two steps, namely, (2.70) and (2.71) which cost N and one AC protocol instances, respectively. The decentralized computation of (2.70) is the most expensive step with cost of order P_3N complex numbers per node.

Chapter 3
Coherent Decentralized DOA Estimation

In this chapter, decentralized DOA estimation using the decentralized ESPRIT (DESPRIT) and decentralized Root-MUSIC algorithms is introduced. These algorithms are based on the AC protocol and the DPM. The performance analysis of the DPM for the eigendecomposition of the sample covariance matrix based on the AC protocol is presented. Analytical expressions for the asymptotic behavior of the second order statistics of the eigenvectors and eigenvalues obtained from the DPM are derived. Based on the performance analysis of the DPM, analytical expressions for the asymptotic behavior of the MSE of DOA estimators using the DESPRIT and decentralized Root-MUSIC algorithms are obtained. Moreover, the problem of source number detection, which is essential for DOA estimation, is considered. A decentralized source enumeration algorithm is introduced based on the ED. In this decentralized algorithm, the decision that a source is present is taken before computing the eigenvalue and eigenvector corresponding to this source. Thus, reducing the communication and computational cost required for implementing the algorithm in a decentralized fashion. The performance of the proposed algorithms is demonstrated by simulations. This chapter is based on the publications in [SPZ13, SPZ16c, SPZ16b, SPZ15b, SPZ16a, SVPZ16].

3.1 The Decentralized ESPRIT Algorithm

In the DESPRIT algorithm [SPZ13], fully decentralized DOA estimation based on the ESPRIT algorithm [RK89] and the DPM [SPK08] is performed. Thus the DESPRIT algorithm is applicable for *partly calibrated* arrays. However, the array must exhibit the *shift-invariance* property, refer to Section 2.3.1. Fig. 3.1 demonstrates the array topology considered for the DESPRIT Algorithm. In Fig. 3.1, a shift-invariant array of $K = 6$ subarrays is shown. The red arrows denote the communication links between the neighboring subarrays. In order to explain the DESPRIT algorithm, the DOA estimation procedure based on the ESPRIT algorithm, as reviewed in Section 2.3.2, is decomposed into the following steps:

Step 1: Computing the sample estimate of the signal eigenvector matrix $\hat{\boldsymbol{U}}_{\mathrm{s}}$ in (2.16).

Step 2: Obtaining the matrices $\hat{\overline{\boldsymbol{U}}}_{\mathrm{s}}$ and $\hat{\underline{\boldsymbol{U}}}_{\mathrm{s}}$ in (2.44) and (2.45), respectively, and computing the matrix $\hat{\boldsymbol{\Psi}}$ in (2.47).

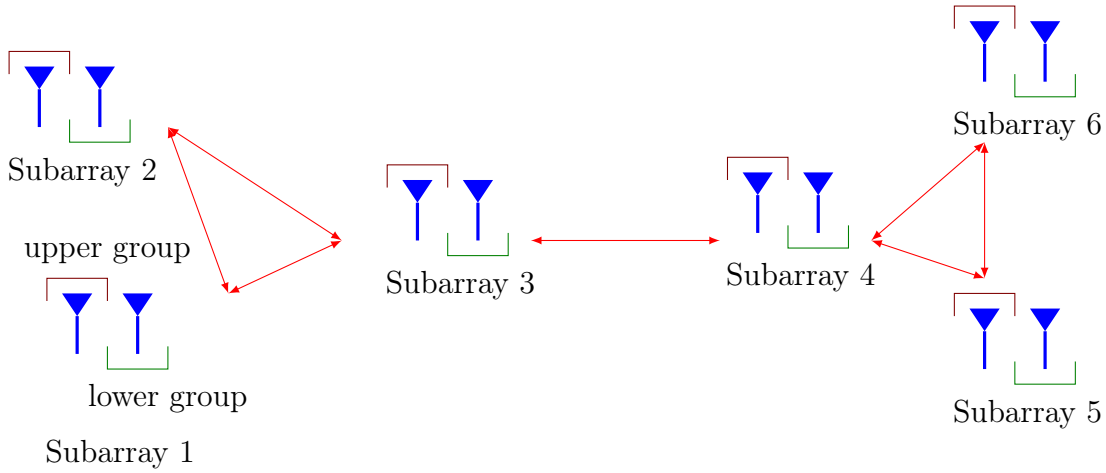

Fig. 3.1. A shift-invariant array of $K = 6$ subarrays. The red arrows indicate the communication links between the neighboring subarrays. The array is assumed to be partly calibrated, i.e., the displacements between the subarrays are assumed to be unknown.

Step 3: Eigendecomposition of the matrix $\hat{\boldsymbol{\Psi}}$ and the computation of the DOA estimates in (2.48).

The decentralized computation of $\hat{\boldsymbol{U}}_{\mathrm{s}}$ in Step 1 is carried out using the DPM as explained in Section 2.4.3. The resulting decentralized estimate of $\hat{\boldsymbol{U}}_{\mathrm{s}}$, denoted as $\check{\boldsymbol{U}}_{\mathrm{s}}$, is distributed among the subarrays, where each subarray stores only the rows of $\check{\boldsymbol{U}}_{\mathrm{s}}$ which correspond to its measurements. Based on the AC protocol, the matrix $\hat{\boldsymbol{\Psi}}$ in Step 2 can be computed in a fully decentralized fashion such that each subarray maintains all entries of the matrix $\hat{\boldsymbol{\Psi}}$, denote the computed matrix at the kth subarray as $\check{\boldsymbol{\Psi}}_{[k]}$, where the notation of Section 2.4.2 is used. The computation of $\check{\boldsymbol{\Psi}}_{[k]}$ is obtained by rewriting (2.47) as

$$\check{\boldsymbol{M}}_{1,[k]}\check{\boldsymbol{\Psi}}_{[k]} = \check{\boldsymbol{M}}_{2,[k]}, \tag{3.1}$$

where $\check{\boldsymbol{M}}_{1,[k]}$ and $\check{\boldsymbol{M}}_{2,[k]}$ are respectively the decentralized estimate of $\overline{\boldsymbol{U}}_{\mathrm{s}}{}^{H}\overline{\boldsymbol{U}}_{\mathrm{s}}$ and $\overline{\boldsymbol{U}}_{\mathrm{s}}{}^{H}\underline{\boldsymbol{U}}_{\mathrm{s}}$, at the kth subarray. The AC protocol is used to compute each entry of the matrices $\check{\boldsymbol{M}}_{1,[k]}$ and $\check{\boldsymbol{M}}_{2,[k]}$ as follows:

$$\check{m}_{1,i,j,[k]} = K\left[\boldsymbol{W}^{P_4}\begin{bmatrix}\check{\boldsymbol{u}}_{i,1}^{H}\overline{\boldsymbol{J}}_1^{H}\overline{\boldsymbol{J}}_1\check{\boldsymbol{u}}_{j,1}\\ \vdots \\ \check{\boldsymbol{u}}_{i,K}^{H}\overline{\boldsymbol{J}}_K^{H}\overline{\boldsymbol{J}}_K\check{\boldsymbol{u}}_{j,K}\end{bmatrix}\right]_k \tag{3.2}$$

and

$$\check{m}_{2,i,j,[k]} = K\left[\boldsymbol{W}^{P_4}\begin{bmatrix}\check{\boldsymbol{u}}_{i,1}^{H}\overline{\boldsymbol{J}}_1^{H}\underline{\boldsymbol{J}}_1\check{\boldsymbol{u}}_{j,1}\\ \vdots \\ \check{\boldsymbol{u}}_{i,K}^{H}\overline{\boldsymbol{J}}_K^{H}\underline{\boldsymbol{J}}_K\check{\boldsymbol{u}}_{j,K}\end{bmatrix}\right]_k, \tag{3.3}$$

where $\check{m}_{1,i,j,[k]}$ and $\check{m}_{2,i,j,[k]}$ denote the (i,j)th entries of the matrices $\check{\boldsymbol{M}}_{1,[k]}$ and $\check{\boldsymbol{M}}_{2,[k]}$, respectively, P_4 denotes the number of AC iterations used to compute $\check{m}_{1,i,j,[k]}$ and $\check{m}_{2,i,j,[k]}$, $\overline{\boldsymbol{J}}$ and $\underline{\boldsymbol{J}}$ defined in (2.38) and (2.39), respectively, and $\boldsymbol{W}$ is the averaging matrix defined in (2.55). Thus, the kth subarray can estimate the matrix $\boldsymbol{\Psi}$ locally as in (3.1). Finally, the DOA estimation in Step 3 is carried out locally in the kth subarray using the eigenvalues of the matrix $\check{\boldsymbol{\Psi}}_{[k]}$ in (2.48).

3.1.1 Communication Cost Analysis

In the DESPRIT algorithm, communication among the subarrays are required for:

1. Computing the signal eigenvectors matrix $\check{\boldsymbol{U}}_{\mathrm{s}}$ in Step 1 using the DPM which is analyzed in Section 2.4.4.

2. Computing $\check{\boldsymbol{M}}_{1,[k]}$ and $\check{\boldsymbol{M}}_{2,[k]}$, for $k = 1, \ldots, K$, in Step 2. Since one instance of the AC protocol is required to compute each entry of the matrices $\check{\boldsymbol{M}}_{1,[k]}$ and $\check{\boldsymbol{M}}_{2,[k]}$, for $k = 1, \ldots, K$, the overall communication cost required to compute these two matrices is $2L^2$ AC protocols, i.e., $2L^2 P_4$ AC iterations.

3.1.2 DOA Estimation for Arbitrary Array Geometries

Similar as in the conventional ESPRIT algorithm [RK89], the DESPRIT algorithm, introduced in Section 3.1, requires a shift-invariant array geometry. In this section, towards extending the DESPRIT algorithm to arbitrary array geometries, we introduce an array interpolation approach. In the proposed approach, the array interpolation is carried out locally at the subarrays, thus, data exchange among the subarrays is required for the DOA estimation but not for the array interpolation. The proposed algorithm achieves better performance than the conventional DESPRIT algorithm in perturbed shift-invariant arrays[1].

[1]Perturbed shift-invariant arrays can be thought of as a shift-invariant array where the sensor locations contain small (compared to the wavelength of the carried frequency) errors. Ignoring these small errors the array will be perfectly shift-invariant

3.1.2.1 Array Interpolation for Fully Calibrated arrays

In the linear interpolation technique [Fri90], the virtual array manifold is obtained by linear transformation of the true array manifold over a given angular sector. Assuming a *fully calibrated* array as in [Fri90], the virtual array manifold sampled at angular positions $\tilde{\boldsymbol{\theta}}$, defined in (2.24), is expressed by $\overline{\boldsymbol{A}}\ (\tilde{\boldsymbol{\theta}})$. This manifold can be written as

$$\overline{\boldsymbol{A}}\ (\tilde{\boldsymbol{\theta}}) = \boldsymbol{\mathfrak{I}}\boldsymbol{A}(\tilde{\boldsymbol{\theta}}), \tag{3.4}$$

where $\boldsymbol{\mathfrak{I}}$ is the interpolation matrix and $\boldsymbol{A}(\tilde{\boldsymbol{\theta}})$ is the true array manifold computed at the grid points $\tilde{\theta}_1, \ldots, \tilde{\theta}_G$. The structure of the virtual array, represented by $\overline{\boldsymbol{A}}\ (\tilde{\boldsymbol{\theta}})$, is chosen according to the used DOA estimation algorithm. For example, in [Fri90], the virtual array geometry is chosen as a ULA since the Root-MUSIC algorithm is applied to the resulting transformed measurements. Whereas, in [WG91] and [BPB03] the virtual array is chosen as a shifted version of the true array and the ESPRIT algorithm is used for DOA estimation. Since both matrices $\overline{\boldsymbol{A}}\ (\tilde{\boldsymbol{\theta}})$ and $\boldsymbol{A}(\tilde{\boldsymbol{\theta}})$ are known, the transformation matrix $\boldsymbol{\mathfrak{I}}$ can be computed from (3.4) using the Least Squares (LS) method. More sophisticated interpolation methods for sectorized interpolation designs are known, see e.g., [PGL02].

3.1.2.2 The Interpolated DESPRIT Algorithm

Since in the DESPRIT algorithm a *partly calibrated* array is assumed, i.e., displacements between the subarrays are unknown, the true array manifold is not completely known as a function of the DOAs. Thus, both matrices $\boldsymbol{A}$ and $\overline{\boldsymbol{A}}$ in (3.4) are not fully known. Nevertheless, in the sequel, a novel array interpolation method is introduced in which the interpolation is applied in the partly calibrated array scenario.

As in [WG91, BPB03], the virtual array manifold is assumed to be a shifted version of the true array manifold which can be expressed at the kth subarray as

$$\overline{\boldsymbol{A}}_k\ (\tilde{\boldsymbol{\theta}}, \boldsymbol{\zeta}_k) = \boldsymbol{A}_k(\tilde{\boldsymbol{\theta}}, \boldsymbol{\zeta}_k)\tilde{\boldsymbol{\Omega}}(\tilde{\boldsymbol{\theta}}, d), \tag{3.5}$$

where

$$\tilde{\boldsymbol{\Omega}}(\tilde{\boldsymbol{\theta}}, d) = \mathrm{diag}\Big(\exp\big(\jmath\frac{2\pi}{\lambda_c} d \sin(\tilde{\theta}_1)\big), \ldots, \exp\big(\jmath\frac{2\pi}{\lambda_c} d \sin(\tilde{\theta}_G)\big)\Big) \tag{3.6}$$

and d is the shift between the true and the virtual subarray, refer to (2.43). Substituting (3.5) in (3.4), yields

$$\boldsymbol{A}_k(\tilde{\boldsymbol{\theta}}, \boldsymbol{\zeta}_k)\tilde{\boldsymbol{\Omega}}(\tilde{\boldsymbol{\theta}}, d) = \boldsymbol{\mathfrak{I}}_k\boldsymbol{A}_k(\tilde{\boldsymbol{\theta}}, \boldsymbol{\zeta}_k). \tag{3.7}$$

Using the factorization (2.7) in (3.7), results in

$$\boldsymbol{V}_k(\tilde{\boldsymbol{\theta}})\boldsymbol{\Phi}_k(\tilde{\boldsymbol{\theta}},\boldsymbol{\zeta}_k)\tilde{\boldsymbol{\Omega}}(\tilde{\boldsymbol{\theta}},d) = \boldsymbol{\mathfrak{I}}_k\boldsymbol{V}_k(\tilde{\boldsymbol{\theta}})\boldsymbol{\Phi}_k(\tilde{\boldsymbol{\theta}},\boldsymbol{\zeta}_k). \tag{3.8}$$

Noting that $\boldsymbol{\Phi}_k(\tilde{\boldsymbol{\theta}},\boldsymbol{\zeta}_k)\tilde{\boldsymbol{\Omega}}(\tilde{\boldsymbol{\theta}},d) = \tilde{\boldsymbol{\Omega}}(\tilde{\boldsymbol{\theta}},d)\boldsymbol{\Phi}_k(\tilde{\boldsymbol{\theta}},\boldsymbol{\zeta}_k)$ since both matrices $\boldsymbol{\Phi}_k(\tilde{\boldsymbol{\theta}},\boldsymbol{\zeta}_k)$ and $\tilde{\boldsymbol{\Omega}}(\tilde{\boldsymbol{\theta}},d)$ are diagonal and invertible, (3.8) is reduced to

$$\boldsymbol{V}_k(\tilde{\boldsymbol{\theta}})\tilde{\boldsymbol{\Omega}}(\tilde{\boldsymbol{\theta}},d) = \boldsymbol{\mathfrak{I}}_k\boldsymbol{V}_k(\tilde{\boldsymbol{\theta}}), \tag{3.9}$$

where both matrices $\boldsymbol{V}_k(\tilde{\boldsymbol{\theta}})$ and $\tilde{\boldsymbol{\Omega}}(\tilde{\boldsymbol{\theta}},d)$ are known. Thus, the transformation matrix $\boldsymbol{\mathfrak{I}}_k$ can be computed using the LS method locally at the kth subarray, since the matrix $\boldsymbol{\mathfrak{I}}_k$ depends only on the (locally available) matrices $\boldsymbol{V}_k(\tilde{\boldsymbol{\theta}})$ and $\tilde{\boldsymbol{\Omega}}(\tilde{\boldsymbol{\theta}},d)$. Note that the matrix $\boldsymbol{\mathfrak{I}}_k$ is computed only once, unless the physical subarray geometry has been modified.

Observe that, the transformation introduced in (3.7) can also be applied jointly to the whole array, i.e., as $\boldsymbol{A}(\tilde{\boldsymbol{\theta}},\boldsymbol{\zeta})\tilde{\boldsymbol{\Omega}}(\tilde{\boldsymbol{\theta}},d) = \boldsymbol{\mathfrak{I}}\boldsymbol{A}(\tilde{\boldsymbol{\theta}},\boldsymbol{\zeta})$, which due to the increased number of interpolation parameters, yields improved interpolation quality. However, the resulting interpolation matrix $\boldsymbol{\mathfrak{I}}$ depends on the geometry of all the subarrays through the matrices $\boldsymbol{V}_1(\tilde{\boldsymbol{\theta}}),\ldots,\boldsymbol{V}_K(\tilde{\boldsymbol{\theta}})$ and, thus, it cannot be computed locally. Consequently, to avoid the communication load associated with the joint interpolation, the local interpolation in (3.7) is used.

The DESPRIT algorithm in Section 3.1 can be generalized to arbitrary array geometries using the interpolation matrices $\boldsymbol{\mathfrak{I}}_1,\ldots,\boldsymbol{\mathfrak{I}}_K$ and the signal eigenvectors $\check{\boldsymbol{u}}_1,\ldots,\check{\boldsymbol{u}}_L$ obtained from the DPM as follows. Firstly, the kth part of the lth signal eigenvector of the virtual array is computed as

$$\overline{\boldsymbol{u}}_{l,k} = \boldsymbol{\mathfrak{I}}_k\check{\boldsymbol{u}}_{l,k}, \tag{3.10}$$

for $l = 1,\ldots,L$ and $k = 1,\ldots,K$, refer to (2.59), where $\check{\boldsymbol{u}}_{l,k}$ is the kth part of the lth eigenvector, i.e., the part corresponding to the kth subarray. Secondly, the DOA estimation follows as in the DESPRIT algorithm in Section 3.1 where the signal eigenvectors of the true and virtual arrays are used as the upper and lower signal eigenvector matrices, respectively. The resulting algorithm is referred to as the interpolated DESPRIT algorithm (IDESPRIT).

3.2 The Decentralized Root-MUSIC Algorithm

The decentralized Root-MUSIC algorithm performs a fully decentralized DOA estimation based on the Root-MUSIC algorithm [Bar83] and the DPM [SPK08]. Similar to

Fig. 3.2. A ULA composed of K subarrays. Displacements between the subarrays are uniform and assumed to be known, i.e., the array is *fully calibrated*. The red lines denote communication links between the neighboring subarrays.

the Root-MUSIC algorithm, the decentralized Root-MUSIC algorithm requires a *fully calibrated ULA*. Fig. 3.2 demonstrates the topology of a ULA composed of K subarrays. In contrast to the topology in Fig. 3.1 (used in the DESPRIT algorithm), the displacements between the subarrays are assumed to be known since the array is a ULA. The red lines in Fig. 3.2 indicate the communication links between the neighboring subarrays.

Similar to the DESPRIT algorithm, in the decentralized Root-MUSIC algorithm, the DPM is utilized to obtain the decentralized estimate of the signal eigenvectors matrix $\check{\boldsymbol{U}}_{\mathrm{s}}$. However, in contrast to the DESPRIT algorithm, in the decentralized Root-MUSIC algorithm, the task to estimate the DOAs from $\check{\boldsymbol{U}}_{\mathrm{s}}$ is carried out at a processing center (PC). Note that the matrix $\check{\boldsymbol{U}}_{\mathrm{s}}$ is distributed among all the subarrays where the kth subarray stores M_k rows of the matrix $\check{\boldsymbol{U}}_{\mathrm{s}}$. Thus, each subarray must send its locally stored part of the matrix $\check{\boldsymbol{U}}_{\mathrm{s}}$ to the PC. Therefore, in the decentralized Root-MUSIC algorithm, decentralized DOA estimation with PC is carried out, refer to Section 1.1. After receiving all rows of the matrix $\check{\boldsymbol{U}}_{\mathrm{s}}$, the PC computes the decentralized Root-MUSIC polynomial $\check{\mathcal{F}}(z)$ using $\check{\boldsymbol{\Pi}}_{\mathrm{n}} = \boldsymbol{I}_M - \check{\boldsymbol{U}}_{\mathrm{s}}\check{\boldsymbol{U}}_{\mathrm{s}}^H$ in (2.20). The DOA estimation follows from the L roots of the polynomial $\check{\mathcal{F}}(z)$, denoted by $\check{z}_1, \ldots, \check{z}_L$, which are inside the unit circle and have the largest magnitude using (2.21).

3.2.1 Communication Cost Analysis

The communication required to achieve DOA estimation using the decentralized Root-MUSIC algorithm includes:

1. Computing the signal eigenvectors using the DPM which is analyzed in Section 2.4.4.
2. Communicating the estimated signal eigenvectors to the PC which is at most ML complex numbers.

3.3 Performance Analysis of the Decentralized DOA Estimation Algorithms

The aforementioned decentralized DOA estimation algorithms, namely, the DESPRIT and the decentralized Root-MUSIC algorithms, are based on the DPM. In order to assess the performance of the aforementioned decentralized DOA estimation algorithms, the statistical performance of the DPM needs to be studied. Thus in this section, performance analysis of the DPM is presented. An asymptotic analytical expression of the second order statistics of the eigenvectors and eigenvalues for the sample covariance matrix obtained from the DPM are derived. The expressions that are derived do not require a special array structure, e.g., shift-invariant or ULA, and are not restricted to the DOA estimation problem. Rather they are applicable for the decentralized eigendecomposition of the sample measurement covariance matrix using the DPM. Based on these expressions, the performance analysis of the DESPRIT and decentralized Root-MUSIC algorithms is derived.

3.3.1 The Decentralized Power Method

The key idea in the performance analysis of the DPM, lies in reformulating the DPM as an equivalent centralized PM. Based on the centralized formulation, asymptotic analytical expressions of the second order statistics of the eigenvectors and eigenvalues for the sample covariance matrix obtained from the DPM are derived. Moreover, the derived expressions reveal that the DPM is not a consistent estimator of the eigenvectors and eigenvalues of the true measurement covariance matrix $\boldsymbol{R}$.

Assumptions

The performance analysis of the DPM focuses on the errors resulting from using a finite number of AC iterations $P < \infty$ to compute the scalars $\{\check{x}_{t,l,[k]}^{(q)}\}_{t=1}^{N}$ in (2.64), because, from a signaling perspective, this step represents the most expensive calculation in the DPM, see Section 2.4.4. Thus, the following assumptions are made:

A1: The number of AC iterations P_1, P_2, and P_3 used to compute the scalars in (2.67), the normalization factors in (2.68), and the scalar in (2.71) respectively, are large compared to the number of AC iterations used to compute the scalars $\{\check{x}_{t,l,[k]}^{(q)}\}_{t=1}^{N}$,

i.e., $P_1 \gg P$, $P_2 \gg P$, and $P_3 \gg P$. Thus, errors resulting from the finite number of AC iterations in (2.67), (2.68), and (2.71) are negligible compared to those in (2.64).

A2: The number of PM iterations Q is sufficiently large such that the errors resulting from the finite number of PM iterations are negligible[2].

Error Expressions for the DPM

The decentralized eigendecomposition of the sample covariance matrix using the DPM yields the vectors $\{\check{\boldsymbol{u}}_l\}_{l=1}^M$ and the values $\{\check{\lambda}_l\}_{l=1}^M$, as explained in Section 2.4.3. Since under Assumptions A1 and A2 these vectors and values depend on P and not on P_1, P_2, P_3 and Q, they are denote as $\{\check{\boldsymbol{u}}_l(P)\}_{l=1}^M$ and $\{\check{\lambda}_l(P)\}_{l=1}^M$ respectively. Due to finite AC iteration effects ($P < \infty$), $\{\check{\boldsymbol{u}}_l(P)\}_{l=1}^M$ and $\{\check{\lambda}_l(P)\}_{l=1}^M$ do not exactly correspond to the eigenvectors and eigenvalues of the matrix $\hat{\boldsymbol{R}}$. The following theorem provides further insights into the properties of $\{\check{\boldsymbol{u}}_l(P)\}_{l=1}^M$ and $\{\check{\lambda}_l(P)\}_{l=1}^M$.

Theorem 1. *Under Assumptions A1 and A2, the vectors $\{\check{\boldsymbol{u}}_l(P)\}_{l=1}^M$ and the values $\{\check{\lambda}_l(P)\}_{l=1}^M$ are the eigenvectors and eigenvalues of the matrix*

$$\check{\boldsymbol{R}}(P) = K\left(\boldsymbol{T}\boldsymbol{W}^P\boldsymbol{T}^T\right) \odot \hat{\boldsymbol{R}}, \tag{3.11}$$

where the sensor selection matrix $\boldsymbol{T}$ is defined in (2.2), the centralized sample covariance matrix $\hat{\boldsymbol{R}}$ is defined in (2.11), and the AC weighting matrix $\boldsymbol{W}$ is defined in (2.55).

Proof. See Appendix A.1. □

Theorem 1 shows that, when the DPM is used with a finite number of samples N and a finite number of AC iterations P to estimate the eigenvectors $\{\boldsymbol{u}_l\}_{l=1}^M$ of the true covariance matrix $\boldsymbol{R}$, three different types of errors occur:

E1: Errors resulting from the finite number of AC iterations P. These errors are expressed in the matrix $\left(\boldsymbol{T}\boldsymbol{W}^P\boldsymbol{T}^T\right)$.

E2: Errors resulting from the finite number of samples N. These errors are expressed in $\hat{\boldsymbol{R}}$.

[2]Later, in the simulations, we show that, a moderate value of $Q = 10$ iteration is sufficient in the sense that the PM errors are negligible.

E3: Errors resulting from the finite number of PM iterations Q, which are neglected as stated in Assumption A2.

Since the averaging matrix $\boldsymbol{W}$ is assumed to satisfy the convergence condition (2.56), it follows that $\lim_{P\to\infty} K\boldsymbol{T}\boldsymbol{W}^P\boldsymbol{T}^T = \mathbf{1}_M\mathbf{1}_M^T$, where $\mathbf{1}_M\mathbf{1}_M^T$ is the all one matrix of size $M \times M$. Consequently, $\lim_{P\to\infty} \check{\boldsymbol{R}}(P) = \hat{\boldsymbol{R}}$. Note that $\lim_{N\to\infty} \check{\boldsymbol{R}}(P) = K\left(\boldsymbol{T}\boldsymbol{W}^P\boldsymbol{T}^T\right) \odot \boldsymbol{R}$, i.e., for a finite number of AC iterations P, the eigendecomposition of the sample covariance matrix using the DPM is not an asymptotically consistent estimator of the eigenvectors and eigenvalues of the true measurement covariance matrix $\boldsymbol{R}$, unless the number of AC iterations approaches infinity.

Theorem 1 simplifies the performance analysis of the DPM, since it provides a link to an equivalent centralized algorithm formulation, which can be analyzed using the conventional statistical analysis techniques and results [Bri81]. In the sequel, firstly, the error vectors and values which represent E1 and E2 types of errors are introduced. Secondly, the analytical expressions for these errors are computed. Finally, the second order statistics of the eigenvectors and eigenvalues obtained from the DPM are derived.

For the centralized eigendecomposition, the sample estimate of the lth eigenvector $\hat{\boldsymbol{u}}_l$ of the true covariance matrix $\boldsymbol{R}$ is expressed as

$$\hat{\boldsymbol{u}}_l = \boldsymbol{u}_l + \delta\boldsymbol{u}_l, \tag{3.12}$$

where the error vector $\delta\boldsymbol{u}_l$ accounts only for the finite sample effects, i.e., E2 type of errors, used in the computation of the sample covariance matrix $\hat{\boldsymbol{R}}$. The decentralized estimate of the lth eigenvector is expressed as

$$\check{\boldsymbol{u}}_l(P) = \boldsymbol{u}_l + \delta\check{\boldsymbol{u}}_l(P), \tag{3.13}$$

where the error vector $\delta\check{\boldsymbol{u}}_l(P)$ accounts for errors resulting from the finite number of samples and the finite number of AC iterations, i.e., E1 and E2 type of errors. Similarly, the centralized and decentralized sample estimates of the eigenvalues are defined as

$$\hat{\lambda}_l = \lambda_l + \delta\lambda_l \tag{3.14}$$

and

$$\check{\lambda}_l(P) = \lambda_l + \delta\check{\lambda}_l(P), \tag{3.15}$$

respectively, where the error term $\delta\lambda_l$ accounts for E2 type of errors and the error term $\delta\check{\lambda}_l(P)$ accounts for both E1 and E2 type of errors. Further, the matrices $\hat{\boldsymbol{R}}$ and $\check{\boldsymbol{R}}(P)$ are expressed as

$$\hat{\boldsymbol{R}} = \boldsymbol{R} + \delta\boldsymbol{R} \tag{3.16}$$

and

$$\check{\boldsymbol{R}}(P) = \boldsymbol{R} + \delta\check{\boldsymbol{R}}(P), \tag{3.17}$$

where $\delta\boldsymbol{R}$ accounts only for E2 type of errors and $\delta\check{\boldsymbol{R}}(P)$ accounts for both E1 and E2 types of errors. Using the aforementioned notation, the second order statistics of the eigenvectors computed using the DPM is expressed as $\mathbb{E}\Big(\delta\check{\boldsymbol{u}}_l(P)\delta\check{\boldsymbol{u}}_m^H(P)\Big)$. Similarly, the second order statistics of the eigenvalues computed using the DPM is expressed as $\mathbb{E}\Big(\delta\check{\lambda}_l(P)\delta\check{\lambda}_m^*(P)\Big)$. Towards computing the aforementioned second order statistics, a simple expressions for $\delta\check{\boldsymbol{u}}_l(P)$ and $\delta\check{\lambda}_l(P)$, based on a first order analysis, are introduced in the following theorem.

Theorem 2. *Under Assumptions A1 and A2, the errors $\delta\check{\boldsymbol{u}}_l(P)$ and $\delta\check{\lambda}_l(P)$ are given by*

$$\delta\check{\boldsymbol{u}}_l(P) = -\boldsymbol{\Xi}_l\left(\delta\boldsymbol{R}\,\boldsymbol{u}_l + \boldsymbol{\kappa}_l(P)\right), \tag{3.18}$$

and

$$\delta\check{\lambda}_l(P) = \boldsymbol{u}_l^H\delta\boldsymbol{R}\boldsymbol{u}_l + \boldsymbol{u}_l^H\boldsymbol{\kappa}_l(P), \tag{3.19}$$

where $\boldsymbol{\Xi}_l$ is $M \times M$ matrix defined in (2.13),

$$\boldsymbol{\kappa}_l(P) = K\sum_{k=2}^{K}\alpha_k^P\,\tilde{\boldsymbol{T}}_k\boldsymbol{R}\tilde{\boldsymbol{T}}_k^H\boldsymbol{u}_l, \tag{3.20}$$

$\tilde{\boldsymbol{T}}_k = \mathrm{diag}\big(\boldsymbol{T}\boldsymbol{\beta}_k\big)$, $\boldsymbol{\beta}_k$ and α_k are defined in (2.57), and $\boldsymbol{T}$ is defined in (2.2).

Proof. See Appendix A.2. □

The expressions of $\delta\check{\boldsymbol{u}}_l(P)$ and $\delta\check{\lambda}_l(P)$ consist of two terms. The first term represents the finite sample error $\delta\boldsymbol{R}$ and does not depend on the number of AC iterations P. This term is the same in the centralized case. The second term depends on the number of AC iterations P and does not exist in the centralized case. Note that in Theorem 2 the E1 type of errors are expressed in terms of the vector $\boldsymbol{\kappa}_l(P)$ which depends on the number of AC iterations P and on the eigenvalues and eigenvectors of the AC weighting matrix $\boldsymbol{W}$, except for the principal eigenvalue and eigenvector. Since the magnitude of α_k is strictly less than one for $k = 2, \ldots, K$ (see Section 2.4.2), it follows from (3.20) that $\lim_{P\to\infty}\boldsymbol{\kappa}_l(P) = \boldsymbol{0}$, i.e., when the AC protocol is carried out for an infinitely large number of iterations, the vector $\boldsymbol{\kappa}_l(P)$ converges to zero. Consequently, $\delta\check{\boldsymbol{u}}_l(P)$ and $\delta\check{\lambda}_l(P)$ contain no E1 type of errors when $P \to \infty$. In Theorem 2, the E2 errors are expressed in terms of the matrix $\delta\boldsymbol{R}$, where , $\lim_{N\to\infty}\delta\boldsymbol{R} = \boldsymbol{0}$, i.e., $\delta\boldsymbol{R}$ converge to zero when an infinite number of samples is available. Consequently, $\lim_{P,N\to\infty}\delta\check{\boldsymbol{u}}_l(P) = \boldsymbol{0}$ and $\lim_{P,N\to\infty}\delta\check{\lambda}_l(P) = 0$, i.e., $\delta\check{\boldsymbol{u}}_l(P)$ and $\delta\check{\lambda}_l(P)$ approach zero as both P and N tend to infinity.

Based on Theorem 2, analytical expressions for the second order statistics of the eigenvalues and eigenvectors of the DPM, i.e., $\mathbb{E}\Big(\delta\check{\boldsymbol{u}}_l(P)\delta\check{\boldsymbol{u}}_m^H(P)\Big)$ and $\mathbb{E}\Big(\delta\check{\lambda}_l(P)\delta\check{\lambda}_m^*(P)\Big)$, and an expression for $\mathbb{E}\Big(\delta\check{\boldsymbol{u}}_l(P)\delta\check{\boldsymbol{u}}_m^T(P)\Big)$ are introduced in the following theorem. These expressions are useful for computing the MSE of estimators which are based on the DPM as it is shown later for the DESPRIT and the decentralized Root-MUSIC algorithms.

Theorem 3. *Under Assumptions A1 and A2*

$$\mathbb{E}\Big(\delta\check{\boldsymbol{u}}_l(P)\delta\check{\boldsymbol{u}}_m^H(P)\Big) = \frac{\lambda_l}{N}\sum_{\substack{i=1\\ i\neq l}}^{M}\frac{\lambda_i}{(\lambda_l-\lambda_i)^2}\boldsymbol{u}_i\boldsymbol{u}_i^H\delta_{l,m} + \boldsymbol{\Xi}_l\boldsymbol{\kappa}_l(P)\boldsymbol{\kappa}_m^H\boldsymbol{\Xi}_m^H(P), \tag{3.21}$$

$$\mathbb{E}\Big(\delta\check{\boldsymbol{u}}_l(P)\delta\check{\boldsymbol{u}}_m^T(P)\Big) = \frac{\lambda_l\lambda_m}{N}\frac{\boldsymbol{u}_l\boldsymbol{u}_m^T}{(\lambda_l-\lambda_m)^2}(\delta_{l,m}-1) + \boldsymbol{\Xi}_l\boldsymbol{\kappa}_l(P)\boldsymbol{\kappa}_m^T(P)\boldsymbol{\Xi}_m^T, \tag{3.22}$$

and

$$\mathbb{E}\Big(\delta\check{\lambda}_l(P)\delta\check{\lambda}_m^*(P)\Big) = \frac{\lambda_l^2}{N}\delta_{l,m} + \boldsymbol{u}_l^H\boldsymbol{\kappa}_l(P)\boldsymbol{\kappa}_m^H(P)\boldsymbol{u}_m, \tag{3.23}$$

where $\delta_{l,m}$ is the Kronecker delta, N is the number of samples, $\boldsymbol{\Xi}_l$ is defined in (2.13), and $\boldsymbol{\kappa}_l$ is defined in (3.20).

Proof. See Appendix A.3. □

The expressions of $\mathbb{E}\Big(\delta\check{\boldsymbol{u}}_l(P)\delta\check{\boldsymbol{u}}_m^H(P)\Big)$, $\mathbb{E}\Big(\delta\check{\boldsymbol{u}}_l(P)\delta\check{\boldsymbol{u}}_m^T(P)\Big)$, and $\mathbb{E}\Big(\delta\check{\lambda}_l(P)\delta\check{\lambda}_m^*(P)\Big)$ consist of two terms. The first term represents the finite sample error and does not depend on the number of AC iterations P. This term decreases with the number of samples N as in the centralized case. The second term depends on the number of AC iterations P through the vectors $\boldsymbol{\kappa}_l(P)$ and $\boldsymbol{\kappa}_m(P)$, defined in (3.20), and as $P\to\infty$, these term converge to zero. Consequently, when $P\to\infty$, $\mathbb{E}\Big(\delta\check{\boldsymbol{u}}_l(P)\delta\check{\boldsymbol{u}}_m^H(P)\Big)$, $\mathbb{E}\Big(\delta\check{\boldsymbol{u}}_l(P)\delta\check{\boldsymbol{u}}_m^T(P)\Big)$, and $\mathbb{E}\Big(\delta\check{\lambda}_l(P)\delta\check{\lambda}_m^*(P)\Big)$ are the same as the corresponding centralized expressions derived in [Bri81]. Moreover, as $N\to\infty$ for $P<\infty$, $\mathbb{E}\Big(\delta\check{\boldsymbol{u}}_l(P)\delta\check{\boldsymbol{u}}_m^H(P)\Big)$, $\mathbb{E}\Big(\delta\check{\boldsymbol{u}}_l(P)\delta\check{\boldsymbol{u}}_m^T(P)\Big)$, and $\mathbb{E}\Big(\delta\check{\lambda}_l(P)\delta\check{\lambda}_m^*(P)\Big)$ do not converge to zero, i.e. the DPM is not a consistent estimator for $\{\boldsymbol{u}_l\}_{l=1}^M$ and $\{\lambda_l\}_{l=1}^M$, unless P is infinitely large. Theorem 3 shows that, in the second order statistics of the eigenvector and eigenvalue estimates, the AC errors appear as an additive error term, whereas in Theorem 1 the corresponding errors for the sample covariance matrix are expressed as an element-wise multiplication with the matrix $\big(\boldsymbol{T}\boldsymbol{W}^P\boldsymbol{T}^T\big)$. The simplified expressions of $\mathbb{E}\Big(\delta\check{\boldsymbol{u}}_l(P)\delta\check{\boldsymbol{u}}_m^H(P)\Big)$, $\mathbb{E}\Big(\delta\check{\boldsymbol{u}}_l(P)\delta\check{\boldsymbol{u}}_m^T(P)\Big)$, and $\mathbb{E}\Big(\delta\check{\lambda}_l(P)\delta\check{\lambda}_m^*(P)\Big)$ in Theorem 3 facilitate the performance analysis of algorithms which are based on the

DPM, as shown in the following two subsections for the DESPRIT and decentralized Root-MUSIC algorithms.

Note that in practice P can not be chosen to be arbitrarily large, thus, the second terms in $\mathbb{E}\Big(\delta\check{\boldsymbol{u}}_l(P)\delta\check{\boldsymbol{u}}_m^H(P)\Big)$, $\mathbb{E}\Big(\delta\check{\boldsymbol{u}}_l(P)\delta\check{\boldsymbol{u}}_m^T(P)\Big)$, and $\mathbb{E}\Big(\delta\check{\lambda}_l(P)\delta\check{\lambda}_m^*(P)\Big)$ will always be non-zero. However, P can usually be chosen such that the second terms in $\mathbb{E}\Big(\delta\check{\boldsymbol{u}}_l(P)\delta\check{\boldsymbol{u}}_m^H(P)\Big)$, $\mathbb{E}\Big(\delta\check{\boldsymbol{u}}_l(P)\delta\check{\boldsymbol{u}}_m^T(P)\Big)$, and $\mathbb{E}\Big(\delta\check{\lambda}_l(P)\delta\check{\lambda}_m^*(P)\Big)$ are of the same order as the first terms. The proper choice of P will be further addressed in the simulations in Section 3.4.1.

3.3.2 The Decentralized ESPRIT Algorithm

To simplify the performance analysis of the DESPRIT algorithm, the following assumption is made:

A3: The number of AC iterations P_4 which is used to compute $\check{\boldsymbol{\Psi}}_{[k]}(P)$ in (3.1) from $\check{\boldsymbol{U}}_{\mathrm{s}}(P)$ is large compared to the number of AC iterations used to compute $\check{\boldsymbol{U}}_{\mathrm{s}}(P)$, i.e., $P_4 \gg P$, refer to (2.64).

Under Assumption A3, the AC errors in the decentralized estimate of the ESPRIT matrix $\boldsymbol{\Psi}$, defined in (2.47), are negligible compared to those in the decentralized eigenvector estimates $\check{\boldsymbol{U}}_{\mathrm{s}}(P)$. Thus, the decentralized estimates of the matrix $\boldsymbol{\Psi}$ can be written as

$$\check{\boldsymbol{\Psi}}_{[k]}(P) = \check{\boldsymbol{\Psi}}(P) = \Big(\check{\overline{\boldsymbol{U}}}_{\mathrm{s}}^H(P)\check{\overline{\boldsymbol{U}}}_{\mathrm{s}}(P)\Big)^{-1}\check{\overline{\boldsymbol{U}}}_{\mathrm{s}}^H(P)\check{\underline{\boldsymbol{U}}}_{\mathrm{s}}(P), \tag{3.24}$$

for $k = 1, \ldots, K$, where $\check{\overline{\boldsymbol{U}}}_{\mathrm{s}}(P) = \overline{\boldsymbol{J}}\check{\boldsymbol{U}}_{\mathrm{s}}(P)$ and $\check{\underline{\boldsymbol{U}}}_{\mathrm{s}}(P) = \underline{\boldsymbol{J}}\check{\boldsymbol{U}}_{\mathrm{s}}(P)$. Let $\check{\psi}_l(P)$, for $l = 1, \ldots, L$, be the eigenvalues of $\check{\boldsymbol{\Psi}}(P)$. In the DESPRIT algorithm, $\check{\psi}_l(P)$ is used as an estimate of the lth eigenvalue of the matrix $\boldsymbol{\Psi}$, denoted as ψ_l in (2.48). Thus, the estimation error $\delta\check{\psi}_l(P)$ is defined as

$$\check{\psi}_l(P) = \psi_l + \delta\check{\psi}_l(P), \tag{3.25}$$

for $l = 1, \ldots, L$, where the error term $\delta\check{\psi}_l(P)$ accounts for both E1 and E2 types of errors.

In [RH89a], the MSE of DOA estimation using the conventional Least Squares ESPRIT is presented. Assumption A3 allows the usage of the analysis from [RH89a] by

replacing $\mathbb{E}\Big(\delta\boldsymbol{u}_l\,\delta\boldsymbol{u}_m^H\Big)$ and $\mathbb{E}\Big(\delta\boldsymbol{u}_l\,\delta\boldsymbol{u}_m^T\Big)$ with $\mathbb{E}\Big(\delta\check{\boldsymbol{u}}_l(P)\delta\check{\boldsymbol{u}}_m^H(P)\Big)$ and $\mathbb{E}\Big(\delta\check{\boldsymbol{u}}_l(P)\delta\check{\boldsymbol{u}}_m^T(P)\Big)$, respectively. Thus, the following result from [RH89a] holds,

$$\mathbb{E}\big((\delta\check{\boldsymbol{\theta}}_l(P))^2\big) = \frac{\mathbb{E}\big(|\delta\check{\psi}_l(P)|^2\big) - \mathrm{Re}\Big(\big(\psi_l^*\big)^2\mathbb{E}\big((\delta\check{\psi}_l(P))^2\big)\Big)}{2\left(2\pi d\cos(\theta_l)/\lambda_c\right)^2}, \tag{3.26}$$

for $l = 1, \ldots, L$, where

$$\mathbb{E}\big(|\delta\check{\psi}_l(P)|^2\big) = \boldsymbol{\gamma}_l^H\mathbb{E}\big(\delta\check{\boldsymbol{U}}_{\mathrm{s}}(P)\;\vec{\boldsymbol{\psi}}_l\vec{\boldsymbol{\psi}}_l^H\;\delta\check{\boldsymbol{U}}_{\mathrm{s}}^H(P)\big)\boldsymbol{\gamma}_l, \tag{3.27}$$

$$\mathbb{E}\big((\delta\check{\psi}_l(P))^2\big) = \boldsymbol{\mu}_l^H\mathbb{E}\big(\delta\check{\boldsymbol{U}}_{\mathrm{s}}(P)\;\vec{\boldsymbol{\psi}}_l\vec{\boldsymbol{\psi}}_l^T\;\delta\check{\boldsymbol{U}}_{\mathrm{s}}^T(P)\big)\boldsymbol{\mu}_l^*, \tag{3.28}$$

$\check{\boldsymbol{U}}_{\mathrm{s}}(P) = \boldsymbol{U}_{\mathrm{s}} + \delta\check{\boldsymbol{U}}_{\mathrm{s}}(P)$, $\boldsymbol{\gamma}_l^H = \overleftarrow{\boldsymbol{\psi}}_l\;\Big(\overline{\boldsymbol{U}}_{\mathrm{s}}^H\overline{\boldsymbol{U}}_{\mathrm{s}}\Big)^{-1}\overline{\boldsymbol{U}}_{\mathrm{s}}^H\left(\overline{\boldsymbol{J}} - \psi_l^*\underline{\boldsymbol{J}}\right)$ and $\boldsymbol{\mu}_l^H = \overleftarrow{\boldsymbol{\psi}}_l\Big(\overline{\boldsymbol{U}}_{\mathrm{s}}^H\overline{\boldsymbol{U}}_{\mathrm{s}}\Big)^{-1}\overline{\boldsymbol{U}}_{\mathrm{s}}^H\left(\underline{\boldsymbol{J}} - \psi_l\overline{\boldsymbol{J}}\right)$. The lth left and right eigenvectors which correspond to the lth eigenvalue of the matrix $\boldsymbol{\Psi}$ are denoted as $\overleftarrow{\boldsymbol{\psi}}_l$ and $\vec{\boldsymbol{\psi}}_l$, respectively. In (3.27) and (3.28), the conventional error $\delta\boldsymbol{U}_{\mathrm{s}}$ that is used in the corresponding expression of [RH89a] is replaced by $\delta\check{\boldsymbol{U}}_{\mathrm{s}}(P)$.

The expectation of the right hand side of (3.27) is rewritten using the expressions from Theorem 3 as

$$\begin{aligned}
\mathbb{E}\big(\delta\check{\boldsymbol{U}}_{\mathrm{s}}(P)\;\vec{\boldsymbol{\psi}}_l\vec{\boldsymbol{\psi}}_l^H\;\delta\check{\boldsymbol{U}}_{\mathrm{s}}^H(P)\big) &= \sum_{i=1}^{L}\sum_{j=1}^{L}\Big[\vec{\boldsymbol{\psi}}_l\vec{\boldsymbol{\psi}}_l^H\Big]_{i,j}\,\mathbb{E}\Big(\delta\check{\boldsymbol{u}}_l(P)\delta\check{\boldsymbol{u}}_m^H(P)\Big)\\
&= \frac{1}{N}\sum_{i=1}^{L}\sum_{\substack{j=1\\ j\neq i}}^{M}\frac{\lambda_i\lambda_j}{(\lambda_i-\lambda_j)^2}\Big[\vec{\boldsymbol{\psi}}_l\vec{\boldsymbol{\psi}}_l^H\Big]_{i,i}\boldsymbol{u}_i\boldsymbol{u}_i^H\\
&\quad+\sum_{i=1}^{L}\sum_{j=1}^{L}\Big[\vec{\boldsymbol{\psi}}_l\vec{\boldsymbol{\psi}}_l^H\Big]_{i,j}\boldsymbol{\Xi}_i\boldsymbol{\kappa}_i(P)\boldsymbol{\kappa}_j^H(P)\boldsymbol{\Xi}_j^H.
\end{aligned} \tag{3.29}$$

Similarly, the expectation of the right hand side of (3.28) is written as

$$\begin{aligned}
\mathbb{E}\big(\delta\check{\boldsymbol{U}}_{\mathrm{s}}(P)\;\vec{\boldsymbol{\psi}}_l\vec{\boldsymbol{\psi}}_l^T\;\delta\check{\boldsymbol{U}}_{\mathrm{s}}^T(P)\big) &= \sum_{i=1}^{L}\sum_{j=1}^{L}\Big[\vec{\boldsymbol{\psi}}_l\vec{\boldsymbol{\psi}}_l^T\Big]_{i,j}\,\mathbb{E}\Big(\delta\check{\boldsymbol{u}}_l(P)\delta\check{\boldsymbol{u}}_m^T(P)\Big)\\
&= -\sum_{i=1}^{L}\sum_{\substack{j=1\\ j\neq i}}^{L}\frac{[\vec{\boldsymbol{\psi}}_l\vec{\boldsymbol{\psi}}_l^T]_{i,j}\lambda_i\lambda_j\boldsymbol{u}_i\boldsymbol{u}_j^T}{N\left(\lambda_i-\lambda_j\right)^2}\\
&\quad+\sum_{i=1}^{L}\sum_{j=1}^{L}[\vec{\boldsymbol{\psi}}_l\vec{\boldsymbol{\psi}}_l^T]_{i,j}\boldsymbol{\Xi}_i\boldsymbol{\kappa}_i(P)\boldsymbol{\kappa}_j^T(P)\boldsymbol{\Xi}_j^T.
\end{aligned} \tag{3.30}$$

Equations (3.26)–(3.30) provide the analytical expression of the MSE for the DOA estimator using the DESPRIT algorithm. The second terms of (3.29) and (3.30) differ from the expressions in [RH89a]. Note that because of these terms, the MSE does not approach zero even if an infinitely large number of samples is available, i.e., the DESPRIT algorithm is not a consistent estimator of the DOAs. However, the simulations in Section 3.4.1 demonstrate that for a finite number of samples and a moderate SNR, a finite number of AC iterations is sufficient to achieve a performance comparable to that of the conventional ESPRIT algorithm.

3.3.3 The Decentralized Root-MUSIC Algorithm

In this section, an analytical expression for the MSE of the DOA estimates obtained from the decentralized Root-MUSIC algorithm is derived. The results introduced in this section are based on the analysis of the DPM in Theorem 3 and the analysis of the conventional Root-MUSIC algorithm in [RH89b, PGH00].

In the decentralized Root-MUSIC algorithm, introduced in Section 3.2, the Root-MUSIC polynomial, defined in (2.20) is evaluated at the PC. In other words, the computation of the polynomial $\check{\mathcal{F}}(z)$ from the decentralized estimate of the signal eigenvectors matrix $\check{\boldsymbol{U}}_{\mathrm{s}}$ contains no AC errors. Moreover, the DOA estimation from the polynomial $\check{\mathcal{F}}(z)$ as defined in (2.21) is carried out at the PC, i.e., the computation of (2.21) does not contain AC errors. Thus, in this case, the following result from [RH89b, Equation 26] holds,

$$\mathbb{E}\big((\delta\check{\boldsymbol{\theta}}_l(P))^2\big) = \left(\frac{1}{2\pi\overline{d}\cos(\theta_l)/\lambda_c}\right)^2 \frac{\pi^2 \mathrm{Re}(\Upsilon_1(P)+\Upsilon_2(P))}{2\big(\dot{\boldsymbol{a}}^H(\theta_l)\boldsymbol{\Pi}_{\mathrm{n}}\dot{\boldsymbol{a}}(\theta_l)\big)^2}, \tag{3.31}$$

where $\overline{d}$ is the inter sensor distance of the ULA, $\boldsymbol{\Pi}_{\mathrm{n}}$ is the projection into the true noise eigenvectors matrix, refer to (2.18), $\dot{\boldsymbol{a}}^H(\theta_l)$ is the derivative of the array response vector $\boldsymbol{a}(\theta_l)$ with respect to θ_l,

$$\begin{aligned}\Upsilon_1(P) = \sum_{m=1}^{L}\sum_{l=1}^{L} \Big(&\dot{\boldsymbol{a}}^H\boldsymbol{u}_l\dot{\boldsymbol{a}}^H\boldsymbol{u}_m\boldsymbol{a}^T\Big(\mathbb{E}\big(\delta\check{\boldsymbol{u}}_l(P)\delta\check{\boldsymbol{u}}_m^T(P)\big)\Big)^*\boldsymbol{a} \\ &+\dot{\boldsymbol{a}}^H\boldsymbol{u}_l\boldsymbol{u}_m^H\boldsymbol{a}\dot{\boldsymbol{a}}^H\mathbb{E}\big(\delta\check{\boldsymbol{u}}_m(P)\delta\check{\boldsymbol{u}}_l^H(P)\big)\boldsymbol{a} \\ &+\boldsymbol{u}_l^H\boldsymbol{a}\dot{\boldsymbol{a}}^H\boldsymbol{u}_m\dot{\boldsymbol{a}}^H\mathbb{E}\big(\delta\check{\boldsymbol{u}}_l(P)\delta\check{\boldsymbol{u}}_m^H(P)\big)\boldsymbol{a} \\ &+\boldsymbol{u}_l^H\boldsymbol{a}\boldsymbol{u}_m^H\boldsymbol{a}\dot{\boldsymbol{a}}^H\mathbb{E}\big(\delta\check{\boldsymbol{u}}_m(P)\delta\check{\boldsymbol{u}}_l^T(P)\big)\dot{\boldsymbol{a}}^*\Big),\end{aligned} \tag{3.32}$$

$$\begin{aligned}\Upsilon_2(P) = \sum_{m=1}^{L}\sum_{l=1}^{L}\Big(&\dot{\boldsymbol{a}}^H\boldsymbol{u}_l\boldsymbol{u}_m^H\dot{\boldsymbol{a}}\boldsymbol{a}^H\mathbb{E}\big(\delta\check{\boldsymbol{u}}_m(P)\delta\check{\boldsymbol{u}}_l^H(P)\big)\boldsymbol{a}\\ &+\dot{\boldsymbol{a}}^H\boldsymbol{u}_l\boldsymbol{a}^H\boldsymbol{u}_m\boldsymbol{a}^T\Big(\mathbb{E}\big(\delta\check{\boldsymbol{u}}_l(P)\delta\check{\boldsymbol{u}}_m^T(P)\big)\Big)^*\dot{\boldsymbol{a}}\\ &+\boldsymbol{u}_l^H\boldsymbol{a}\boldsymbol{u}_m^H\dot{\boldsymbol{a}}\dot{\boldsymbol{a}}^H\mathbb{E}\big(\delta\check{\boldsymbol{u}}_l(P)\delta\check{\boldsymbol{u}}_m^T(P)\big)\boldsymbol{a}^*\\ &+\boldsymbol{u}_l^H\boldsymbol{a}\boldsymbol{a}^H\boldsymbol{u}_m\dot{\boldsymbol{a}}\mathbb{E}\big(\delta\check{\boldsymbol{u}}_l(P)\delta\check{\boldsymbol{u}}_m^H(P)\big)\dot{\boldsymbol{a}}\Big),\end{aligned} \tag{3.33}$$

where the dependency on θ_l is omitted for ease of notation.

Equations (3.31)–(3.33) represent the analytical expression of the MSE for the DOA estimator using the decentralized Root-MUSIC algorithm. Note from Theorem 3 that for a finite number of AC iterations $P < \infty$, the error covariances $\mathbb{E}\big(\delta\check{\boldsymbol{u}}_l(P)\delta\check{\boldsymbol{u}}_m^H(P)\big)$ and $\mathbb{E}\big(\delta\check{\boldsymbol{u}}_l(P)\delta\check{\boldsymbol{u}}_m^T(P)\big)$ do not converge to zero as N approaches to infinity. Therefore, also the MSE of the DOA estimates obtained from the decentralized Root-MUSIC algorithm does not converge to zero as the number of observations tends to infinity. Consequently, the decentralized Root-MUSIC algorithm is not an asymptotically consistent estimator of the DOAs. However, in Section 3.4.3, it is demonstrated by simulation that a small number of AC iterations is sufficient to attain performance similar to that of the conventional Root-MUSIC algorithm [Bar83] when the number of snapshots N is small or the SNR is low.

3.4 Simulation Results

In this section, the performance of the DPM and the DESPRIT algorithm is demonstrated using a *partly calibrated* array configuration. The performance of the decentralized Root-MUSIC algorithms is demonstrated by simulations using a *fully calibrated* array, precisely, a ULA, since it is only applicable in such an array geometry. Further, the performance of the DESPRIT algorithm is also displayed in the ULA array configuration, since ULAs exhibit the shift-invariance property as shown in Section 2.3.1.

3.4.1 The DPM and DESPRIT Algorithm

An array composed of $K = 6$ subarrays each containing 2 sensors, i.e., $M = 12$, separated by half-wavelength is considered in the simulations. The locations of the first sensors at the subarrays are $(0, 0)$, $(0.45, 0.99)$, $(3.02, 0.45)$, $(5.61, 0.90)$, $(8.03, 1.46)$ and $(8.70, 0.50)$ measured in half-wavelength. The upper and lower selection matrices of the kth subarray are $\overline{\boldsymbol{J}}_k = [1, 0]$ and $\underline{\boldsymbol{J}}_k = [0, 1]$. The array topology depicted in

Fig. 3.1 is assumed. Thus, the neighboring subarray sets are $\mathcal{N}_1 = \{2, 3\}$, $\mathcal{N}_2 = \{1, 3\}$, $\mathcal{N}_3 = \{1, 2, 4\}$, $\mathcal{N}_4 = \{3, 5, 6\}$, $\mathcal{N}_5 = \{4, 6\}$, and $\mathcal{N}_6 = \{4, 5\}$, where the kth subarray communicates only with its neighbors $\mathcal{N}_k$. The entries of the weighting matrix $\boldsymbol{W}$ are selected as follows:

$$[\boldsymbol{W}]_{i,j} = \begin{cases} 1/\max\{\mathrm{card}(\mathcal{N}_i), \mathrm{card}(\mathcal{N}_j)\}, & \text{if } i \neq j \text{ and } j \in \mathcal{N}_i \\ w_i, & \text{if } i = j \\ 0, & \text{otherwise,} \end{cases} \tag{3.34}$$

for $i, j = 1, \ldots, K$, where $\mathrm{card}(\mathcal{N}_i)$ is the number of elements in the set $\mathcal{N}_i$. The weighting factors $\{w_i\}_{i=1}^K$ are chosen as $w_i = 1 - \sum_{j=1, j\neq i}^K [\boldsymbol{W}]_{i,j}$, refer to [XB04] for further details. This choice of the weighting factors only requires that each node knows the degree of its neighbors, thus, local but not global knowledge about the network topology is required at the node level. The weighting matrix $\boldsymbol{W}$ resulting from the weighting scheme in (3.34) guarantees asymptotic convergence of the AC protocol, provided that the graph associated with the network is not bipartite [XB04].

Signals from $L = 3$ equal-powered Gaussian sources impinge onto the array from directions $-14°, -10°$ and $5°$. In the sequel, the analytical expressions for the performance of the DPM and the DESPRIT algorithm are evaluated.

Performance of the DPM

In the first set of simulations, shown in Fig. 3.3, Fig. 3.4 and Fig. 3.5, the performance of the DPM is evaluated as follows. The signal eigenvectors of the sample covariance matrix are estimated at the ith realization using the DPM, i.e., the vectors $\check{\boldsymbol{U}}_{\mathrm{s}}(i) = [\check{\boldsymbol{u}}_1(i), \check{\boldsymbol{u}}_2(i), \check{\boldsymbol{u}}_3(i)]$, for $i = 1, \ldots, 500$ realizations[3], are computed. Then, the normalized square error (SE) of the DPM at the ith realization is computed using

$$\mathrm{SE}_{\mathrm{DPM}}(i) = \mathrm{tr}\big(\delta\check{\boldsymbol{U}}_{\mathrm{s}}(i)\delta\check{\boldsymbol{U}}_{\mathrm{s}}^H(i)\big)/\mathrm{tr}\big(\boldsymbol{U}_{\mathrm{s}}\boldsymbol{U}_{\mathrm{s}}^H\big), \tag{3.35}$$

where $\delta\check{\boldsymbol{U}}_{\mathrm{s}}(i) = \check{\boldsymbol{U}}_{\mathrm{s}}(i) - \boldsymbol{U}_{\mathrm{s}}$ is the error matrix and $\boldsymbol{U}_{\mathrm{s}} = [\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{u}_3]$ is the true signal eigenvectors. Finally, the root mean square error (RMSE) is computed as

$$\mathrm{RMSE}_{\mathrm{DPM}} = \left(\frac{1}{500}\sum_{i=1}^{500} \mathrm{SE}_{\mathrm{DPM}}(i)\right)^{1/2}. \tag{3.36}$$

[3]Since the eigendecomposition is unique up to a multiplication with a unity-magnitude complex scalar, the method introduced in [FW98, Equation (54)] is used to compute this scalar and correct the estimated eigenvectors.

The RMSE which is obtained from the analytical expression for the DPM algorithm, denoted as $\mathrm{ARMSE}_{\mathrm{DPM}}$, is computed using

$$\mathrm{ARMSE}_{\mathrm{DPM}} = \left(\sum_{l=1}^{3} \mathrm{tr}\left(\mathbb{E}\big(\delta\check{\boldsymbol{u}}_l(P)\, \delta\check{\boldsymbol{u}}_l^H(P) \big) \right) / \mathrm{tr}\big(\boldsymbol{U}_{\mathrm{s}} \boldsymbol{U}_{\mathrm{s}}^H \big) \right)^{1/2}, \tag{3.37}$$

where $\mathbb{E}\big(\delta\check{\boldsymbol{u}}_l(P)\, \delta\check{\boldsymbol{u}}_l^H(P)\big)$ is given in Theorem 3. In Fig. 3.3, Fig. 3.4 and Fig. 3.5, the RMSE of the conventional (centralized) PM averaged over 500 realizations is also shown.

In Fig. 3.3, $\mathrm{RMSE}_{\mathrm{DPM}}$ from 500 realizations is compared with the $\mathrm{ARMSE}_{\mathrm{DPM}}$ at different SNR values where the number of samples is fixed to $N = 100$ and the number of AC iterations P is taken to be 10, 20 and 30. The number of PM iterations is fixed to $Q = 10$ for all simulations. It can be seen from Fig. 3.3 that the error in the estimated eigenvectors decreases with increasing SNR until it reaches a certain value, which depends on P, then it is saturated. This saturation effect corresponds to the second terms on the right-hand-side of (3.21) and (3.22). Note that the error computed using the analytical expression $\mathrm{ARMSE}_{\mathrm{DPM}}$ corresponds well to the empirical performance $\mathrm{RMSE}_{\mathrm{DPM}}$.

In Fig. 3.4, the SNR is set to 10 dB and both $\mathrm{RMSE}_{\mathrm{DPM}}$ and $\mathrm{ARMSE}_{\mathrm{DPM}}$ are computed for different numbers of samples N for three different numbers of AC iterations, which are 10, 20 and 30. The number of PM iterations is fixed to $Q = 10$. From Fig. 3.4, it can be observed that the error in the estimated eigenvectors decreases with N for small values of N. However when N is large, $\mathrm{RMSE}_{\mathrm{DPM}}$ and $\mathrm{ARMSE}_{\mathrm{DPM}}$ do not change with N as it can be seen in Fig. 3.4 for $P = 10$ and $P = 20$. For $P = 30$, the values of $\mathrm{RMSE}_{\mathrm{DPM}}$ and $\mathrm{ARMSE}_{\mathrm{DPM}}$ are saturated at very large values of N (which is not displayed in the figure). Moreover, a larger number of AC iterations results in a smaller error. This behaviour of the $\mathrm{RMSE}_{\mathrm{DPM}}$ is in accordance with the conclusion that the DPM is a consistent estimator of the eigenvectors of the true measurement covariance matrix only when P is infinitely large, see Section 3.3.1. It can also be observed in Fig. 3.4 that the error computed using the analytical expressions $\mathrm{ARMSE}_{\mathrm{DPM}}$ corresponds to the values of $\mathrm{RMSE}_{\mathrm{DPM}}$.

In Fig. 3.5, the $\mathrm{RMSE}_{\mathrm{DPM}}$ is shown as a function of SNR for different values of Q, where the number of AC iterations is set to $P = 30$ and the number of samples is $N = 100$. Note that the empirical results $\mathrm{RMSE}_{\mathrm{DPM}}$ converge fast to the analytical expression $\mathrm{ARMSE}_{\mathrm{DPM}}$ with Q. In Fig. 3.5, for $Q = 10$, observe that the values of $\mathrm{ARMSE}_{\mathrm{DPM}}$ corresponds well to those of $\mathrm{RMSE}_{\mathrm{DPM}}$.

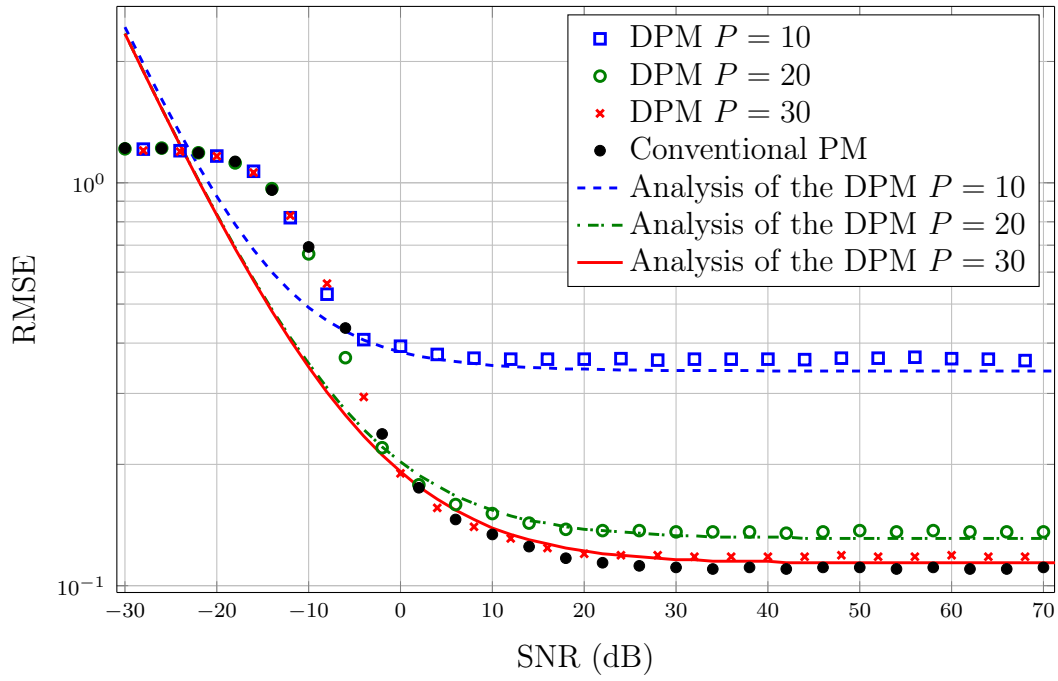

Fig. 3.3. The performance of eigendecomposition using the DPM as a function of SNR for a fixed number of samples $N = 100$.

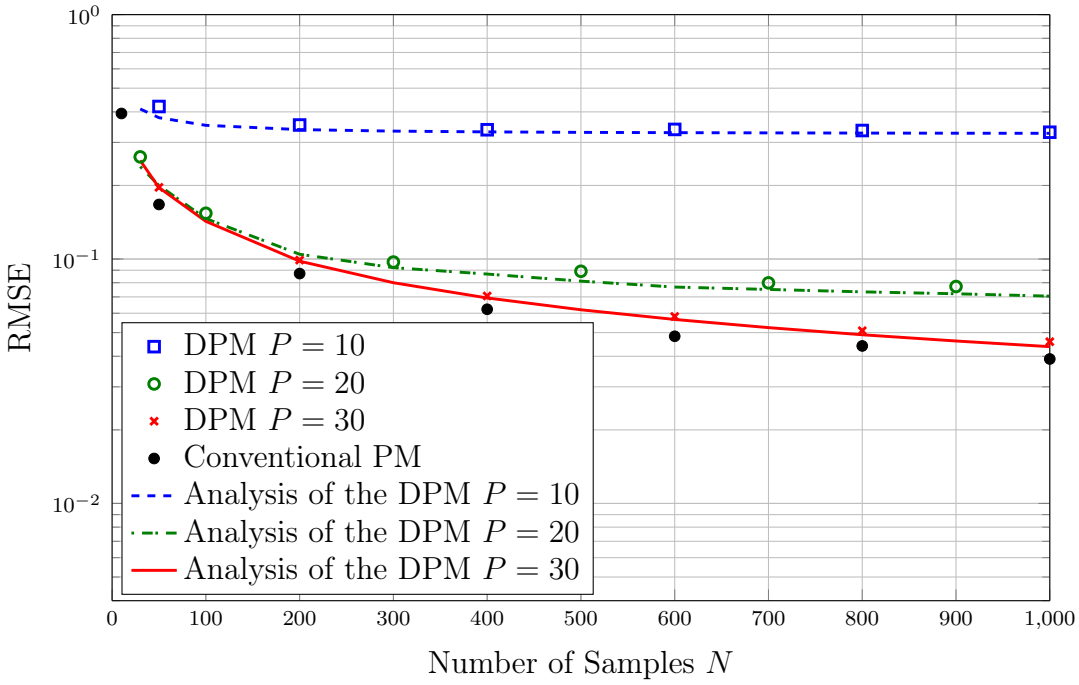

Fig. 3.4. The performance of eigendecomposition using the DPM as a function of N for a fixed SNR = 10 dB.

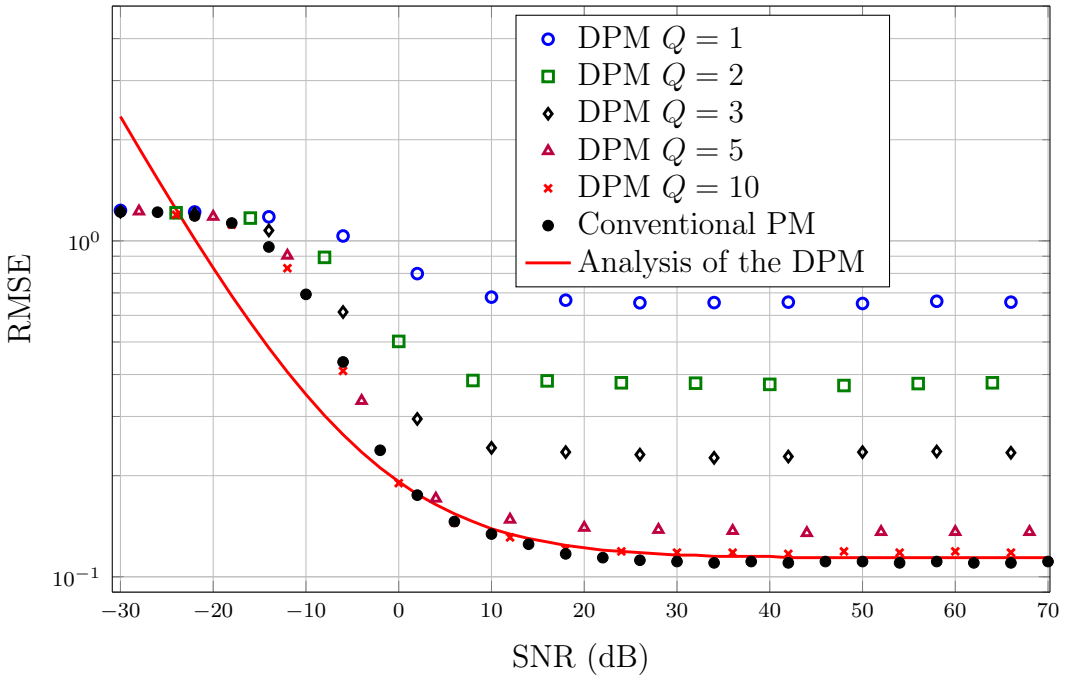

Fig. 3.5. The performance of eigendecomposition using the DPM as a function of SNR for a fixed number of AC iterations $P = 30$ and for a fixed $N = 100$ samples.

Performance of the DESPRIT Algorithm

In the second set of simulations, whose results are shown in Fig. 3.6 and Fig. 3.7, the performance of the DESPRIT algorithm is evaluated and compared to the analytical expressions of Section 3.3.2. The RMSE of the DESPRIT algorithm is computed over 500 realizations as

$$\mathrm{RMSE}_{\mathrm{DESPRIT}} = \left(\frac{1}{500} \sum_{i=1}^{500} \frac{1}{3} \sum_{l=1}^{3} (\check{\theta}_l(i) - \theta_l)^2 \right)^{\frac{1}{2}}, \tag{3.38}$$

where $\check{\theta}_l(i)$ is the estimate of θ_l computed at the ith realization using the DESPRIT algorithm. The analytical expression of the RMSE of the DESPRIT algorithm is

$$\mathrm{ARMSE}_{\mathrm{DESPRIT}} = \left(\frac{1}{3} \sum_{l=1}^{3} \mathbb{E}\big((\delta\check{\theta}_l)^2\big) \right)^{\frac{1}{2}}, \tag{3.39}$$

where $\mathbb{E}\big((\delta\check{\theta}_l)^2\big)$ is computed using Equations (3.26)–(3.30). In this set of simulations, the number of PM iterations is set to $Q = 10$. Moreover, in these simulations, the RMSE of the conventional ESPRIT algorithm is plotted along with its theoretical asymptotic approximations computed as in [RH89a] and the CRB for the conventional partly calibrated arrays [SG04].

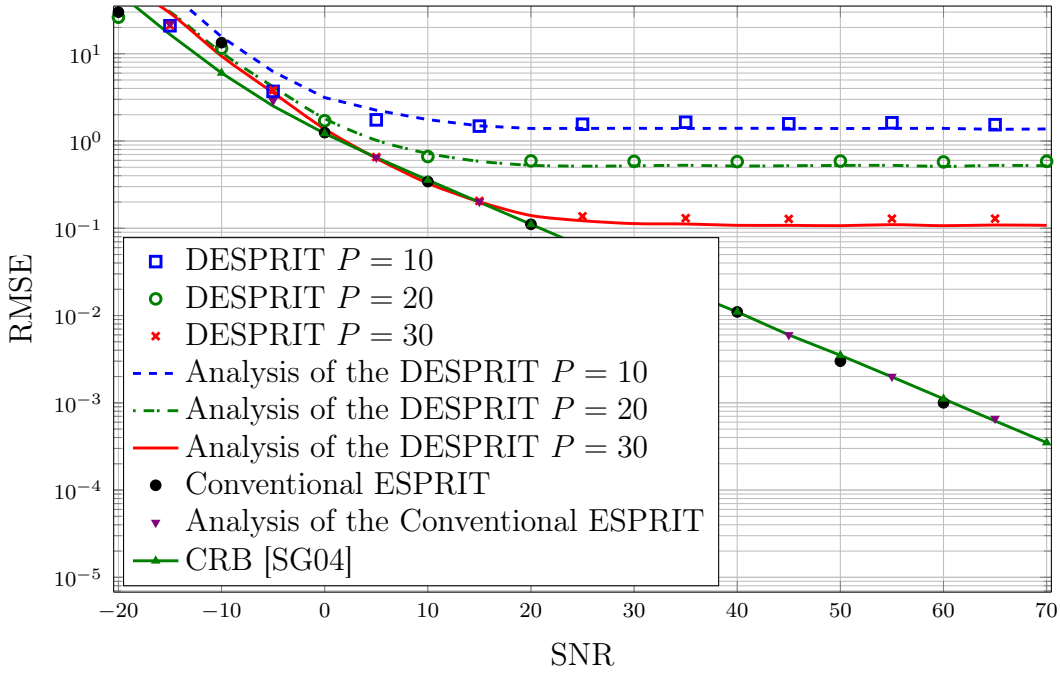

Fig. 3.6. The performance of DOA estimation using the DESPRIT algorithm as a function of SNR for a fixed number of samples $N = 100$.

Fig. 3.6 displays the values of $\text{ARMSE}_{\text{DESPRIT}}$ and $\text{RMSE}_{\text{DESPRIT}}$ for different values of the SNR, where a fixed number of samples $N = 100$ is assumed. Note that at low SNRs the performance of the DESPRIT algorithm is similar to that of the conventional ESPRIT algorithm and it improves with increasing SNR values. However, at high SNR, it can be observed that the performance of the DESPRIT algorithm deviates from that of the conventional ESPRIT algorithm. It is clear from Fig. 3.6 that this deviation depends on the number of AC iterations P. Thus, for $P = 30$ and SNR values up to SNR $= 15$ dB the performance of the DESPRIT algorithm is similar to that of the conventional ESPRIT algorithm and both achieve the conventional CRB, whereas for $P = 10$ the curves deviate for SNR values larger than 0 dB. Moreover, it can be seen from Fig. 3.6 that the RMSE of the DESPRIT algorithm at high SNRs is saturated and cannot be decreased unless the number of AC iterations is increased.

In Fig. 3.7, the SNR is fixed to 10 dB and $\text{ARMSE}_{\text{DESPRIT}}$ and $\text{RMSE}_{\text{DESPRIT}}$ are computed for different number of samples N. It is obvious that the error in the DESPRIT algorithm does not approach zero when $N \to \infty$, which is in accordance with the conclusion in Section 3.3.2, that the DESPRIT algorithm is not a consistent estimator of the DOAs, unless the number of AC iterations P is infinitely large.

In Fig. 3.6 and Fig. 3.7, it can be observed that the values obtained for the averaged RMSE of DESPRIT algorithm correspond well to the results of the analytical

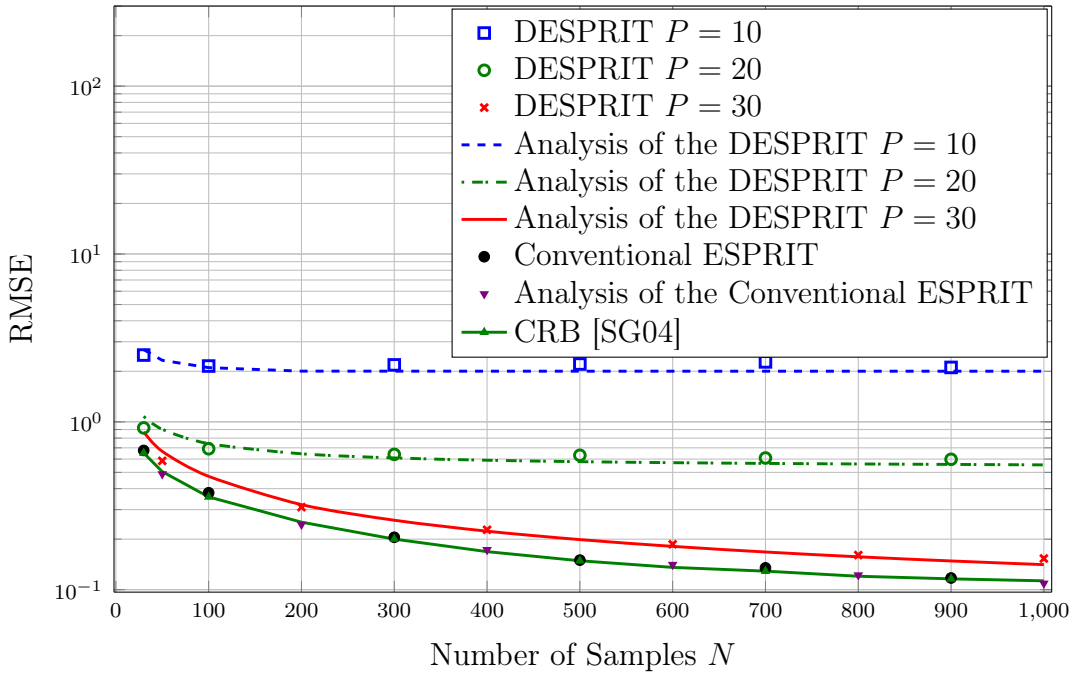

Fig. 3.7. The performance of DOA estimation using the DESPRIT algorithm as a function of N for a fixed SNR = 10 dB.

expression. The saturation effect which appears in the performance of the DESPRIT algorithm (for finite number of AC iterations P) corresponds to the second terms on the right-hand-side of (3.29) and (3.30).

3.4.2 The IDESPRIT Algorithm

In this section, the performance of the IDESPRIT algorithm is considered. Two simulation scenarios with $K = 6$ subarrays each consists of two sensors are considered. In the first scenario, the location of the sensors is chosen such that the array exhibits the shift-invariant property. Then, the location of the second sensors of each subarray is slightly perturbed to introduce small deviations around the nominal (shift-invariant) locations. The locations of the sensors of the 6 subarrays measured in half-wavelength are $\{(0,0), (1.01,0.03\}$, $\{(0.45,0.99), (1.44,1\}$, $\{(3.02,0.45), (4.04,0.44\}$, $\{(5.61,0.90), (6.60,0.91\}$, $\{(8.03,1.46), (9.00,1.45\}$, and $\{(8.70,0.50, (9.68,0.51\}$. In the second scenario, the location of the second sensor in all subarrays is selected arbitrarily. The locations of the sensors of the 6 subarrays measured in half-wavelength in this scenario are $\{(0,0), (1.51,0.13)\}$, $\{(0.45,0.99), (1.24,1)\}$, $\{(3.02,0.45), (4.00,0.84)\}$, $\{(5.61,0.90), (6.01,0.91)\}$, $\{(8.03,1.46), (9.20,1.85)\}$, and $\{(8.70,0.50), (9.98,0.01)\}$. In both scenarios, the neighboring subarrays and the AC weighting matrix are defined

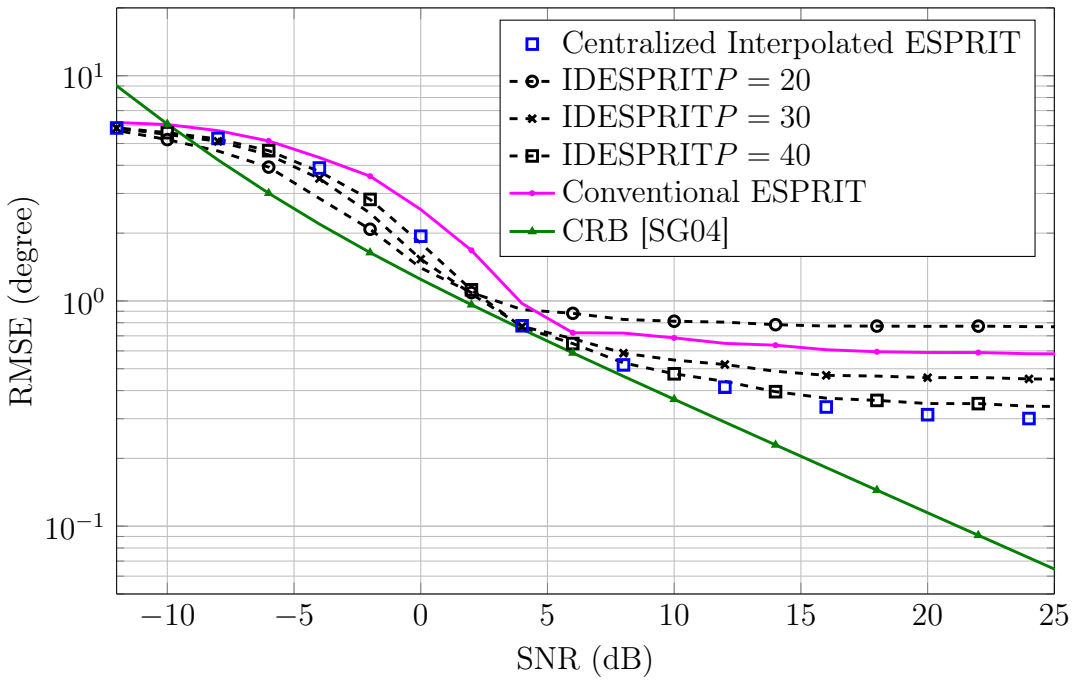

Fig. 3.8. RMSE as a function of SNR for Scenario 1 where $N = 100$.

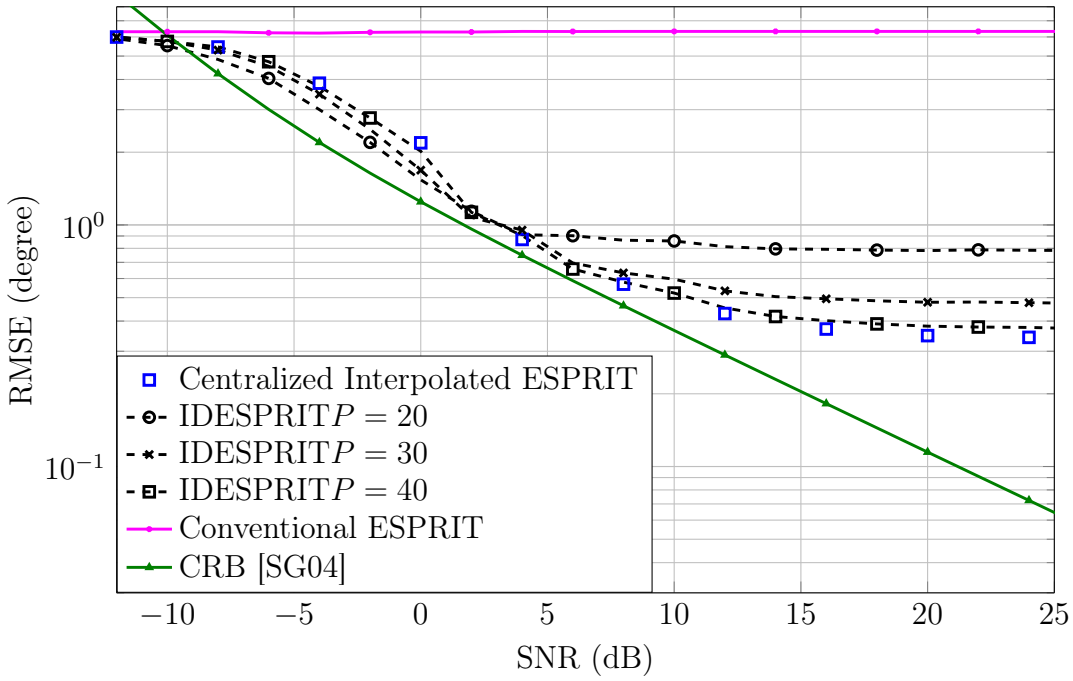

Fig. 3.9. RMSE as a function of SNR for Scenario 2 where $N = 100$.

as in Section 3.4.1. Further, in both scenarios, signals from $L = 2$ far-field narrow-band sources impinge onto the subarrays from directions $-14°$ and $-10°$. The RMSE of the IDESPRIT algorithm is computed over 500 realizations for 3 different numbers of AC iterations, $P = 20$, 30, and 40, in both scenarios. The following parameters are selected for the IDESPRIT algorithm: $d = 1$, $Q = 10$, and the sampling step size is $0.1°$ in the grid $\tilde{\boldsymbol{\theta}}$. For benchmarking, the empirical performance of the conventional ESPRIT and the centralized interpolated ESPRIT algorithms, i.e., the IDESPRIT for $P \to \infty$, and the CRB [SG04] are plotted.

Scenario 1: Shift-Invariant Array with Perturbations

The RMSEs of the above mentioned algorithms are demonstrated in Fig. 3.8 as a function of the SNR where the number of snapshots is fixed to $N = 100$. It can be observed in this figure that the conventional ESPRIT algorithm does not achieve the CRB due to the perturbation in the subarray structure. The centralized interpolated ESPRIT algorithm achieves the CRB for moderate SNR (in Fig. 3.8 for SNR ≤ 8 dBs) since it does not assume a shift-invariant array. The proposed IDESPRIT algorithm achieves a performance comparable to the conventional ESPRIT for $P = 20$ and achieves a better performance than the conventional ESPRIT for $P = 30$ and 40. The performance of the centralized interpolated ESPRIT algorithm does not improve at high SNR (in Fig 3.8 for SNR > 20 dB) because of the errors resulting from the interpolation [WG91]. Observe that at high SNR the performance of the IDESPRIT algorithm does not improve with SNR, due to the errors introduced by the finite number of AC iterations and the interpolation errors. This behaviour is noticeable in the DESPRIT algorithm and has been analyzed in Section 3.3.2. However, in the DESPRIT algorithm, the finite number of AC iteration is the only cause of this behaviour. Interestingly, the performance of the IDESPRIT for $P = 40$ is similar to that of the centralized interpolated ESPRIT algorithm in Fig. 3.8. That is because for $P = 40$ the AC errors is smaller than that of the interpolation.

Scenario 2: Arbitrary Array

The simulation results for the second scenario are demonstrated in Fig. 3.9. It can be observed that the RMSE of the conventional ESPRIT algorithm is very high, since in this scenario the conventional ESPRIT algorithm is not able to resolve the two sources. However, the centralized interpolated ESPRIT and the IDESPRIT algorithms achieve performance similar to Scenario 1, since they both do not rely on the assumption of shift-invariant arrays.

3.4.3 The Decentralized Root-MUSIC Algorithm

In this section using a ULA, first, the performance of the decentralized Root-MUSIC algorithm is verified by simulations and compared to the performance predicted by the expression in Section 3.3.3. Then, the performance of the decentralized Root-MUSIC algorithm which exploits the structure of the *fully calibrated* ULA is compared to that of the DESPRIT algorithm, where the ULA is considered to be *partly calibrated*.

A ULA composed of $K = 6$ subarrays, where each subarray consists of $M_k = 2$ sensors separated by half wavelength, is considered. The array topology of Fig. 3.2 is considered. For the AC protocol, the local weighting scheme of (3.34) is applied. Signals from $L = 3$ equal-powered sources impinge onto the ULA from directions $-15°, 0°$ and $4°$. In the simulations in Fig. 3.10 and Fig. 3.11, the RMSE of the decentralized Root-MUSIC algorithm averaged over 500 realizations and over L is compared with the corresponding results obtained from the analytical expression of Section 3.3.3, for different numbers of AC iterations $P = 10$, 20 and 30. Moreover, the RMSE of the conventional Root-MUSIC algorithm [Bar83] averaged over 500 realizations and the performance analysis of the conventional Root-MUSIC algorithm [RH89b] as compared with the CRB [SN90] are shown.

In Fig. 3.10, the number of snapshots is fixed to $N = 100$, and the aforementioned curves are plotted as a function of the SNR. Observe that the decentralized Root-MUSIC algorithm achieves the CRB with a performance similar to that of the conventional Root-MUSIC algorithm for low SNRs. However, for high SNRs (SNR > 20 dB, for $P = 30$ AC iterations) the performance of the decentralized Root-MUSIC algorithm saturates. Note that for larger number of AC iterations P a lower RMSE is achieved.

In Fig. 3.11, the SNR is fixed to 10 dB, and the aforementioned curves are plotted as a function of N. Note that the decentralized Root-MUSIC algorithm achieves the CRB with a performance similar to the conventional Root-MUSIC algorithm for small N. However, for large N ($N > 200$, for $P = 30$ AC iterations) the performance of the decentralized Root-MUSIC algorithm saturates. This observations are in perfect accordance with the conclusion made in Section 3.3.3, that the decentralized Root-MUSIC is a consistent estimator of the DOAs only when the number of AC iterations P becomes infinitely large. Observe that, in Fig. 3.10 and Fig. 3.11, the analytical expression from Section 3.3.3 is consistent with the RMSE averaged over 500 realizations.

In Fig. 3.12, the performance of the decentralized Root-MUSIC and the DESPRIT algorithms are plotted versus SNR for $N = 100$ samples and $P = 30$ AC iterations. For

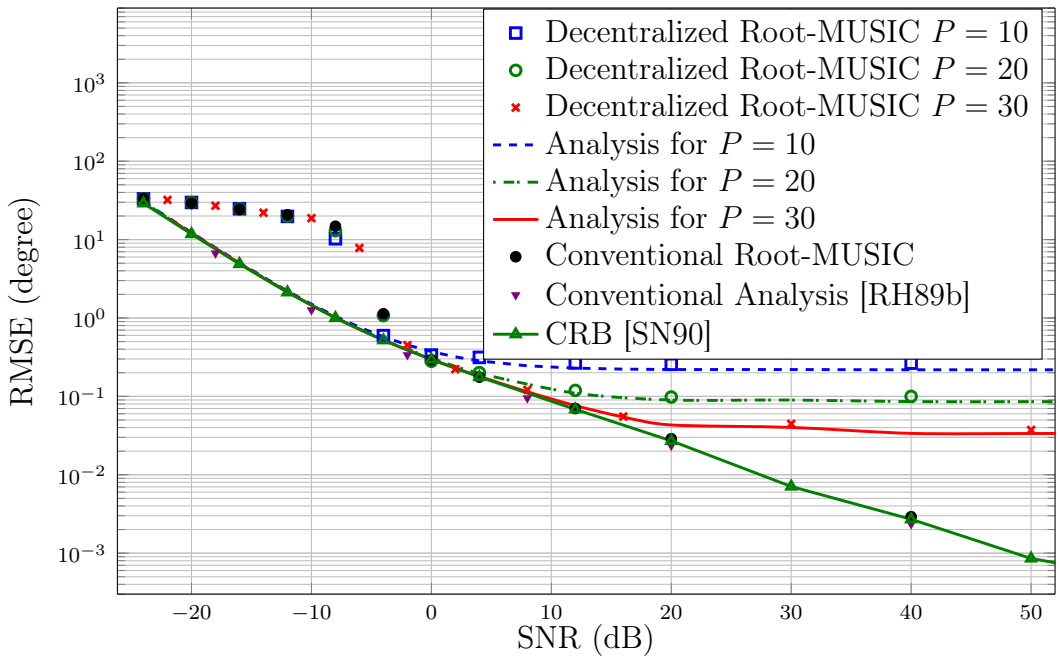

Fig. 3.10. The performance of the decentralized Root-MUSIC algorithm as a function of SNR for $N = 100$ snapshots.

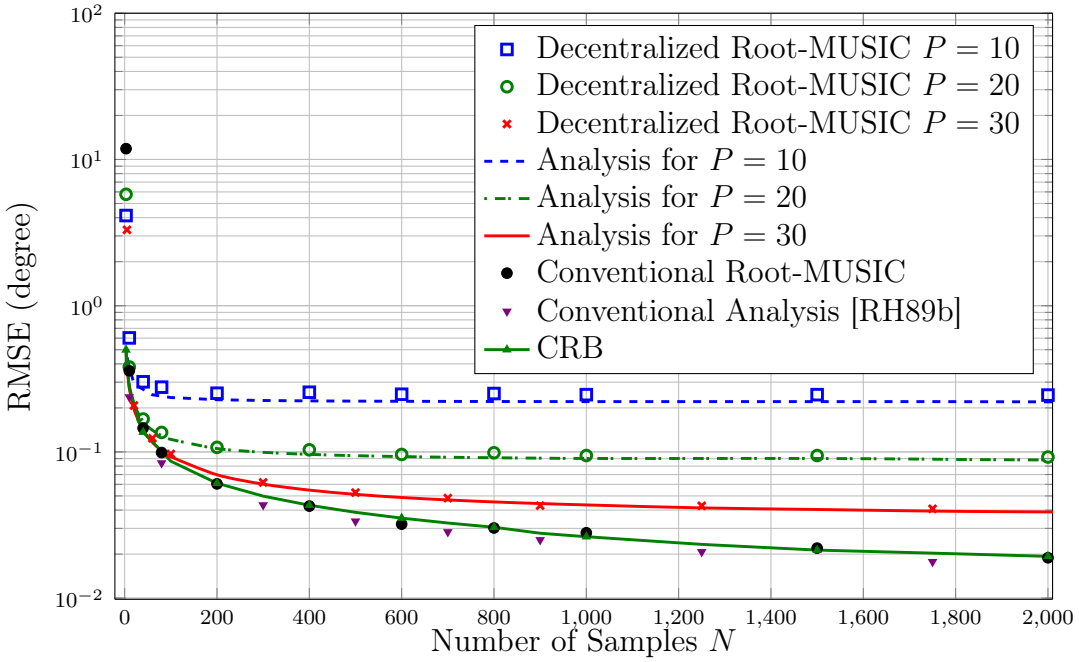

Fig. 3.11. The performance of the decentralized Root-MUSIC algorithm as a function of N for SNR = 10 dB.

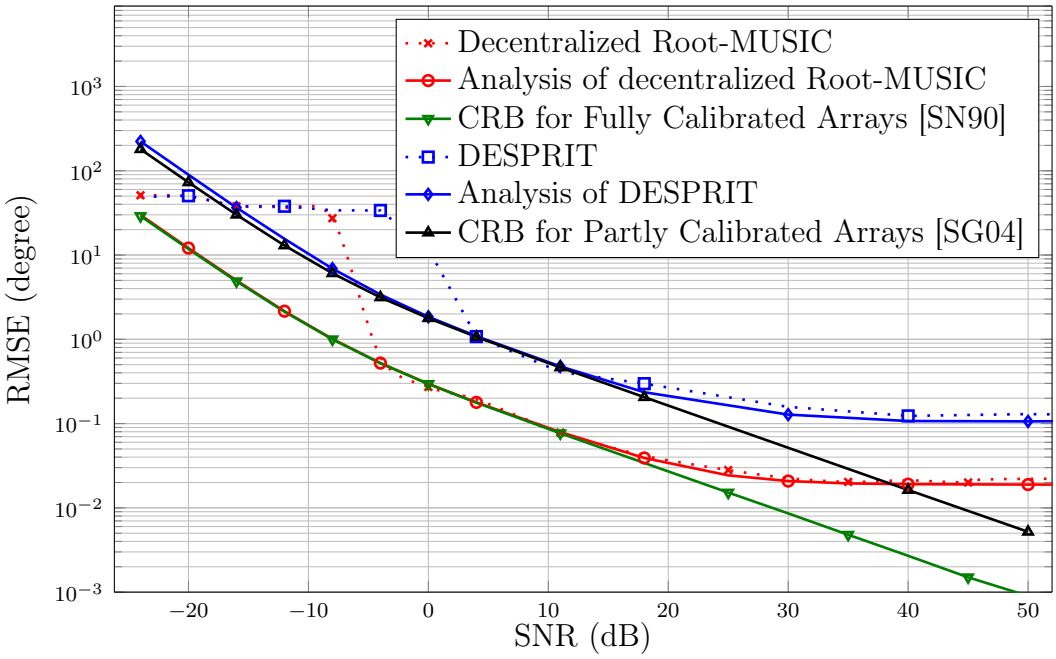

Fig. 3.12. The performance of the decentralized Root-MUSIC and DESPRIT algorithms as a function of SNR for $N = 100$ samples and AC iterations $P = 30$.

the DESPRIT algorithm, the upper group consists of the first sensor of each subarray, whereas the lower group consists of the second sensor of each subarray, thus, the upper and lower selection matrices are defined as $\overline{\boldsymbol{J}}_k = [1, 0]$ and $\underline{\boldsymbol{J}}_k = [0, 1]$, for $k = 1, \ldots, 6$, respectively. Note that the aforementioned definition of the upper and lower groups does not require the displacement between the subarrays to be known[4]. In other words, by this definition of the upper and lower groups, the DESPRIT algorithm uses less information about the array structure than the decentralized Root-MUSIC algorithm. Particularly, the array is assumed to be *fully calibrated* in the decentralized Root-MUSIC, while in the DESPRIT algorithm, the array is assumed to be *partly calibrated*. Moreover, in Fig. 3.12, the CRBs corresponding to fully and partly calibrated arrays, which are derived in [SN90] and [SG04], respectively, are shown. In Fig. 3.12, it can be observed that the CRB corresponding to partly calibrated array is above that corresponding to fully calibrated array, i.e., better DOA estimation performance can be achieved in the case of fully calibrated array. The performance of the DESPRIT algorithm in Fig. 3.12 corresponds to the CRB of partly calibrated array for SNR values

[4] Usually for a fully calibrated ULA, the upper and lower groups are defined as the first $M-1$ and last $M-1$ sensors, respectively, refer to [XSRK94]. This definition of the upper and lower groups exploits the structure of the ULA better than the definition introduced above. The DOA estimation performance obtained by this definition is superior to that obtained from the above definition. However, the above definition is considered in the simulations, since it takes into consideration the structure of the subarrays and the decentralized processing scheme. Whereas, the definition in [XSRK94] is used in a centralized processing scheme.

up to 20 dB. For higher SNR values, the performance of the DESPRIT algorithm is saturated because of the errors resulting from a finite number of AC iterations P. The performance of the decentralized Root-MUSIC algorithm corresponds to the CRB of fully calibrated array and shows the same saturation effect as the DESPRIT algorithm for high SNR values. Observe that by exploiting more information about the array structure, the decentralized Root-MUSIC algorithm achieves better performance than the DESPRIT algorithm.

3.5 Decentralized Source Enumeration

In this section, the problem of detecting one source using a system of multiple subarrays is considered. Based on the AC protocol, a decentralized implementation of the energy detector (ED), which is conventionally applied for spectrum sensing in a centralized fashion, is proposed. The exact (non-asymptotic) null distribution of the decentralized ED test statistic is derived and used to compute the test threshold. The communication overhead of the proposed detector is low compared to the existing decentralized source detecting algorithms. Moreover, the ED is extended to the case of multiple source to detect the number of sources impinging onto a system of subarrays.

3.5.1 The Single Source Case

In the single source case, the subarrays make a decision whether the signal $s(t)$ in (2.10), is present ($s(t) \neq 0$) or absent ($s(t) = 0$) in the measurements. This decision is formulated using the binary hypothesis testing framework as

$$\begin{aligned} \mathcal{H}_0 : &\quad \boldsymbol{x}(t) = \boldsymbol{n}(t), \\ \mathcal{H}_1 : &\quad \boldsymbol{x}(t) = \boldsymbol{a}(\theta_1)s(t) + \boldsymbol{n}(t), \end{aligned} \tag{3.40}$$

where $\mathcal{H}_0$ is the null hypothesis, $\mathcal{H}_1$ is the alternative hypothesis, θ_1 is the direction of the single source, and $t = 1, \ldots, N$. In centralized sensing architectures, the generalized likelihood ratio test (GLRT) and the energy detector (ED) can be applied, irrespectively of the knowledge of the response vector $\boldsymbol{a}(\theta_1)$ and the source power $p_1 = \mathbb{E}\big(s(t)s^*(t)\big)$. The GLRT [TNKG10] results in the test statistic

$$T_{\mathrm{GLRT}} = \hat{\lambda}_1/\sigma^2 \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \eta_{\mathrm{GLRT}}, \tag{3.41}$$

where $\hat{\lambda}_1$ is the largest eigenvalue of the sample covariance matrix $\hat{\boldsymbol{R}}$ in (2.15). For a given false alarm rate FAR γ, the test threshold η_{GLRT} is determined using the null

distribution of T_{GLRT} as $\mathcal{P}\big(T_{\text{GLRT}} > \eta_{\text{GLRT}}|\mathcal{H}_0\big) = \gamma$. The ED [CMB04, TS05] uses the decision metric

$$T_{\text{ED}} = \sum_{t=1}^{N} \boldsymbol{x}^H(t)\boldsymbol{x}(t) = \text{tr}\big(N\hat{\boldsymbol{R}}\big) \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \eta_{\text{ED}}, \tag{3.42}$$

where the threshold η_{ED} is determined as in the GLRT.

In [PS12], a decentralized implementation of the GLRT is presented. In this implementation, the principal eigenvalue and eigenvector of the sample covariance matrix $\hat{\boldsymbol{R}}$ are computed using the DPM as introduced in Section 2.4.3. The performance of the DPM is analyzed in Section 3.3.1 and it is proven that, under mild assumptions, the eigendecomposition of the sample covariance matrix $\hat{\boldsymbol{R}}$ using the DPM is equivalent to the centralized eigendecomposition of the matrix $\check{\boldsymbol{R}}$. Thus, when the DPM is used to compute the principal eigenvalue of $\hat{\boldsymbol{R}}$ two technical challenges arise. Firstly, according to Theorem 1 in Section 3.3.1, the distribution of the principal eigenvalue of $\hat{\boldsymbol{R}}$ computed using the DPM does not exactly correspond to that of the principal eigenvalue of the matrix $\hat{\boldsymbol{R}}$ [Mui09, p. 421], unless the number of AC iterations P used to compute the principal eigenvalue is large. An analytical expression for the exact null distribution of the principal eigenvalue of $\check{\boldsymbol{R}}$ is not available. Therefore, in practice, an asymptotic (in P, N, and M) approximation of this distribution is used. Secondly, according to [GVL13], the convergence rate of the conventional PM (and also that of the DPM) is proportional to the ratio $(\hat{\lambda}_1/\hat{\lambda}_2)$, where $\hat{\lambda}_1$ and $\hat{\lambda}_2$ are the largest and second largest eigenvalues of the matrix $\hat{\boldsymbol{R}}$, respectively, in (2.12). Hence, if the considered frequency band is free, i.e., the signal of the source is absent in the measurements, the matrix $\boldsymbol{R}$ has one eigenvalue of multiplicity M, since $\boldsymbol{R} = \sigma^2\boldsymbol{I}_M$. In this case, the DPM exhibits very slow convergence. To overcome these difficulties, in the following section, an alternative decentralized detection algorithm based on the ED, instead of the PM, is introduced.

3.5.2 The Decentralized Energy Detector

The test statistics of the ED in (3.42) can be partitioned as $T_{\text{ED}} = \sum_{k=1}^{K} T_{\text{ED},k}$, where the kth subarray computes a part of the test statistics T_{ED} as

$$T_{\text{ED},k} = \sum_{t=1}^{N} \boldsymbol{x}_k^H(t)\boldsymbol{x}_k(t), \tag{3.43}$$

for $k = 1, \ldots, K$. Similar to (2.64), in order to obtain the decentralized ED, the AC protocol is carried out for P iterations to compute T_{ED}, hence

$$\check{T}_{\mathrm{ED},[k]} = \left[K \boldsymbol{W}^P \begin{bmatrix} T_{\mathrm{ED},1} \\ \vdots \\ T_{\mathrm{ED},K} \end{bmatrix} \right]_k , \tag{3.44}$$

where $\boldsymbol{W}$ is the AC weighting matrix defined in (2.55) and P is the number of AC iterations used to compute this test statistic.

The threshold of the conventional test statistic T_{ED} is computed as follows. The null distribution of the random variable NT_{ED} is Chi-Square with $2NM$ degrees-of-freedom (DOF) [GS05]. Thus, for a given value γ, the threshold is computed using $\mathcal{P}\big(T_{\mathrm{ED}} > N\eta_{\mathrm{ED}}|\mathcal{H}_0\big) = \gamma$.

However, in the proposed decentralized implementation (3.44), each of the random variables $\check{T}_{\mathrm{ED},[1]}, \ldots, \check{T}_{\mathrm{ED},[K]}$ is a linear combination of statistically independent Chi-Square distributed random variables, see [Jon83,Box54]. The computation of the exact density function, the cumulative density function and its inverse is introduced in [RP49,Far84]. Approximations for this distribution using simpler distributions with closed form densities and cumulative functions are presented in [Box54, Woo89, SS77]. Note that for a sufficiently large number of AC iterations P, the random variables $\check{T}_{\mathrm{ED},[1]}, , \ldots, \check{T}_{\mathrm{ED},[K]}$ have the same distribution as the conventional test statistic T_{ED}, since the average in (3.44) is accurate when P is large. However, as it is demonstrated in the simulations in Section 3.5.4, for small P this approximation is no longer valid. Interestingly, using the scaled Chi-Square approximation in [Box54] for each of the random variables $\check{T}_{\mathrm{ED},[1]}, \ldots, \check{T}_{\mathrm{ED},[K]}$ is more accurate than using the aforementioned centralized Chi-Square with $2NM$ DOFs approximation[5]. The test threshold at the kth subarray, denoted as $\eta_{\mathrm{ED},k}$, is computed according to $\mathcal{P}\big(\check{T}_{\mathrm{ED},[k]} > N\eta_{\mathrm{ED},k}|\mathcal{H}_0\big) = \gamma$, for $k = 1, \ldots, K$.

Regarding the computational complexity and the associated signaling overhead of the detectors, observe that the computation of the decentralized ED only requires one instant of the AC protocol, while the computation of the principal eigenvalue of the matrix $\hat{\boldsymbol{R}}$ using the DPM requires more than NQ AC protocols, refer to Section 2.4.4.

[5]The approximation in [Box54] is based on matching the moments of the real distribution of linear combination of Chi-Square variables with the moments of a scaled Chi-Square variable. Whereas the centralized approximation does not take into consideration the real distribution and is only accurate when the number of AC iterations P is large.

3.5.3 Source Enumeration Using the Decentralized ED

The problem of source enumeration using passive arrays can be considered as a generalization of the aforementioned single source problem. However, instead of the binary hypotheses testing (used in the single source case) a multiple hypothesis testing framework is utilized in detecting the number of sources [WJ90, BZP02]. The test statistics of conventional source enumeration algorithms use all the eigenvalues of the sample covariance matrix [WJ90, BZP02, WK85b, LZ13], which is not affordable in decentralized implementations.

To overcome the difficulties of computing all the eigenvalues $\check{\lambda}_1, \ldots, \check{\lambda}_M$ using the DPM in the decentralized sensing architecture, the ED approach is extended to the detection in the multi-source scenario considered in this section as shown in Algorithm 3. In contrast to conventional source enumeration methods, in the proposed detection procedure, it is decided that a source signal is present before actually computing the corresponding eigenvalue. Thus, in Algorithm 3, the unnecessary computation of the noise eigenvalues is avoided. In Algorithm 3, it is assumed that the decentralized test statistic $T_{\mathrm{ED}}^{(n)}$ is available to all subarrays, thus, it is denoted as $T_{\mathrm{ED}}^{(n)}$ (not as $\check{T}_{\mathrm{ED},[k]}^{(n)}$). This can be achieved by carrying out the AC protocol in (3.44) for a large number of AC iterations. Moreover, it is assumed that all subarrays computes the same eigenvalues, although the computation of the eigenvalues is carried out in a decentralized fashion using the DPM. This can be achieved by running the AC protocol used to compute the eigenvalues for a sufficiently large number of iterations. Consequently, the test threshold computed at each iteration of Algorithm 3 is the same for all subarrays and, thus, the sub-index of the subarray is dropped from the threshold and test statistics in Algorithm 3.

Algorithm 3 Detecting the number of sources

Step 0: Set $n = 0$, compute $T_{\mathrm{ED}}^{(0)} = \mathrm{tr}\big(\hat{\boldsymbol{R}}\big)$, and compute the initial test threshold $\eta_{\mathrm{ED}}^{(0)}$.
while ($T_{\mathrm{ED}}^{(n)} > \eta_{\mathrm{ED}}^{(n)}$ and $n < M$) **do**
  Step 1: Set $n = n + 1$.
  Step 2: Compute the nth eigenvalue $\check{\lambda}_n$ of the matrix $\hat{\boldsymbol{R}}$ using the DPM.
  Step 3: Set $T_{\mathrm{ED}}^{(n)} = T_{\mathrm{ED}}^{(n-1)} - \check{\lambda}_n$.
  Step 4: Compute the new test threshold $\eta_{\mathrm{ED}}^{(n)}$.
end while
return the number of detected sources n.

In contrast to the initial test statistic $T_{\mathrm{ED}}^{(0)}$, the null distribution of the test statistics $T_{\mathrm{ED}}^{(n)}$, for $n = 1, \ldots, M$, is difficult to derive. Thus, a simple approximation of the null

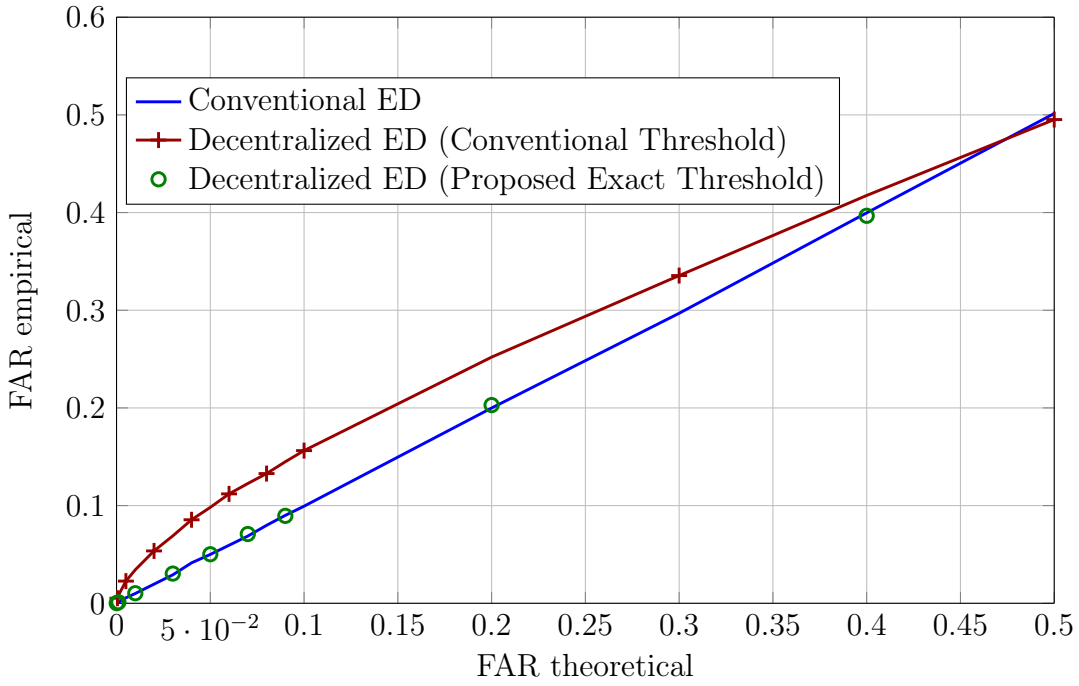

Fig. 3.13. The empirical FAR as function of the theoretical FAR.

distribution of $T_{\mathrm{ED}}^{(n)}$ using the Chi-Square distribution is sought. Note that, the test statistic $T_{\mathrm{ED}}^{(n)}$ can be written as $T_{\mathrm{ED}}^{(n)} = \mathrm{tr}(\hat{\boldsymbol{R}}) - \sum_{i=1}^{n} \check{\lambda}_i = \sum_{i=n+1}^{M} \check{\lambda}_i$, since the trace of a matrix is equal to the sum of its eigenvalues. Thus, the null distribution of $T_{\mathrm{ED}}^{(n)}$ can be approximated with that of the trace of covariance matrix with $(M-n)$ noise eigenvalues which we take to be full rank, i.e., of size $(M-n) \times (M-n)$. Thus, the distribution of $NT_{\mathrm{ED}}^{(n)}$ is approximated by a Chi-Square with $2N(M-n)$ DOF, which is used to compute the threshold $\eta_{\mathrm{ED}}^{(n)}$ in Step 4.

3.5.4 Simulation Results

An array of $K = 6$ subarrays each equipped with two sensors is considered. The array configuration and the AC weighting scheme are that of Section 3.4.1. The noise variance is set to $\sigma^2 = 1$, and the number of PM iterations is $Q = 10$. The empirical results in the following sections are averaged over 10^5 realizations.

The Single Source Case

In the simulations of this section, the number of AC iteration is $P = 4$ and the number of samples is $N = 30$. In Fig. 3.13, the case of unoccupied spectrum resources is

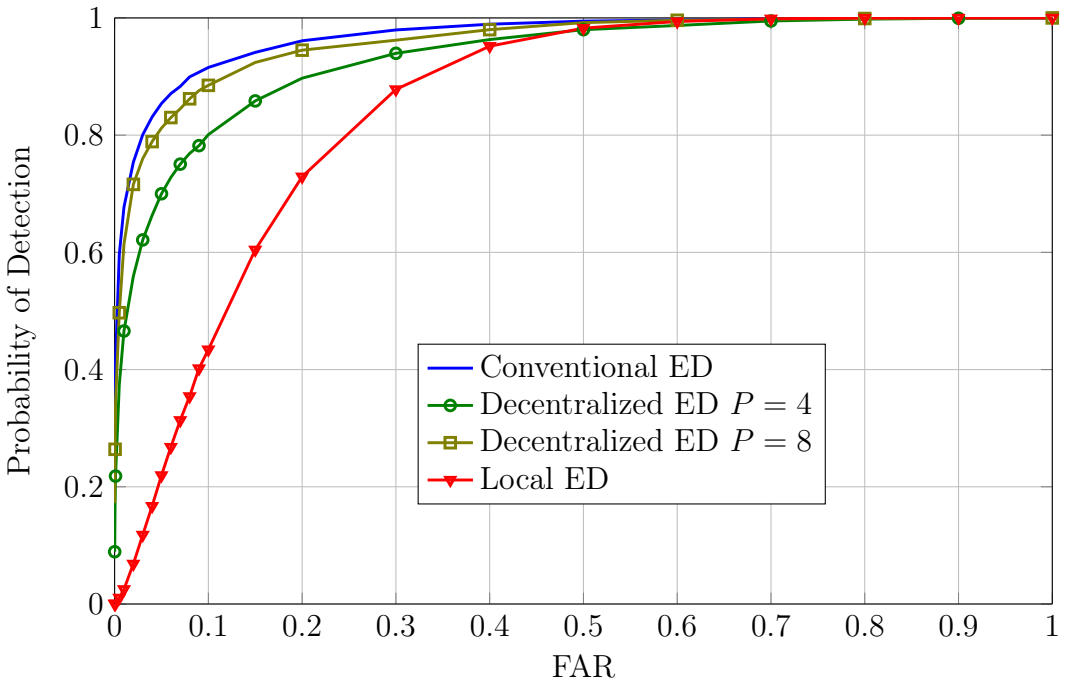

Fig. 3.14. The probability of detection as function of the theoretical FAR.

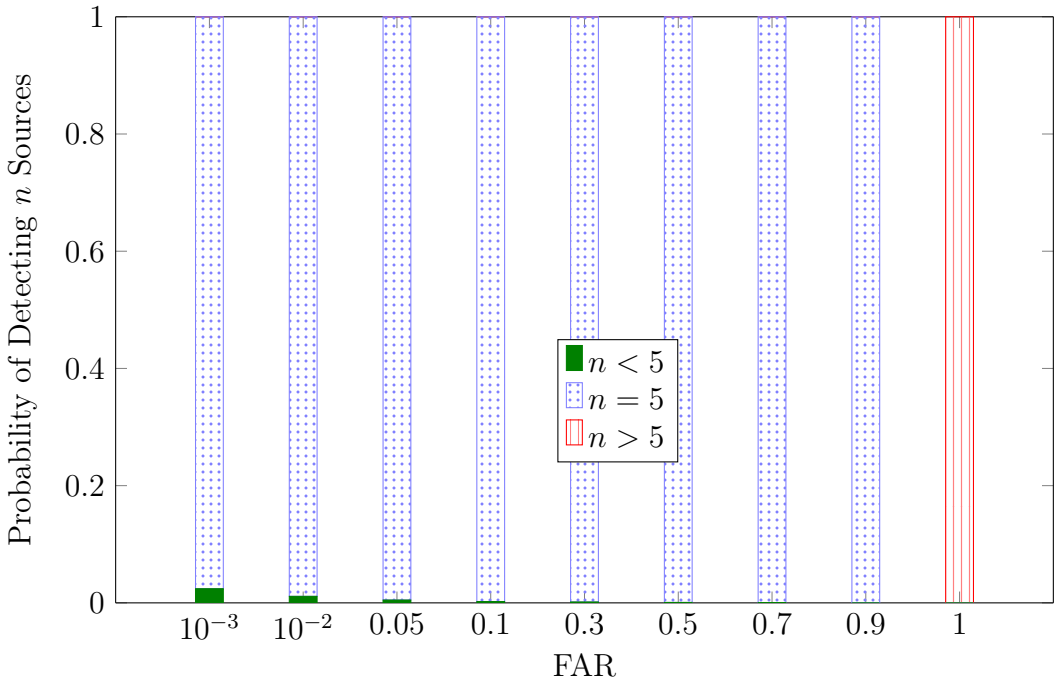

Fig. 3.15. The probability of deciding for n sources as function of the FAR.

considered. The empirical FAR is plotted as a function of the theoretical FAR for the conventional ED, the decentralized ED using the proposed exact threshold, and the decentralized ED with the threshold computed using the centralized approximation. It can be observed in Fig. 3.13 that the centralized approximation is not able to maintain the theoretical FAR. However, using the proposed exact computation the theoretical FAR is maintained.

In Fig. 3.14, the case of one source with SNR$=-10$ dB occupying the channel is considered. The probability of correctly detecting the source is plotted versus varying FAR for the centralized ED, the proposed decentralized ED with $P = 4$ and $P = 8$ AC iterations, and the ED for one (not cooperating) subarray. Note that the decentralized ED achieves better detection performance than the local ED of one subarray and that the decentralized ED performance quickly converges to that of the centralized case for increasing number of AC iterations P.

Detecting the Number of Sources

In the simulations of this section, it is assumed that the number of AC iteration is $P = 15$, the number of samples is $N = 100$, and 5 sources with SNR of 0 dB impinge onto the subarrays from directions $-10°$, $-5°$, $10°$, $25°$, and $40°$. In Fig. 3.15, the probability of detecting the correct number of sources $n = 5$, (miss-detection) $n < 5$, and (false alarm) $n > 5$ in the proposed decentralized ED is plotted for varying FAR. It can be observed in Fig. 3.15 that the decentralized ED achieves a correct detection probability of 0.97 for a FAR $\gamma = 10^{-3}$ and larger probabilities for increasing FAR. For $\gamma = 1$ the decentralized ED decides for the maximum number of sources as expected.

3.6 Summary

In this chapter, decentralized DOA estimation using the DESPRIT algorithm has been introduced. The DESPRIT algorithm is applicable for partly calibrated arrays and it can identify more sources than each subarray individually can. Unlike the conventional algorithms, the DESPRIT algorithm scales well with increasing number of subarrays. Similar to the conventional ESPRIT algorithm, the DESPRIT algorithm is robust against imperfections in the synchronization between different subarrays. Since the DESPRIT algorithm requires the array structure to be shift-invariant, using interpolation, a generalization of the DESPRIT algorithm to arbitrary array geometries has

been introduced. For fully calibrated ULAs, a decentralized DOA estimation algorithm which exploits the array structure has been introduced based on the Root-MUSIC algorithm. Similar to the conventional DOA estimation algorithms, the DESPRIT and decentralized Root-MUSIC algorithms require the number of sources to be available beforehand. The problem of source number detection has been considered and a decentralized algorithm based on the ED has been proposed which decides that a source is present before computing the eigenvalue corresponding to it.

The decentralized algorithms introduced in this chapter require the eigendecomposition of the sample covariance matrix which is performed using the DPM in Section 2.4.3. The DPM requires multiple PM iterations to converge for each eigenvector which results in a large communication cost. Moreover, the DPM performs a batch-based eigendecomposition i.e., in the DPM, first the individual subarrays collect and store measurements, then, they perform decentralized eigendecomposition which requires computational power and communication between the subarrays. This results in an unbalanced usage of subarray resources (memory, bandwidth and processing power), which can be avoided by online processing. In the next chapter two decentralized eigendecomposition algorithms are introduced which mitigate these drawbacks of the DPM.

Chapter 4

Coherent Decentralized Eigendecomposition

In the previous chapter, the decentralized DOA estimation and source enumeration are performed based on the DPM for eigendecomposition of the sample covariance matrix. For each eigenvector, the DPM requires a sufficiently large number of PM iterations to converge, which results in a large communication cost. Moreover, the DPM is a batch processing algorithm, i.e., in the DPM, first the individual subarrays collect and store measurements, then, they perform decentralized estimation assuming that the sources are stationary in the batch of the collected samples. However, if the sources are moving the DOAs of the signals of sources must be updated online, i.e., at each sample. In this chapter, the following two algorithms are proposed:

1. The decentralized Lanczos algorithm for eigendecomposition is introduced to mitigate the communication cost required for computing the signal eigenvectors. A low cost decentralized implementation of the ESPRIT algorithm based on the decentralized Lanczos method is introduced.
2. An online generalized eigendecomposition algorithm is proposed and used for online DOA estimation, also referred to as DOA tracking.

Although applied for DOA estimation, the decentralized eigendecomposition algorithms introduced in this chapter are general and can be used for other applications, e.g., spectrum sensing, source enumeration, and linear discriminant analysis to name a few. This chapter is based on the publications in [SPPZ14, SPZ15a].

4.1 The Decentralized Lanczos Method

As explained in Section 2.4.3, the DPM requires Q PM iterations for each computed eigenvector $\check{\boldsymbol{u}}_l$, for $l = 1, \ldots, M$. Where in each iteration the scalars $\{\check{x}^{(q)}_{t,l,[k]}\}_{t=1}^{N}$ in (2.64) are computed. This is the most expensive step of the DPM, refer to Section 2.4.4. To reduce the communication cost without compromising the performance of the DOA estimation algorithms proposed in the previous chapter, we introduce the decentralized Lanczos method for estimating the signal eigenvectors based on the AC protocol. The problem of spurious eigenvalues that usually arises in the Lanczos method, due

to numerical errors [GVL13], is addressed. A decentralized implementation of Lanczos method is presented in [PS12] whose main focus lies on the computation of the eigenvalues of the sample covariance matrix. However, in [PS12], neither the computation of the eigenvectors nor the problem of the spurious eigenvalues are addressed, which are two essential issues in the distributed implementation of the Lanczos method. In this section, these two aspects are considered.

4.1.1 The Conventional Lanczos Method

Algorithm 4 The Conventional Lanczos Method

Step 1: $\hat{\alpha}_1 \leftarrow 1, \breve{\boldsymbol{u}}_0 \leftarrow \boldsymbol{0}, \breve{\boldsymbol{u}}_1 \leftarrow M \times 1$ random vector
for $j = 1, 2, \ldots, \breve{L}$ **do**
  Step 2: $(AC1)$ $\hat{\boldsymbol{u}}_j \leftarrow \hat{\boldsymbol{R}}\breve{\boldsymbol{u}}_j$ (4.1)
  Step 3: $(AC2)$ $\breve{\alpha}_j \leftarrow \breve{\boldsymbol{u}}_j^H \hat{\boldsymbol{u}}_j$
  if $j < \breve{L}$ **then**
    Step 4: (Local) $\hat{\boldsymbol{u}}_j \leftarrow \hat{\boldsymbol{u}}_j - \breve{\alpha}_j \breve{\boldsymbol{u}}_j - \hat{\alpha}_j \breve{\boldsymbol{u}}_{j-1}$
    Step 5: $(AC3)$ $\hat{\alpha}_{j+1} \leftarrow \|\hat{\boldsymbol{u}}_j\|_2$
    Step 6: (Local) $\breve{\boldsymbol{u}}_{j+1} \leftarrow \hat{\boldsymbol{u}}_j / \hat{\alpha}_{j+1}$
  end if
end for
Step 7: (Local) $\breve{\boldsymbol{T}} \leftarrow$

$$\begin{pmatrix} \breve{\alpha}_1 & \hat{\alpha}_2 & & & \\ \hat{\alpha}_2 & \breve{\alpha}_2 & \hat{\alpha}_3 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & \hat{\alpha}_{\breve{L}} \\ & & & \hat{\alpha}_{\breve{L}} & \breve{\alpha}_{\breve{L}} \end{pmatrix} \quad (4.2)$$

Step 8: (Local) Perform eigendecomposition of $\breve{\boldsymbol{T}}$, denote the eigenvalues and the eigenvectors of $\breve{\boldsymbol{T}}$ as $\breve{t}_1 > \ldots > \breve{t}_{\breve{L}}$ and $\breve{\boldsymbol{t}}_1, \ldots, \breve{\boldsymbol{t}}_{\breve{L}}$, respectively.
Step 9: (Local) Compute the first $\breve{L}$ eigenvectors of $\hat{\boldsymbol{R}}$ from $\hat{\boldsymbol{u}}_i = \breve{\boldsymbol{U}}\breve{\boldsymbol{t}}_i, i = 1, \ldots, \breve{L}$, where $\breve{\boldsymbol{U}} = [\breve{\boldsymbol{u}}_1, \ldots, \breve{\boldsymbol{u}}_{\breve{L}}]$.

The conventional Lanczos method [GVL13, p.549], summarized in Algorithm 4, is used to iteratively calculate the eigenvalues and eigenvectors of Hermitian matrices. In this subsection, the Lanczos method is introduced for computing the eigenvalues and eigenvectors of the sample covariance matrix $\hat{\boldsymbol{R}}$ in a centralized fashion. The decentralized implementation of the Lanczos method is considered in the next subsection, where the steps 2, 3, and 5, labeled as AC1, AC2 and AC3, are carried out using the AC protocol. The iteration of Lanczos method starts with a norm-one random vector $\breve{\boldsymbol{u}}_1$, and in the jth iteration, the vector $\breve{\boldsymbol{u}}_{j+1}$ and the two scalars $\breve{\alpha}_j$ and $\hat{\alpha}_{j+1}$ are calculated (refer to Algorithm 4). A total of $\breve{L}$ iterations are carried out, where a larger iteration number $\breve{L}$ results in more accurate subspace estimates. However, as shown in the simulation

results, using $\breve{L}$ slightly larger than L, is generally sufficient in practice. After $\breve{L}$ iterations, the tridiagonal matrix $\breve{\boldsymbol{T}}$ in (4.2) is constructed from $\breve{\alpha}_j$ and $\hat{\alpha}_{j+1}$. The matrices $\breve{\boldsymbol{T}}$ and $\hat{\boldsymbol{R}}$ share the same $\breve{L}$ largest eigenvalues [GVL13, p.549], and the corresponding eigenvectors of $\hat{\boldsymbol{R}}$ are computed as shown in Step 9 of Algorithm 4.

4.1.2 The Decentralized Lanczos Method

The decentralized implementation of Lanczos method follows the same steps of the conventional method shown in Algorithm 4. Since the measurement vector $\boldsymbol{x}(t)$ in (2.10) is distributed among all the subarrays, where the kth subarray maintains the part $\boldsymbol{x}_k(t)$, the sample covariance matrix $\hat{\boldsymbol{R}}$ can not be computed at any subarray. The main idea of the decentralized implementation of the Lanczos method is similar to that of the DPM. Precisely, in the decentralized Lanczos method, each vector $\breve{\boldsymbol{u}}_j$ is partitioned into K sub-vectors as

$$\breve{\boldsymbol{u}}_j = [\breve{\boldsymbol{u}}_{j,1}^T, \ldots, \breve{\boldsymbol{u}}_{j,K}^T]^T, \tag{4.3}$$

where the kth subarray stores and updates one part of the vector $\breve{\boldsymbol{u}}_j$, which corresponds to its measurements, denoted as $\breve{\boldsymbol{u}}_{j,k}^T \in \mathbb{C}^{\breve{M}_k \times 1}$. Further, the vector $\hat{\boldsymbol{u}}_j$ is partitioned as

$$\hat{\boldsymbol{u}}_j = [\hat{\boldsymbol{u}}_{j,1}^T, \ldots, \hat{\boldsymbol{u}}_{j,K}^T]^T, \tag{4.4}$$

where the kth part $\hat{\boldsymbol{u}}_{j,k}^T \in \mathbb{C}^{\breve{M}_k \times 1}$ is stored and updated at the kth subarray. In this subsection, steps 2, 3, and 5 of the Lanczos method, which are marked as "(AC)" are implemented using the AC protocol of Section 2.4.2. Moreover, we show that the remaining steps, which are marked as "(local)" in Algorithm 4, can be carried out locally at the subarrays, i.e., they do not require communication among the subarrays.

Step 2 in Algorithm 4 is computed in a decentralized fashion similar to (2.60). Substituting the definition of the sample covariance matrix (2.15) in Step 2 yields

$$\hat{\boldsymbol{u}}_j = \frac{1}{N} \sum_{t=1}^{N} \boldsymbol{x}(t) \breve{u}_{t,j}. \tag{4.5}$$

The scalars $\breve{u}_{t,j} = \boldsymbol{x}^H(t)\breve{\boldsymbol{u}}_j$, for $t = 1, \ldots, N$, are computed using the AC protocol such that each subarray maintains the value $\breve{u}_{t,j}$, denoted as $\breve{u}_{t,j,[k]}$ at the kth subarray, as

$$\breve{u}_{t,j,[k]} = K \left[\boldsymbol{W}^{\breve{P}} \begin{bmatrix} \boldsymbol{x}_1^H(t)\breve{\boldsymbol{u}}_{j,1} \\ \vdots \\ \boldsymbol{x}_K^H(t)\breve{\boldsymbol{u}}_{j,K} \end{bmatrix} \right]_k, \tag{4.6}$$

where $\breve{P}$ is the number of AC iterations used in this step of the Lanczos method and the AC weighting matrix $\boldsymbol{W}$ is defined in (2.55). The subscript $[k]$ in the notation $\breve{u}_{t,j,[k]}$ is used since all subarrays maintain the value of $\breve{u}_{t,j}$, for details about this notation refer to Section 2.4.2. In decentralized implementation, steps 3 and 5 are written as

$$\breve{\alpha}_{j,[k]} = K \left[\boldsymbol{W}^{\breve{P}_1} \begin{bmatrix} \breve{\boldsymbol{u}}_{j,1}^H(t)\hat{\boldsymbol{u}}_{j,1} \\ \vdots \\ \breve{\boldsymbol{u}}_{j,1}^H(t)\hat{\boldsymbol{u}}_{j,K} \end{bmatrix} \right]_k \tag{4.7}$$

and

$$\hat{\alpha}_{j+1,[k]} = \left[\boldsymbol{W}^{\breve{P}_2} \begin{bmatrix} \hat{\boldsymbol{u}}_{j,1}^H(t)\hat{\boldsymbol{u}}_{j,1} \\ \vdots \\ \hat{\boldsymbol{u}}_{j,1}^H(t)\hat{\boldsymbol{u}}_{j,K} \end{bmatrix} \right]_k , \tag{4.8}$$

where $\breve{P}_1$ and $\breve{P}_2$ are the number of AC iterations used, respectively, in Step 3 and 4 of the Lanczos method. The scalars $\breve{\alpha}_{j,[k]}$ and $\hat{\alpha}_{j+1,[k]}$ are maintained by the kth subarray as indicated by the sub-index $[k]$. Thus, each subarray has access to the matrix $\breve{\boldsymbol{T}}$, denoted as $\breve{\boldsymbol{T}}_{[k]}$.

Denote the ordered eigenvalues and the corresponding eigenvectors of the matrix $\breve{\boldsymbol{T}}_{[k]}$ as $\breve{t}_{1,[k]} \geq \cdots \geq \breve{t}_{\breve{L},[k]}$ and $\breve{\boldsymbol{t}}_{1,[k]}, \ldots, \breve{\boldsymbol{t}}_{\breve{L},[k]}$, respectively. The estimation of the signal eigenvectors (Step 9 of Algorithm 4) is carried out locally for $i = 1, \ldots, \breve{L}$ as

$$\breve{\boldsymbol{u}}_{i,k} = \breve{\boldsymbol{U}}_k \breve{\boldsymbol{t}}_{i,[k]}, \tag{4.9}$$

where $\breve{\boldsymbol{U}}_k = [\breve{\boldsymbol{u}}_{1,k}, \ldots, \breve{\boldsymbol{u}}_{\breve{L},k}]$ stored at the kth subarray is the $M_k \times \breve{L}$ sub-matrix of $\breve{\boldsymbol{U}}$. The vector $\breve{\boldsymbol{u}}_{i,k}$ is the kth part of the ith eigenvector of the matrix $\hat{\boldsymbol{R}}$ computed in a decentralized fashion using the decentralized Lanczos method. At this stage, the decentralized estimation of the signal eigenvector matrix is accomplished.

In *partly calibrated arrays*, DOA estimation can be completed as presented in Section 3.1 using the decentralized ESPRIT algorithm. The resulting DOA estimation algorithm is referred to as the Lanczos-based decentralized ESPRIT algorithm (LDESPRIT). Note that only L out of $\breve{L}$ eigenvectors $\breve{\boldsymbol{t}}_{i,[k]}$ are necessary to estimate the signal eigenvector matrix $\boldsymbol{U}_\mathrm{s}$ in (2.14). If the set of eigenvalues of the matrix $\breve{\boldsymbol{T}}_{[k]}$ does not contain spurious eigenvalues, then the L principal eigenvectors of $\breve{\boldsymbol{T}}_{[k]}$ are used to estimate the signal eigenvectors $\boldsymbol{U}_\mathrm{s}$. However, this is not the case if spurious eigenvalues are encountered, e.g., as a result of an insufficient precision due to a small number of AC iterations. In the following subsection, this problem is investigated.

4.1.3 A Low Cost Scheme for Preventing the Occurrence of Spurious Eigenvalues

In [GVL13, p.566], spurious eigenvalues are defined as multiple eigenvalues of $\breve{\boldsymbol{T}}_{[k]}$ that correspond to simple (without multiplicity) eigenvalues of $\hat{\boldsymbol{R}}$. The problem of spurious eigenvalues arises when, due to precision errors, orthogonality to a converged eigenvector is lost. This problem also exists in the centralized implementation of the Lanczos method, however, due to imprecision introduced by the AC step, the problem is more pronounced in the decentralized implementation. While in the centralized Lanczos method, round-off errors resulting from the finite accuracy of the processing machine is the major source of errors, in the decentralized Lanczos method, the finite number of AC iterations gives rise to the occurrence of spurious eigenvalues. The existence of spurious eigenvalues dramatically degrades the performance of the LDESPRIT method, due to the incorrect signal eigenvectors estimation resulting from eigenvectors corresponding to the spurious eigenvalues. To overcome this problem in centralized computation, complete or selective reorthogonalization is used [GVL13]. However, in the decentralized scenario, orthogonalization has to be carried out using the AC algorithm, which requires more communication and larger convergence time. In [CW02, p.125], using the properties of the matrix $\breve{\boldsymbol{T}}_{[k]}$, a spurious eigenvalue identification test is introduced, which is locally implementable in the decentralized scenario. However, this test can not always detect the spurious eigenvalues even at high SNR. Thus, in the following, a solution, which avoids the occurrence of spurious eigenvalues with a negligible increase in the communication cost, is introduced.

As mentioned above, using an insufficiently larger number of AC iterations causes the loss of orthogonality in Lanczos method, and this, consequently, leads to the occurrence of the spurious eigenvalues. Thus, by using large values for $\breve{P}$, $\breve{P}_1$, and $\breve{P}_2$ in the LDESPRIT method, the spurious eigenvalues problem can be avoided. The communication cost of a solution with large $\breve{P}$ is similar to that of the DESPRIT method, which is not desired. Note that, at each iteration of the decentralized Lanczos method, $\breve{P}_1$ and $\breve{P}_2$ are used only in two AC operations in steps 3 and 5 while $\breve{P}$ is used in N AC operations in Step 2. Therefore, a compromise by increasing the AC iterations only in critical steps of the decentralized Lanczos method is sought. Towards this goal, observe that the eigendecomposition of $\breve{\boldsymbol{T}}$ in Step 7 of Algorithm 4 is very sensitive to the errors in $\breve{\alpha}_j$ and $\hat{\alpha}_{j+1}$. Moreover, $\breve{\alpha}_j$ and $\hat{\alpha}_{j+1}$ are used in steps 3 and 5 of Algorithm 4 to ensure the orthogonality of the vector $\breve{\boldsymbol{u}}_{j+1}$ to the vectors $\breve{\boldsymbol{u}}_j$ and $\breve{\boldsymbol{u}}_{j-1}$. Thus, accurate calculation of $\breve{\alpha}_j$ and $\hat{\alpha}_{j+1}$ is critical to prevent the loss of orthogonality, hence, to avoid the occurrence of the spurious eigenvalues. Consequently, only the number of AC iterations for steps 3 and 5 is increased to obtain more accurate estimates $\breve{\alpha}_j$ and

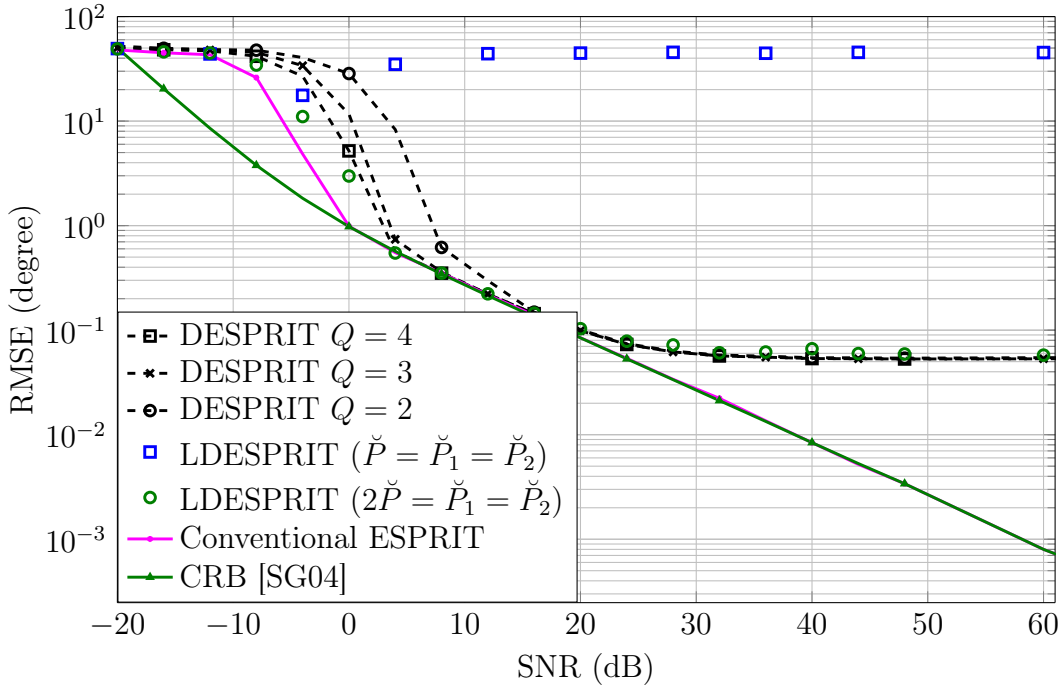

Fig. 4.1. RMSE as a function of SNR for DESPRIT and LDESPRIT algorithms.

$\hat{\alpha}_{j+1}$. Subsequently, a more accurate estimates of the matrix $\breve{\boldsymbol{T}}$ is obtained. Hence, only large values for $\breve{P}_1$ and $\breve{P}_2$ but not for $\breve{P}$ are used. Note that the increase in the communication cost due to the increase of the AC iterations in steps 3 and 5 can be considered as negligible.

4.1.4 Communication Cost Analysis

The communication cost of the proposed LDESPRIT method is $\breve{L}N\breve{P} + \breve{L}(\breve{P}_1 + \breve{P}_2)$ AC iterations. Since, for large L, $\breve{L}$ is of order L (in the simulations, $\breve{L} = L + 2$ is used), and, $\breve{P}_1$ and $\breve{P}_2$ can be selected such that $N\breve{P} \gg (\breve{P}_1 + \breve{P}_2)$ (in the simulations, $\breve{P}_2 = \breve{P}_1 = 2\breve{P}$), the cost of the LDESPRIT method is of order $LN\breve{P}$ AC iterations, which corresponds to a Q fold reduction as compared to the DESPRIT method, refer to Section 3.1.1.

4.1.5 Simulation Results

In this section, an array of $K = 6$ subarrays is considered. The array configuration is similar to that of Section 3.4.1. Five independent equal-powered sources impinge

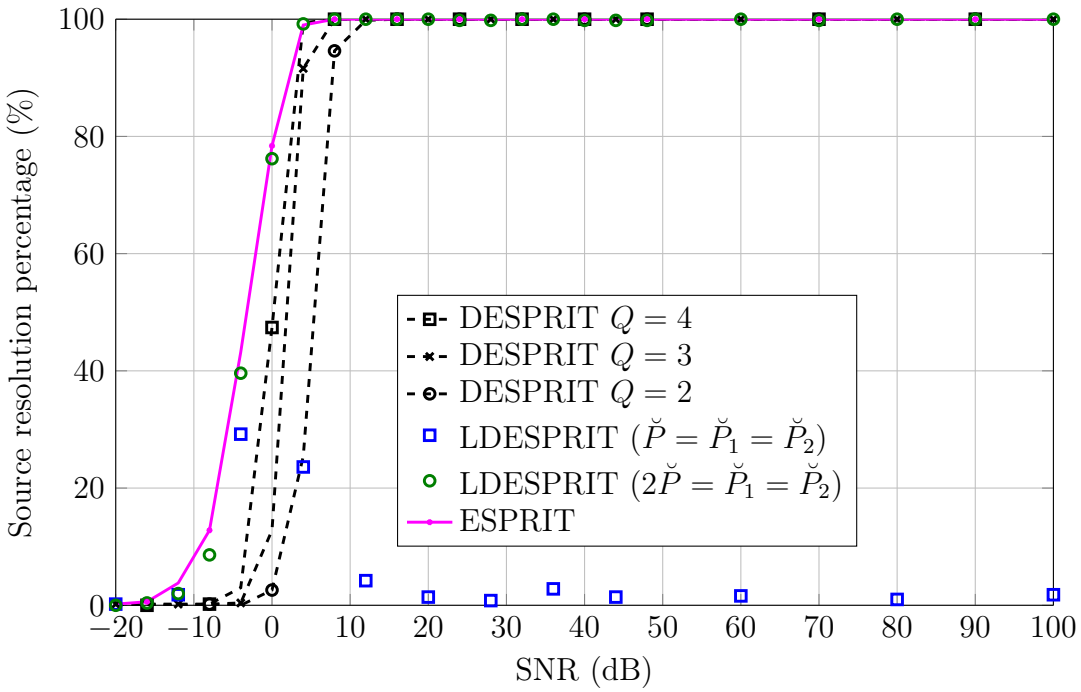

Fig. 4.2. Resolution percentage as a function of SNR for DESPRIT and LDESPRIT.

onto the subarrays from directions $-30°, -20°, 0°, 27°$, and $30°$. Each subarray collects $N = 500$ snapshots. The performance of the DESPRIT algorithm is shown for a number of AC iterations equal to 40 and for numbers of PM iteration Q of 2, 3, and 4. The LDESPRIT algorithm with $\breve{L} = L + 2 = 6$ is considered in two setups. In the first setup, the values $\breve{P} = \breve{P}_1 = \breve{P}_2 = 40$ are chosen, while in the second setup $\breve{P} = 40$ and $\breve{P}_1 = \breve{P}_2 = 80$. All results are averaged over 500 realizations.

Fig. 4.1 displays the performance of the LDESPRIT and the DESPRIT methods with the above mentioned settings. The CRB for partly calibrated arrays and the performance of centralized ESPRIT algorithm [RK89] are also shown. The LDESPRIT method with $\breve{P}_1 = \breve{P}_2 = 40$ performs poorly compared to the other methods due to the spurious eigenvalues. As can be inferred from Fig. 4.1, using LDESPRIT with $\breve{P}_1 = \breve{P}_2 = 80$ (even with $\breve{P} = 40$) prevents the occurrence of spurious eigenvalues in all the runs. It can be observed that the performance of LDESPRIT with $\breve{P}_1 = \breve{P}_2 = 80$ is similar to the performance of the DESPRIT method with $Q = 3$, while a reduction in the communication cost of factor 3 is gained.

Fig. 4.2, displays the resolution percentage, as defined in [PPG11], of all the mentioned methods. Note that, the occurrence of the spurious eigenvalues limits the capability of the LDESPRIT method with $\breve{P}_1 = \breve{P}_2 = 40$ in resolving the sources even at high SNR. The LDESPRIT method with $\breve{P}_1 = \breve{P}_2 = 80$ achieves resolution probability similar to

the DESPRIT and both methods achieve a 100% resolution percentage at high SNR. Thus, the proposed LDESPRIT method with $\breve{P}_1 = \breve{P}_2 = 80$ achieves the same DOA estimation and resolution performance as the DESPRIT method with $Q = 3$ while the communicational cost is substantially reduced.

4.2 Decentralized Generalized Eigendecomposition

An online algorithm, which tracks the generalized eigenvalues of a non-Hermitian pair of covariance matrices, from which the DOAs are estimated, is proposed. In contrast to the DPM, the proposed algorithm does not assume the sources to be stationary during the sampling time. The algorithm is applicable in *partly calibrated shift-invariant arrays*, refer to Section 2.3.1. In other words, similar to the ESPRIT algorithm, the subarrays are assumed to be shift-invariant and the displacement between the subarrays are unknown. The generalized eigendecomposition problem is converted to an eigendecomposition problem. The approximate simultaneous power method, which is proposed in [HXC+99] under the name "natural power method" (NP2), is used to compute the eigenvalues of the resulting eigendecomposition problem. The resulting centralized algorithm is referred to as the generalized eigendecomposition-based ESPRIT algorithm (GESPRIT). Based on this (GESPRIT) algorithm, fully decentralized online DOA estimation algorithm, referred to as decentralized GESPRIT algorithm (DGESPRIT), is introduced.

4.2.1 Generalized Eigendecomposition and DOA Estimation

In the sequel, DOA estimation based on the generalized eigendecomposition[1] as introduced in [Oui86], is reviewed. Let

$$\overline{\boldsymbol{x}}(t) = \overline{\boldsymbol{J}}\boldsymbol{x}(t) \tag{4.10}$$

and

$$\underline{\boldsymbol{x}}(t) = \underline{\boldsymbol{J}}\boldsymbol{x}(t) \tag{4.11}$$

denote the measurements corresponding, respectively, to the upper and lower sensor groups of the shift-invariant array. Further, two matrices $\overline{\boldsymbol{C}}$ and $\underline{\boldsymbol{C}}$ are defined as

$$\overline{\boldsymbol{C}} = \mathbb{E}\big(\overline{\boldsymbol{x}}(t)\overline{\boldsymbol{x}}^H(t)\big) - \sigma^2\boldsymbol{I}_{\overline{M}} \tag{4.12}$$

[1] Also called the matrix pencil.

and

$$\underline{\boldsymbol{C}} = \mathbb{E}\big(\underline{\boldsymbol{x}}(t)\overline{\boldsymbol{x}}^H(t)\big), \tag{4.13}$$

where $\overline{M}$ is the number of sensors in the upper and lower sensor groups. Let $g_1, \ldots, g_L \in \mathbb{C}$ be the L generalized eigenvalues of the matrix pair $(\overline{\boldsymbol{C}}, \underline{\boldsymbol{C}})$, which have the largest amplitudes, then,

$$\theta_l = \sin^{-1}\big(\arg(g_l)\lambda_c/(2\pi d)\big), \tag{4.14}$$

for $l = 1, \ldots, L$, where λ_c is the carrier frequency and d is the displacement between the upper and lower groups, for details see [Oui86].

4.2.2 The GESPRIT Algorithm

In this subsection, an online algorithm, which tracks the generalized eigenvalues of the non-Hermitian matrix pair $(\overline{\boldsymbol{C}}, \underline{\boldsymbol{C}})$ for each measurement vector in a centralized setup, is proposed. In the following subsection a decentralized implementation of this algorithm is introduced. The lth generalized eigenvalue g_l of the matrix pair $(\overline{\boldsymbol{C}}, \underline{\boldsymbol{C}})$ and its corresponding right generalized eigenvector $\boldsymbol{h}_l$ are defined as [BDD+00, p. 233]

$$\underline{\boldsymbol{C}}\boldsymbol{h}_l = g_l\overline{\boldsymbol{C}}\boldsymbol{h}_l. \tag{4.15}$$

By multiplying (4.15) with $\overline{\boldsymbol{C}}^{-1}$, the generalized eigendecomposition is reduced to an eigendecomposition of the form

$$\overline{\boldsymbol{C}}^{-1}\underline{\boldsymbol{C}}\boldsymbol{h}_l = g_l\boldsymbol{h}_l. \tag{4.16}$$

In very large sensor networks, the dimensions of $\overline{\boldsymbol{C}}$ are very large, consequently, computing its inverse is impractical. Thus, iterative methods for finding g_l and $\boldsymbol{h}_l$ from (4.16) are sought. In these iterative methods, multiplying a vector with $\overline{\boldsymbol{C}}^{-1}$ is achieved by iteratively solving a system of linear equations, see [BDD+00, Chapter. 8] for examples. In decentralized implementations, the iterative procedure results in a large undesired communication costs. To avoid the iterative procedure, we approximate the matrix $\overline{\boldsymbol{C}}$, which has only L dominant eigenvalues corresponding to the L DOAs [RK89], as

$$\overline{\boldsymbol{C}} = \overline{\boldsymbol{H}}\,\overline{\boldsymbol{G}}\,\overline{\boldsymbol{H}}^H, \tag{4.17}$$

where $\overline{\boldsymbol{G}} = \operatorname{diag}\big(\overline{g}_1, \ldots, \overline{g}_L\big)$ and $\overline{\boldsymbol{H}} = [\overline{\boldsymbol{h}}_1, \ldots, \overline{\boldsymbol{h}}_L]$ are the matrices containing the largest L eigenvalues of $\overline{\boldsymbol{C}}$ and their corresponding eigenvectors, respectively. Substituting (4.17) in (4.16) yields

$$\overline{\boldsymbol{H}}\,\overline{\boldsymbol{G}}^{-1}\overline{\boldsymbol{H}}^H\underline{\boldsymbol{C}}\boldsymbol{h}_l = g_l\boldsymbol{h}_l. \tag{4.18}$$

Equation (4.18) is rewritten in a matrix form as

$$\overline{\boldsymbol{H}}\,\overline{\boldsymbol{G}}^{-1}\overline{\boldsymbol{H}}^{H}\underline{\boldsymbol{C}}\boldsymbol{H} = \boldsymbol{H}\boldsymbol{G}, \tag{4.19}$$

where $\boldsymbol{G} = \operatorname{diag}\big(g_1, \ldots, g_L\big)$ and $\boldsymbol{H} = [\boldsymbol{h}_1, \ldots, \boldsymbol{h}_L]$. The sample estimates of $\overline{\boldsymbol{C}}$ and $\underline{\boldsymbol{C}}$ can be defined as a rank one update at each time instant t, as follows

$$\begin{aligned}\hat{\overline{\boldsymbol{C}}}(t) &= \varepsilon\hat{\overline{\boldsymbol{C}}}(t-1) + \overline{\boldsymbol{x}}(t)\,\overline{\boldsymbol{x}}^{H}(t) - \sigma^2\boldsymbol{I}_{\overline{M}},\\ \hat{\underline{\boldsymbol{C}}}(t) &= \varepsilon\hat{\underline{\boldsymbol{C}}}(t-1) + \underline{\boldsymbol{x}}(t)\,\overline{\boldsymbol{x}}^{H}(t),\end{aligned} \tag{4.20}$$

where $0 \leq \varepsilon \leq 1$ is a forgetting factor [HXC+99] and the noise variance σ^2 is considered to be known. Let $\hat{\overline{\boldsymbol{G}}}(t)$, $\hat{\overline{\boldsymbol{H}}}(t)$, $\hat{\boldsymbol{G}}(t)$ and $\hat{\boldsymbol{H}}(t)$ be the sample estimates of $\overline{\boldsymbol{G}}(t), \overline{\boldsymbol{H}}(t), \boldsymbol{G}(t)$ and $\boldsymbol{H}(t)$, respectively, at time instant t. Based on (4.20), our proposed generalized eigendecomposition algorithm updates $\hat{\overline{\boldsymbol{G}}}(t), \hat{\overline{\boldsymbol{H}}}(t), \hat{\boldsymbol{G}}(t)$ and $\hat{\boldsymbol{H}}(t)$ at time t according to the newly acquired measurement vectors $\overline{\boldsymbol{x}}(t)$ and $\underline{\boldsymbol{x}}(t)$. The approximate power method [BDD+00, p. 51] is used for this update since even for non-Hermitian matrices, the power iteration converges to the eigenvalue with the largest amplitude [Ste76]. The online update of the matrix $\hat{\overline{\boldsymbol{H}}}(t)$ is carried out using the approximate simultaneous power method, which is proposed in [HXC+99] under the name "natural power method" (NP2).

In the NP2 algorithm, at time t, the matrix $\hat{\overline{\boldsymbol{H}}}(t)$ is computed as follows

$$\hat{\overline{\boldsymbol{H}}}(t) = \overline{\boldsymbol{Y}}(t)\overline{\boldsymbol{F}}^{-1/2}(t), \tag{4.21}$$

where $\overline{\boldsymbol{Y}}(t) = \hat{\overline{\boldsymbol{C}}}(t)\hat{\overline{\boldsymbol{H}}}(t-1)$ and $\overline{\boldsymbol{F}}(t) = \overline{\boldsymbol{Y}}^{H}(t)\overline{\boldsymbol{Y}}(t)$. We remark that, in contrast to the conventional PM algorithm [GVL13], where Q PM iterations are carried out to compute each eigenvector using the collected batch of samples as shown in Section 2.4.1, in the NP2 the eigenvectors are updated at each time instant t. Substituting (4.20) in the definition of $\overline{\boldsymbol{Y}}(t)$ yields

$$\overline{\boldsymbol{Y}}(t) = \varepsilon\overline{\boldsymbol{Y}}(t-1) + \overline{\boldsymbol{x}}(t)\overline{\boldsymbol{y}}^{H}(t) - \sigma^2\hat{\overline{\boldsymbol{H}}}(t-1), \tag{4.22}$$

where $\overline{\boldsymbol{y}}(t) = \hat{\overline{\boldsymbol{H}}}^{H}(t-1)\overline{\boldsymbol{x}}(t)$ and the approximation $\hat{\overline{\boldsymbol{H}}}(t-1) \approx \hat{\overline{\boldsymbol{H}}}(t-2)$ is used [HXC+99]. Moreover, the matrix $\overline{\boldsymbol{F}}(t)$ can be rewritten as

$$\begin{aligned}\overline{\boldsymbol{F}}(t) = {}& \varepsilon^2\overline{\boldsymbol{F}}(t-1) + \overline{\boldsymbol{f}}(t)\overline{\boldsymbol{y}}^{H}(t) - \varepsilon\sigma^2\overline{\boldsymbol{Y}}^{H}(t-1)\hat{\overline{\boldsymbol{H}}}(t-1)\\ &+ \overline{\boldsymbol{y}}(t)\overline{\boldsymbol{f}}^{H}(t) + \overline{\boldsymbol{y}}(t)\eta_{\overline{\boldsymbol{x}}}(t)\overline{\boldsymbol{y}}^{H}(t) - \sigma^2\overline{\boldsymbol{y}}(t)\overline{\boldsymbol{y}}^{H}(t)\\ &- \sigma^2\varepsilon\hat{\overline{\boldsymbol{H}}}^{H}(t-1)\overline{\boldsymbol{Y}}(t-1) - \sigma^2\overline{\boldsymbol{y}}(t)\overline{\boldsymbol{y}}^{H}(t) + \sigma^4\hat{\overline{\boldsymbol{H}}}^{H}(t-1)\hat{\overline{\boldsymbol{H}}}(t-1)\end{aligned} \tag{4.23}$$

where

$$\overline{\boldsymbol{f}}(t) = \varepsilon\overline{\boldsymbol{Y}}^{H}(t-1)\overline{\boldsymbol{x}}(t) = \varepsilon\big(\hat{\overline{\boldsymbol{H}}}(t-1)\overline{\boldsymbol{F}}^{1/2}(t-1)\big)^{H}\overline{\boldsymbol{x}}(t) = \varepsilon\big(\overline{\boldsymbol{F}}^{1/2}(t-1)\big)^{H}\overline{\boldsymbol{y}}(t) \tag{4.24}$$

and

$$\eta_{\overline{\boldsymbol{x}}}(t) = \overline{\boldsymbol{x}}^H(t)\overline{\boldsymbol{x}}(t). \tag{4.25}$$

Since $\hat{\overline{\boldsymbol{H}}}^H(t-1)\hat{\overline{\boldsymbol{H}}}(t-1) = \boldsymbol{I}_L$ and $\hat{\overline{\boldsymbol{H}}}^H(t-1)\overline{\boldsymbol{Y}}(t-1) = \hat{\overline{\boldsymbol{H}}}^H(t-1)\hat{\overline{\boldsymbol{H}}}(t-1)\overline{\boldsymbol{F}}^{1/2}(t-1)$ $= \overline{\boldsymbol{F}}^{1/2}(t-1)$, (4.23) is further simplified to

$$\begin{aligned}\overline{\boldsymbol{F}}(t) = \varepsilon^2\overline{\boldsymbol{F}}(t-1) + \overline{\boldsymbol{f}}(t)\overline{\boldsymbol{y}}^H(t) + \overline{\boldsymbol{y}}(t)\overline{\boldsymbol{f}}^H(t) + (\eta_{\overline{\boldsymbol{x}}}(t) - 2\sigma^2)\overline{\boldsymbol{y}}(t)\overline{\boldsymbol{y}}^H(t) \\ - \varepsilon\sigma^2\overline{\boldsymbol{F}}^{1/2}(t-1) - \varepsilon\sigma^2\big(\overline{\boldsymbol{F}}^{1/2}(t-1)\big)^H + \sigma^4\boldsymbol{I}_L.\end{aligned} \tag{4.26}$$

Equations (4.21)–(4.26) represent the update of $\hat{\overline{\boldsymbol{H}}}^H(t)$ at time t. Note that although the update of the eigenvectors $\hat{\overline{\boldsymbol{H}}}^H(t)$ of the matrix $\overline{\boldsymbol{C}}$ is considered in equations (4.21)–(4.26), the update of the eigenvalues $\overline{\boldsymbol{G}}(t)$ is not treated. We proposed to update the eigenvalues $\overline{\boldsymbol{G}}(t)$ as follows. Substituting (4.20) in the definition of the eigenvalues $\hat{\overline{\boldsymbol{G}}}(t) = \hat{\overline{\boldsymbol{H}}}^H(t)\hat{\overline{\boldsymbol{C}}}(t)\hat{\overline{\boldsymbol{H}}}(t)$ produces

$$\hat{\overline{\boldsymbol{G}}}(t) = \varepsilon\hat{\overline{\boldsymbol{H}}}^H(t)\hat{\overline{\boldsymbol{C}}}(t-1)\hat{\overline{\boldsymbol{H}}}(t) + \overline{\boldsymbol{y}}(t)\overline{\boldsymbol{y}}^H(t) - \sigma^2\boldsymbol{I}_L. \tag{4.27}$$

In analogy to [HXC+99], the approximation $\hat{\overline{\boldsymbol{H}}}(t) \approx \hat{\overline{\boldsymbol{H}}}(t-1)$[2], is substituted in (4.27) which yields

$$\hat{\overline{\boldsymbol{G}}}(t) = \varepsilon\hat{\overline{\boldsymbol{G}}}(t-1) + \operatorname{diag}\big(\overline{\boldsymbol{y}}(t) \odot \overline{\boldsymbol{y}}^*(t)\big) - \sigma^2\boldsymbol{I}_L, \tag{4.28}$$

where the off-diagonal elements of $\overline{\boldsymbol{y}}(t)\overline{\boldsymbol{y}}^H(t)$ are zeros since both matrices $\hat{\overline{\boldsymbol{G}}}(t)$ and $\hat{\overline{\boldsymbol{G}}}(t-1)$ are diagonal. Observe that, by using (4.21)–(4.26) and (4.28), the update of both $\overline{\boldsymbol{G}}(t)$ and $\overline{\boldsymbol{H}}(t)$ is achieved. In the following, the NP2 and the proposed eigenvalues update in (4.28) are applied to achieve the generalized eigendecomposition of (4.19), i.e., to update $\boldsymbol{G}(t)$ and $\boldsymbol{H}(t)$.

In analogy to (4.21), applying the NP2 to the eigendecomposition in (4.19) results in the following iteration

$$\hat{\boldsymbol{H}}(t) = \boldsymbol{Y}(t)\boldsymbol{F}^{-1/2}(t) \tag{4.29}$$

where $\boldsymbol{F}(t) = \boldsymbol{Y}^H(t)\boldsymbol{Y}(t)$ and $\boldsymbol{Y}(t) = \hat{\overline{\boldsymbol{H}}}(t-1)\hat{\overline{\boldsymbol{G}}}^{-1}(t-1)\hat{\overline{\boldsymbol{H}}}^H(t-1)\hat{\underline{\boldsymbol{C}}}(t)\hat{\boldsymbol{H}}(t-1)$. Substituting (4.20) in the definition of $\boldsymbol{Y}(t)$ yields

$$\boldsymbol{Y}(t) = \varepsilon\boldsymbol{Y}(t-1) + \underline{\boldsymbol{q}}(t)\boldsymbol{f}^H(t), \tag{4.30}$$

where $\boldsymbol{f}(t) = \hat{\boldsymbol{H}}^H(t-1)\overline{\boldsymbol{x}}(t)$, $\underline{\boldsymbol{q}}(t) = \hat{\overline{\boldsymbol{H}}}(t-1)\hat{\overline{\boldsymbol{G}}}^{-1}(t-1)\underline{\boldsymbol{y}}(t)$, $\underline{\boldsymbol{y}}(t) = \hat{\overline{\boldsymbol{H}}}^H(t-1)\underline{\boldsymbol{x}}(t)$, and the approximations $\hat{\overline{\boldsymbol{H}}}(t-2) \approx \hat{\overline{\boldsymbol{H}}}(t-1)$ and $\hat{\boldsymbol{H}}(t-2) \approx \hat{\boldsymbol{H}}(t-1)$ are used.

The matrix $\boldsymbol{F}(t)$ can be rewritten as

$$\boldsymbol{F}(t) = \varepsilon^2\boldsymbol{F}(t-1) + \underline{\boldsymbol{n}}(t)\boldsymbol{f}^H(t) + \boldsymbol{f}(t)\underline{\boldsymbol{n}}^H(t) + \eta_{\underline{\boldsymbol{q}}}(t)\boldsymbol{f}(t)\boldsymbol{f}^H(t), \tag{4.31}$$

[2]This approximation is justified at convergence, i.e., when $t \to \infty$ $\hat{\overline{\boldsymbol{H}}}(t) = \hat{\overline{\boldsymbol{H}}}(t-1)$.

where $\underline{\boldsymbol{n}}(t) = \varepsilon \boldsymbol{Y}^H(t-1)\underline{\boldsymbol{q}}(t) = \varepsilon(\boldsymbol{F}^{1/2}(t-1))^H \boldsymbol{q}(t)$, $\boldsymbol{q}(t) = \hat{\boldsymbol{H}}^H(t-1)\underline{\boldsymbol{q}}(t)$, and $\eta_{\underline{\boldsymbol{q}}}(t) = \underline{\boldsymbol{q}}^H(t)\underline{\boldsymbol{q}}(t)$. Similar to the proposition in (4.28) the generalized eigenvalues can be updated as

$$\hat{\boldsymbol{G}}(t) = \varepsilon \hat{\boldsymbol{G}}(t-1) + \operatorname{diag}\big(\boldsymbol{q}(t) \odot \boldsymbol{f}^*(t)\big). \tag{4.32}$$

Equations (4.29)–(4.32), represent the update of the generalized eigenvalues and eigenvectors of the matrix pair $(\overline{\boldsymbol{C}}, \underline{\boldsymbol{C}})$.

Algorithm 5 The centralized GESPRIT

Step 0: Init $\hat{\overline{\boldsymbol{H}}}(0)$ and $\hat{\boldsymbol{H}}(0)$ at random and orthogonalize them. Set $\overline{\boldsymbol{Y}}(0)$, $\boldsymbol{Y}(0)$, $\overline{\boldsymbol{F}}(0)$, $\boldsymbol{F}(0)$, $\hat{\overline{\boldsymbol{G}}}(0)$ and $\hat{\boldsymbol{G}}(0)$ to zero.
for each input $\overline{\boldsymbol{x}}(t)$ and $\underline{\boldsymbol{x}}(t)$ **do**
Part I: Network computation (AC)
Step 1 (AC1): $\overline{\boldsymbol{y}}(t) \leftarrow \hat{\overline{\boldsymbol{H}}}^H(t-1)\overline{\boldsymbol{x}}(t)$
Step 2 (AC1): $\underline{\boldsymbol{y}}(t) \leftarrow \hat{\overline{\boldsymbol{H}}}^H(t-1)\underline{\boldsymbol{x}}(t)$
Step 3 (AC1): $\eta_{\overline{\boldsymbol{x}}}(t) \leftarrow \overline{\boldsymbol{x}}^H(t)\overline{\boldsymbol{x}}(t)$
Step 4 (AC1): $\boldsymbol{f}(t) \leftarrow \hat{\boldsymbol{H}}^H(t-1)\overline{\boldsymbol{x}}(t)$
Step 5: $\underline{\boldsymbol{q}}(t) \leftarrow \hat{\overline{\boldsymbol{H}}}(t-1)\hat{\overline{\boldsymbol{G}}}^{-1}(t-1)\underline{\boldsymbol{y}}(t)$
Step 6 (AC2): $\eta_{\underline{\boldsymbol{q}}}(t) \leftarrow \underline{\boldsymbol{q}}^H(t)\underline{\boldsymbol{q}}(t)$
Step 7 (AC2): $\boldsymbol{q}(t) \leftarrow \hat{\boldsymbol{H}}^H(t-1)\underline{\boldsymbol{q}}(t)$
Part II: Node computation (local update)
Step 8: $\overline{\boldsymbol{Y}}(t) \leftarrow \varepsilon\overline{\boldsymbol{Y}}(t-1) + \overline{\boldsymbol{x}}(t)\overline{\boldsymbol{y}}^H(t) - \sigma^2\hat{\overline{\boldsymbol{H}}}(t-1)$
Step 9: $\overline{\boldsymbol{f}}(t) \leftarrow \varepsilon\big(\overline{\boldsymbol{F}}^{1/2}(t-1)\big)^H\overline{\boldsymbol{y}}(t)$
Step 10: $\overline{\boldsymbol{F}}(t) \leftarrow \varepsilon^2\overline{\boldsymbol{F}}(t-1) + \overline{\boldsymbol{f}}(t)\overline{\boldsymbol{y}}^H(t) + \overline{\boldsymbol{y}}(t)\overline{\boldsymbol{f}}^H(t) + (\eta_{\overline{\boldsymbol{x}}}(t) - 2\sigma^2)\overline{\boldsymbol{y}}(t)\overline{\boldsymbol{y}}^H(t)$
$-\varepsilon\sigma^2\overline{\boldsymbol{F}}^{1/2}(t-1) - \varepsilon\sigma^2\big(\overline{\boldsymbol{F}}^{1/2}(t-1)\big)^H + \sigma^4\boldsymbol{I}_L$
Step 11: $\hat{\overline{\boldsymbol{H}}}(t) \leftarrow \overline{\boldsymbol{Y}}(t)\overline{\boldsymbol{F}}^{-1/2}(t)$
Step 12: $\hat{\overline{\boldsymbol{G}}}(t) = \varepsilon\hat{\overline{\boldsymbol{G}}}(t-1) + \operatorname{diag}\big(\overline{\boldsymbol{y}}(t) \odot \overline{\boldsymbol{y}}^*(t)\big) - \sigma^2\boldsymbol{I}_L$
Step 13: $\boldsymbol{Y}(t) \leftarrow \varepsilon\boldsymbol{Y}(t-1) + \underline{\boldsymbol{q}}(t)\boldsymbol{f}^H(t)$
Step 14: $\underline{\boldsymbol{n}}(t) \leftarrow \varepsilon(\boldsymbol{F}^{1/2}(t-1))^H\boldsymbol{q}(t)$
Step 15: $\boldsymbol{F}(t) \leftarrow \varepsilon^2\boldsymbol{F}(t-1) + \underline{\boldsymbol{n}}(t)\boldsymbol{f}^H(t) + \boldsymbol{f}(t)\underline{\boldsymbol{n}}^H(t) + \eta_{\underline{\boldsymbol{q}}}(t)\boldsymbol{f}(t)\boldsymbol{f}^H(t)$
Step 16: $\hat{\boldsymbol{H}}(t) \leftarrow \boldsymbol{Y}(t)\boldsymbol{F}^{-1/2}(t)$
Step 17: $\hat{\boldsymbol{G}}(t) \leftarrow \varepsilon\hat{\boldsymbol{G}}(t-1) + \operatorname{diag}\big(\boldsymbol{q}(t) \odot \boldsymbol{f}^*(t)\big)$
Step 18: DOA estimation using (4.14).
end for

The centralized generalized eigendecomposition is summarized in Algorithm 5. Note that the steps are rearranged and clustered into two parts. Part I contains all the operations that require communication among the subarrays in the decentralized implementation, which will be introduced later. Part II contains local updates, which are carried out at the subarrays and do not require communication between the subarrays.

In the next section, it is shown how Algorithm 5 can be implemented in decentralized fashion, using the AC protocol in Section 2.4.2.

4.2.3 The DGESPRIT Algorithm

In the proposed decentralized implementation, each subarray stores locally the part of the variables $\overline{\breve{\boldsymbol{H}}}(t), \breve{\boldsymbol{H}}(t), \overline{\breve{\boldsymbol{Y}}}(t), \breve{\boldsymbol{Y}}(t)$ and $\breve{\underline{\boldsymbol{q}}}(t)$ which corresponds to its measurements where the notation $\breve{(\cdot)}$ is presented in 2.4.2. Thus, these variables are partitioned as follows, $\overline{\breve{\boldsymbol{H}}}(t) = [\overline{\breve{\boldsymbol{H}}}_1^T(t) \, , \ldots , \overline{\breve{\boldsymbol{H}}}_K^T(t)]^T$, where the kth matrix block $\overline{\breve{\boldsymbol{H}}}_k(t)$ is stored locally at the kth subarray. Note that in this partition, each subarray stores a part of the lth estimated eigenvector $\overline{\breve{\boldsymbol{h}}}_l(t) = [\overline{\breve{\boldsymbol{h}}}_{l,1}^T(t), \ldots, \overline{\breve{\boldsymbol{h}}}_{l,K}^T(t)]^T$. The same partition is assumed for $\breve{\boldsymbol{H}}(t), \overline{\breve{\boldsymbol{Y}}}(t), \breve{\boldsymbol{Y}}(t)$ and $\breve{\underline{\boldsymbol{q}}}(t)$. Moreover, in the proposed decentralized implementation, using the AC protocol, the kth subarray maintains a local copy of the variables $\overline{\boldsymbol{y}}(t)$, $\overline{\boldsymbol{f}}(t)$, $\underline{\boldsymbol{y}}(t)$, $\eta_{\overline{\boldsymbol{x}}}(t)$, $\boldsymbol{f}_k(t)$, $\eta_{\underline{\boldsymbol{q}}}(t)$, $\boldsymbol{q}(t)$, $\underline{\boldsymbol{n}}(t)$, $\hat{\overline{\boldsymbol{G}}}(t)$, $\hat{\boldsymbol{G}}(t)$, $\overline{\boldsymbol{F}}(t)$ and $\boldsymbol{F}(t)$, denoted as $\breve{\overline{\boldsymbol{y}}}_{[k]}(t)$, $\breve{\overline{\boldsymbol{f}}}_{[k]}(t)$, $\breve{\underline{\boldsymbol{y}}}_{[k]}(t)$, $\breve{\eta}_{\overline{\boldsymbol{x}},[k]}(t)$, $\breve{\boldsymbol{f}}_{[k]}(t)$, $\breve{\eta}_{\underline{\boldsymbol{q}},[k]}(t)$, $\breve{\boldsymbol{q}}_{[k]}(t)$, $\breve{\underline{\boldsymbol{n}}}_{[k]}(t)$, $\breve{\overline{\boldsymbol{G}}}_{[k]}(t)$, $\breve{\boldsymbol{G}}_{[k]}(t)$, $\breve{\overline{\boldsymbol{F}}}_{[k]}(t)$ and $\breve{\boldsymbol{F}}_{[k]}(t)$, where the $\breve{(\cdot)}_{[k]}$ is used as in Section 2.4.2.

In Algorithm 5, all operations, which require the AC protocol, are arranged in Part I. These operations are of two types. The first type is matrix-vector multiplication and includes steps 1, 2, 4 and 7. The second type is vector-vector multiplication and it includes steps 3 and 6. Step 1 which is a matrix-vector multiplication can be rewritten as

$$\overline{\boldsymbol{y}}(t) = [\hat{\overline{\boldsymbol{h}}}_1^H(t-1)\overline{\boldsymbol{x}}(t), \ldots, \hat{\overline{\boldsymbol{h}}}_L^H(t-1)\overline{\boldsymbol{x}}(t)]^T, \tag{4.33}$$

in centralized scenario. The lth entry of the vector $\overline{\boldsymbol{y}}(t)$ can be rewritten as an average of K scalars which are distributed over the K subarrays as follows

$$\hat{\overline{\boldsymbol{h}}}_l^H(t-1)\overline{\boldsymbol{x}}(t) = K\Big(\frac{1}{K}\sum_{k=1}^{K}\hat{\overline{\boldsymbol{h}}}_{l,k}^H(t-1)\overline{\boldsymbol{x}}_k(t)\Big), \tag{4.34}$$

where the scalar $\hat{\overline{\boldsymbol{h}}}_{l,k}^H(t-1)\overline{\boldsymbol{x}}_k(t)$ is computed locally at the kth subarray. In the proposed decentralized implementation, the AC protocol, introduced in Section 2.4.2, is used to compute this average such that all subarrays maintain access to this average. Thus, using L AC operations, the kth subarray computes a local estimate of $\overline{\boldsymbol{y}}(t)$, which is denoted earlier as $\breve{\overline{\boldsymbol{y}}}_{[k]}(t)$. The remaining matrix-vector multiplication operations are achieved in similar manner. Also the vector-vector multiplications in steps 3 and 6 are carried out as in (4.34).

Note that Step 6 and all steps of Part II of Algorithm 5 are local steps, where the kth subarray computes either its local copy of the variables $\breve{\overline{\boldsymbol{G}}}_{[k]}(t), \breve{\boldsymbol{G}}_{[k]}(t), \breve{\overline{\boldsymbol{F}}}_{[k]}(t)$, $\breve{\boldsymbol{F}}_{[k]}(t)$,

and the L DOAs, or it computes the part of the variables $\overset{\check{}}{\overline{\boldsymbol{H}}}(t), \check{\boldsymbol{H}}(t), \overset{\check{}}{\overline{\boldsymbol{Y}}}(t), \check{\boldsymbol{Y}}(t)$ and $\check{\underline{\boldsymbol{q}}}(t)$ which correspond to its measurements.

4.2.4 Communication Cost Analysis

In Algorithm 5, the AC protocol is used for matrix-vector multiplications in steps 1, 2, 4 and 7. Each of these multiplications requires L AC operations. The AC protocol is also used for vector-vector multiplications in steps 3 and 6. Thus, the communication cost associated with the proposed Algorithm at each iteration is $4L + 2$ AC protocols.

The steps of Algorithm 5, which are marked as AC1 do not depend on each other and are carried out in parallel using $3L+1$ parallel AC protocol. Note that running parallel AC protocols minimizes the latency and the communication overhead of the proposed algorithm. The same applies to the steps which are marked as AC2.

4.2.5 Simulation Results

An array composed of $K = 6$ subarrays, where each subarray consists of $M_k = 2$ sensors is considered. The array configuration is the same as that of Section 3.4.1.

In the first simulation, the signals of $L = 3$ equal-powered stationary sources impinge onto the array from directions $-14°$, $-10°$ and $5°$ with SNR = 10 dB. For each AC operation, 15 AC iterations are carried out, using the weighting scheme from (3.34). The forgetting factor is taken to be $\varepsilon = 0.99$. Fig. 4.3 illustrates the DOA estimates obtained from one subarray for $t = 1, \ldots, 10000$. Note that after $t = 700$, the DGESPRIT algorithm is able to resolve the three DOAs within a reasonable accuracy.

In the second simulation, two equal-powered moving sources with SNR = 10 dB are considered. The direction of the sources are changing linearly with time from $\theta_1 = 40°$ and $\theta_2 = 0°$ at $t = 0$ to $\theta_1 = 0°$ and $\theta_2 = 40°$ at $t = 10000$. The parameter setup of the AC algorithm is taken as in the first simulation. The forgetting factor is set to $\varepsilon = 0.95$. Fig. 4.4 displays the estimated DOAs, at one subarray for $t = 1, \ldots, 10000$. The DGESPRIT algorithm is able to track the DOAs after $t = 700$. However, when the angular separation between the two sources is small, which correspond to the region around $t = 5000$ in Fig. 4.4, the DGESPRIT algorithm is not able to resolve the two sources, and noisy generalized eigenvalues appear causing errors. Observe in Fig. 4.4 that the DGESPRIT algorithm is able to recover and estimate the DOAs of the two sources when the angular separation is enough.

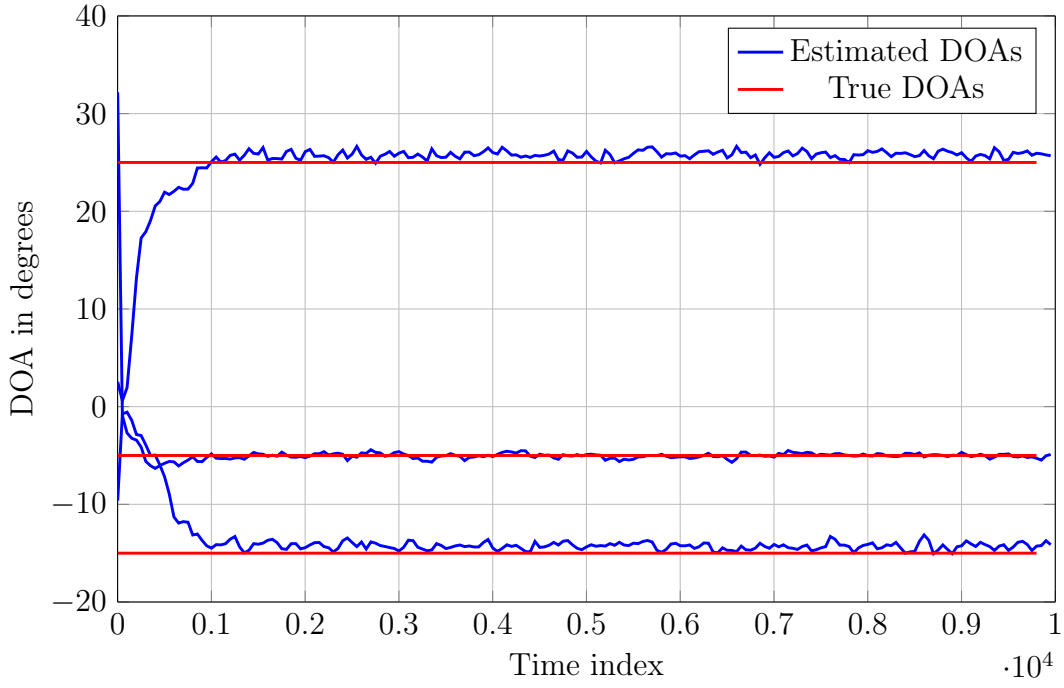

Fig. 4.3. Stationary sources.

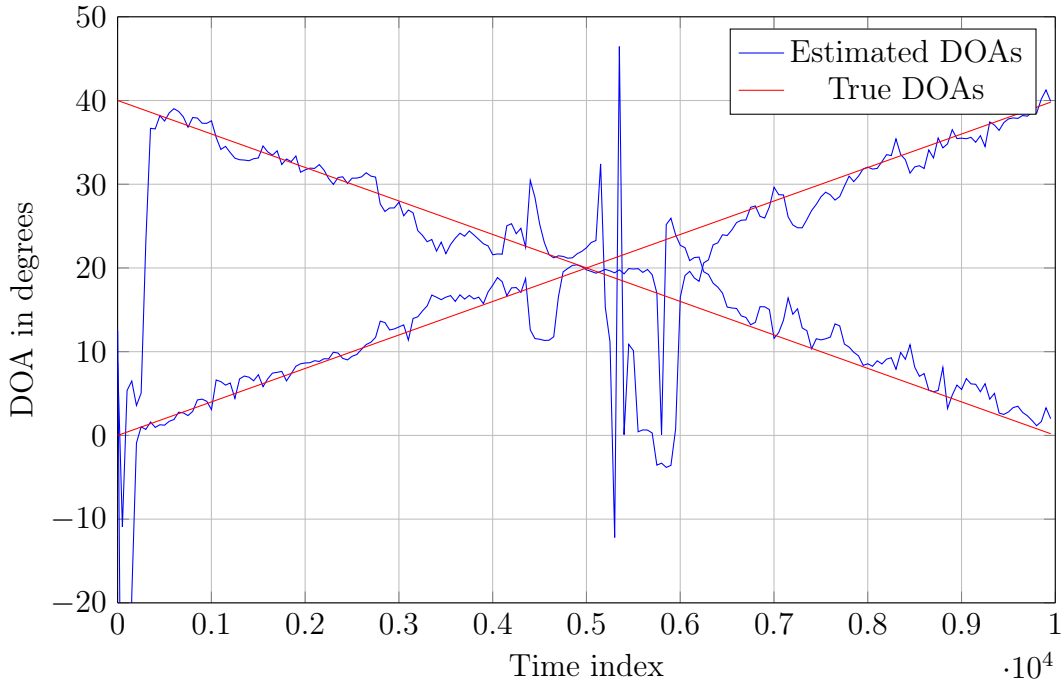

Fig. 4.4. Tracking moving sources.

4.3 Summary

In this chapter, shortcomings of the DPM for estimating the eigendecomposition of the sample covariance matrix have been addressed. The decentralized Lanczos method for eigendecomposition, which yields reduced communication costs compared to the DPM, has been introduced. To achieve decentralized online DOA estimation the decentralized generalized eigendecomposition has been proposed. An implementation of the ESPRIT algorithm in decentralized fashion using the AC protocol and the generalized eigendecomposition, which achieve DOA tracking, has been introduced. Similar to the previous chapter, the algorithms introduced in this chapter are only applicable for coherent processing, i.e., when the overall array covariance matrix is available. Non-coherent DOA estimation when only the subarrays covariance matrices are available, but not the inter-subarray covariance matrices, is presented in the following chapter.

Chapter 5
Non-coherent DOA Estimation

In this chapter, DOA estimation using non-coherent processing for partly calibrated arrays composed of multiple subarrays is considered. The subarrays are assumed to compute locally the sample covariance matrices of their measurements and communicate them to the processing center (PC). A sufficient condition for the unique identifiability of the sources in the aforementioned non-coherent processing scheme is presented. This conditions shows that, using non-coherent processing, the number of uniquely identifiable sources using multiple subarrays is larger than the number identifiable by the individual subarrays, unless the subarrays are identical. This property of non-coherent processing has not been investigated before. The Maximum Likelihood estimator (MLE) for DOA estimation at the PC using the sample covariance matrices received from the subarrays is derived. The CRB for the measurement model is derived and is used to assess the presented DOA estimators. The behaviour of the CRB at high SNR is analyzed. In contrast to coherent processing, it is proven that the CRB approaches zero at high SNR only if at least one subarray can identify the sources individually. Moreover, based on the Root-MUSIC algorithm, a low cost DOA estimation method is proposed for the case when all the subarrays can identify the sources. This chapter is based on the publications in [SP14] and [SPPZ17].

5.1 Introduction

Conventional DOA estimation algorithms reviewed in Chapter 2 and decentralized DOA estimation algorithms introduced in Chapter 3 and Chapter 4 perform coherent processing, i.e, they require the covariance matrix of the whole array including the inter-subarray covariance matrices[1].

Since, non-coherent processing techniques are carried out using only the subarray covariance matrices [SNS95], the largest available covariance lag[2] in non-coherent processing is generally the one corresponding to the subarray with the largest aperture, i.e., the subarray which possesses the largest inter-sensor distance. Whereas, in coherent processing, the largest available covariance lag corresponds to the whole array

[1]The sample covariance matrix is implicitly computed in the algorithms introduced in Chapter 3 and Chapter 4

[2]Covariance lags are the delays between copies of the signals received at different sensors.

aperture which is larger than that of the individual subarrays. Thus, the DOA estimation performance of non-coherent processing is inferior to that of coherent-processing, since the DOA estimation performance increases with the array aperture [KV96]. Nevertheless, non-coherent processing is preferred in large wireless sensor networks since it offers a huge reduction in the overall system complexity and in the communication overhead associated with communicating the raw subarray measurements to the PC or using the AC protocol as required in coherent processing. The computational load associated with non-coherent processing is also much smaller than that of the coherent processing, since only the small subarray covariance matrices are computed and not the large overall array covariance matrix. Thus, non-coherent processing is convenient for decentralized processing [SNS95]. Moreover, the computation of the inter-subarray covariance matrices in coherent processing requires synchronized subarrays, which is not always possible especially for widely separated subarrays [SNS95]. Hence, in large arrays, it is sometimes necessary to resort to non-coherent processing. In such cases, the measurements of each subarray are processed coherently, namely, the subarray covariance matrices are computed locally at the subarrays and communicated to the PC. Then, in the PC, non-coherent processing (using only local subarray covariance matrices) is carried out to perform the DOA estimation task.

In [WK85a, RF04], the MUSIC algorithm is generalized to non-coherent processing where it is assumed that the subarrays locally estimate their noise subspaces and send them to the PC. In [SS92], another version of the MUSIC algorithm for non-coherent processing is analyzed. In this algorithm, the subarrays send the locally estimated DOAs and their estimated variances to the PC. A similar method which is robust against uncertainties in the statistical distribution of the noise is presented in [LKM90]. In [SNS95], it is proposed to perform DOA estimation using the MODE algorithm individually in each subarray. At the PC, the DOA estimates are optimally combined as in [SS92]. Although the algorithms proposed in [WK85a, LKM90, SS92, SNS95, RF04] are designed for non-coherent processing, they all assume that each subarray can locally identify all the sources. The primary goal of this chapter is to overcome this restricting assumption and show that under mild conditions, using multiple non-coherent subarrays, the number of identifiable sources is larger than the number identifiable by each subarray individually.

In [SW99], direction finding using fewer receivers than the number of sources is introduced. Since only fewer receiver than the sources (and hence fewer than the sensors) are available, it is impossible to sample the output of all the sensors simultaneously. Thus, time varying processing is introduced where a different subset of the available sensors are sampled at each time period and their measurement covariance matrix is

computed. The DOA estimation problem in this context can be considered as a non-coherent processing DOA estimation problem, since the covariance matrices between different sensor subsets are not available. However, the authors of [SW99] assume a fully calibrated array, whereas this assumption is not made in this chapter. Moreover, the algorithms introduced in [SW99] perform an exhaustive search over the directions which is impractical when the number of sources is larger than two.

In this chapter, DOA estimation using non-coherent processing for partly calibrated arrays is considered for the case where none of the subarrays is able to identify all the sources locally. A bound on the maximum number of identifiable sources is presented. This bound shows that for particular array geometries, it is possible to identify more sources than each subarray can identify individually. Furthermore, two DOA estimation approaches are proposed: 1) the MLE and 2) a computationally simpler DOA estimation approach based on the SSR aproach. Moreover, the CRB for the considered measurement model is presented and analyzed. Finally, the case when all the subarrays can identify the sources is considered. An algorithm which outperforms the existing ones is introduced.

We remark for completeness that non-coherent processing-based DOA estimation approaches considered in this chapter and in [WK85a, LKM90, SS92, SNS95, RF04] differs from that of [KHE15] in the sense that in the approach of [KHE15], DOA estimation is achieved from magnitude only measurements. Thus, the approach of [KHE15] introduces ambiguities in the DOA estimates which have been resolved by assuming sources at known locations. However, the approach of [KHE15] assumes less information about the structure of the subarrays when compared to the approaches considered in this chapter and in [WK85a, LKM90, SS92, SNS95, RF04].

5.2 Signal Model

Similar to Section 2.1. A planar partly calibrated array composed of K perfectly calibrated subarrays is considered in this section. The kth subarray is comprised of M_k sensors, thus, the total number of sensors in the array is M defined in (2.1). Signals of L narrow-band far-field sources impinge onto the array from directions $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_L]^T$. The array geometry and thus the array manifold is defined as in Section 2.1. The subarray response for a source at direction θ, denoted as $\boldsymbol{a}_k(\theta)$ defined in (2.3), is partitioned into a subarray manifold vector $\boldsymbol{v}_k(\theta)$ defined in (2.4), which has a fully known description as a function of θ, and an unknown phase shift $\phi(\theta, \boldsymbol{\zeta}_k)$, which

depends on the unknown displacement $\boldsymbol{\zeta}_k$ of the kth subarray with respect to the reference sensor in the first subarray. The array topology is illustrated in Fig. 2.1.

The vector of the baseband signals received at the kth subarray is given by

$$\boldsymbol{x}_k(t-\tau_k) = \boldsymbol{A}_k(\boldsymbol{\theta}, \boldsymbol{\zeta}_k)\boldsymbol{s}(t-\tau_k) + \boldsymbol{n}_k(t-\tau_k) \tag{5.1}$$

where τ_k is the sampling offset at the kth subarray and $\boldsymbol{n}_k(t-\tau_k)$ is the vector containing the complex circular Gaussian sensor noise with zero-mean and variance σ^2. The vector $\boldsymbol{s}(t-\tau_k)$ contains the complex circular Gaussian source signals with zero-mean and covariance $\boldsymbol{P}$. The statistical properties of the sources observed by different subarrays are assumed to be identical, i.e.,

$$\boldsymbol{P} = \mathbb{E}\big(\boldsymbol{s}(t-\tau_k)\boldsymbol{s}^H(t-\tau_k)\big), \tag{5.2}$$

for $k = 1, \ldots, K$. Note that the difference between (2.5) for the coherent processing case and (5.1) for the non-coherent processing case is the offset τ_k. That means in coherent processing at time t all the subarrays observe the same source signal, whereas this is not true in non-coherent processing. However, in both cases (coherent and non-coherent processing) the same source correlation structure $\boldsymbol{P}$ are observed at all subarrays. In the following, the dependency on $\boldsymbol{\theta}$ and $\boldsymbol{\zeta}_k$ is dropped for notation convenience.

The true measurement covariance matrix of the kth subarray is written as

$$\boldsymbol{R}_k = \mathbb{E}\big(\boldsymbol{x}_k(t-\tau_k)\boldsymbol{x}_k^H(t-\tau_k)\big) = \boldsymbol{V}_k\boldsymbol{\Phi}_k\boldsymbol{P}\boldsymbol{\Phi}_k^H\boldsymbol{V}_k^H + \sigma^2\boldsymbol{I}_{M_k}, \tag{5.3}$$

where (2.7) is used, the $M_k \times M_k$ identity matrix is denoted by $\boldsymbol{I}_{M_k}$, $\boldsymbol{V}_k$ is define in (2.8), $\boldsymbol{\Phi}_k$ is define in (2.9), and $\boldsymbol{P}$ is defined in (5.2).

The sample estimate of $\boldsymbol{R}_k$ is computed using N snapshots of the kth subarray output as done for the whole array in (2.15)[3]. Without loss of generality, the same number of samples N is assumed to be available at all subarrays.

In non-coherent processing, it is assumed that the subarrays send their locally estimated sample covariance matrices $\hat{\boldsymbol{R}}_k$, for $k = 1, \ldots, K$, to the PC[4], which carries out the DOA estimation algorithm [SNS95]. This processing type is referred to as *non-coherent processing*, since only the local subarray covariance matrices are available at the PC. Compared to coherent processing where the sample estimate of the cross-subarrays covariance matrices, i.e., $\mathbb{E}[\boldsymbol{x}_k(t)\boldsymbol{x}_i^H(t)]$, for $i \neq k$, $i, k =, 1 \ldots, K$, are available at the

[3] $\boldsymbol{x}_k(t)$ is used instead of $\boldsymbol{x}$(t) in (2.15) to compute $\hat{\boldsymbol{R}}_k$

[4] This requires sending M_k^2 real numbers to the PC, instead of $2NM_k$ in the case of sending raw measurements.

PC[5]. This requires a synchronized subarray system, i.e., $\tau_k = 0$ for $k = 1, \ldots, K$. Observe that:

- In non-coherent processing, the resolution capability of the array is limited, compared to coherent processing, since generally the largest available covariance lag corresponds to the largest subarray. Whereas, in coherent processing, the largest available covariance lag corresponds generally to the array aperture.
- The non-coherent processing scheme is more suitable for decentralized processing than the coherent processing one, since each subarray can act as a decentralized processing node which computes the local covariance matrix of the subarray and sends it to the PC. Whereas, in coherent processing, the computation of the cross-subarray covariance matrices requires either sending the raw measurement to the PC or the use of the AC protocol, i.e., it involves a much larger communication overhead compared to non-coherent processing, refer to Section 2.4.2.

5.3 DOA Estimation for Uncorrelated Sources

In this section, the special case of perfectly uncorrelated sources is considered for which the structure of the covariance matrix introduced in (5.3) can be simplified. The identifiability of the non-coherent model is analyzed in this case. The CRB and the MLE are derived. Moreover, DOA estimation using SSR is presented.

Under the assumption of uncorrelated sources, the source covariance matrix $\boldsymbol{P}$ in (5.2) is diagonal. Since the matrix $\boldsymbol{\Phi}_k$, defined in (5.42), is also diagonal with unit amplitude entries, one can write

$$\boldsymbol{\Phi}_k \boldsymbol{P} \boldsymbol{\Phi}_k^H = \boldsymbol{P} = \operatorname{diag}(\boldsymbol{p}), \tag{5.4}$$

where the vector $\boldsymbol{p} = [p_1, \ldots, p_L]^T$ is the diagonal of $\boldsymbol{P}$. Substituting (5.4) in (5.3) yields

$$\boldsymbol{R}_k = \boldsymbol{V}_k \boldsymbol{P} \boldsymbol{V}_k^H + \sigma^2 \boldsymbol{I}_{M_k}. \tag{5.5}$$

The vectorization of the product of three matrices $\boldsymbol{M}_1, \boldsymbol{M}_2$, and $\boldsymbol{M}_3$ of appropriate sizes can be written as [Gra81]

$$\operatorname{vec}\big(\boldsymbol{M}_1 \boldsymbol{M}_2 \boldsymbol{M}_3\big) = (\boldsymbol{M}_3^T \otimes \boldsymbol{M}_1) \operatorname{vec}(\boldsymbol{M}_2). \tag{5.6}$$

[5]Note that in the model (5.1), the computation of $\mathbb{E}[\boldsymbol{x}_k(t-\tau_k)\boldsymbol{x}_i^H(t-\tau_i)]$ when $k \neq i$ yields a covariance of zero if $|\tau_k - \tau_i|$ exceeds the coherence time of the signal waveform such that $\mathbb{E}\big(\boldsymbol{s}(t-\tau_k)\boldsymbol{s}^H(t-\tau_i)\big) = \boldsymbol{0}_L \boldsymbol{0}_L^T$.

Denote as $\boldsymbol{r}_k = \mathrm{vec}(\boldsymbol{R}_k)$ the vectorization of the kth subarray measurement covariance matrix. Then, substituting (5.5) and (5.6) in $\boldsymbol{r}_k$ yields

$$\boldsymbol{r}_k = \big(\boldsymbol{V}_k^* \otimes \boldsymbol{V}_k\big)\mathrm{vec}(\boldsymbol{P}) + \sigma^2 \boldsymbol{i}_k, \tag{5.7}$$

where $\boldsymbol{i}_k = \mathrm{vec}(\boldsymbol{I}_{M_k})$. Since $\boldsymbol{P}$ is a diagonal matrix, (5.7) is further reduced to

$$\boldsymbol{r}_k = \breve{\boldsymbol{V}}_k \boldsymbol{p} + \sigma^2 \boldsymbol{i}_k, \tag{5.8}$$

where the vector $\boldsymbol{p}$ is defined in (5.4) and the $M_k^2 \times L$ matrix

$$\breve{\boldsymbol{V}}_k = \big(\boldsymbol{V}_k^* \circ \boldsymbol{V}_k\big), \tag{5.9}$$

where $\circ$ denotes the Katri-Rao product, contains the columns of the matrix $\big(\boldsymbol{V}_k^* \otimes \boldsymbol{V}_k\big)$ corresponding to the diagonal of $\boldsymbol{P}$. The matrix $\breve{\boldsymbol{V}}_k$ is referred to as the co-subarray manifold[6]. The concatenation of all vectorized measurement covariance matrices is written as

$$\boldsymbol{r} = [\boldsymbol{r}_1^T, \ldots, \boldsymbol{r}_K^T]^T, \tag{5.10}$$

where $\boldsymbol{r}$ is of size $\breve{M} = \sum_{k=1}^{K} M_k^2$. By substituting (5.8) in (5.10), the vector $\boldsymbol{r}$ becomes

$$\boldsymbol{r} = \breve{\boldsymbol{V}} \boldsymbol{p} + \sigma^2 \boldsymbol{i}, \tag{5.11}$$

where

$$\breve{\boldsymbol{V}} = [\breve{\boldsymbol{V}}_1^T, \ldots, \breve{\boldsymbol{V}}_K^T]^T \tag{5.12}$$

is the co-array manifold and

$$\boldsymbol{i} = [\boldsymbol{i}_1^T, \ldots, \boldsymbol{i}_K^T]^T. \tag{5.13}$$

Denote as $\hat{\boldsymbol{r}}$ and $\hat{\boldsymbol{r}}_k$, for $k = 1, \ldots, K$, the sample estimate of $\boldsymbol{r}$ and $\boldsymbol{r}_k$, respectively,

5.3.1 Identifiability

In this subsection, first the condition of parameter identifiability as introduced in [HN96] is revised, then a sufficient condition on the maximum number of identifiable (uncorrelated) sources is presented.

Let $\boldsymbol{\theta'} = [\theta'_1, \ldots, \theta'_L]^T$ and $\boldsymbol{\theta''} = [\theta''_1, \ldots, \theta''_L]^T$ denote two vectors each of them containing L pairwise-different DOAs. Where, pairwise-different DOA vector $\boldsymbol{\theta'}$ means that $\theta'_i \neq \theta'_j$ for $i \neq j$ and $i, j = 1, \ldots, L$. Further, the notation $\boldsymbol{\theta'} \not\equiv \boldsymbol{\theta''}$ expresses that there exist an index $i \leq L$ where for all $j \leq L$, $\theta'_i \neq \theta''_j$. In other words, at least one entry of $\boldsymbol{\theta'}$ is not equal to any entry of $\boldsymbol{\theta''}$. In the following, the definition of identifiability [HN96] is presented.

[6]The expression co-array manifold has been used in [ASG99] in the context of nonuniform linear antenna arrays to denote the Katri-Rao product of the conjugate array response with itself.

Definition 1 (Identifiability). *In the noise free case, L sources with DOAs $\boldsymbol{\theta}$ and powers $\boldsymbol{p}$ are uniquely identifiable if*

$$\breve{\boldsymbol{V}}(\boldsymbol{\theta})\boldsymbol{p} \neq \breve{\boldsymbol{V}}(\boldsymbol{\theta}')\boldsymbol{p}', \tag{5.14}$$

for any vector with positive entries $\boldsymbol{p}'$ and for any pairwise-different DOA vector $\boldsymbol{\theta}'$, where $\boldsymbol{\theta} \not\equiv \boldsymbol{\theta}'$.

Note that in the noise free case, the product $\breve{\boldsymbol{V}}(\boldsymbol{\theta})\boldsymbol{p}$ consist in the vectorized measurement covariances, i.e., $\boldsymbol{r} = \breve{\boldsymbol{V}}(\boldsymbol{\theta})\boldsymbol{p}$. Let $\mathcal{P}(\boldsymbol{x}(t)|\boldsymbol{\theta})$ denotes the distribution of the array measurements for a particular source directions $\boldsymbol{\theta}$. Since the subarray measurements follows a zero mean Gaussian distribution with (vectorized) covariances $\boldsymbol{r}$, Definition 1 implies that, the direction of the sources are uniquely identifiable if two parameter vectors $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$, where $\boldsymbol{\theta} \not\equiv \boldsymbol{\theta}'$, yield different measurement distributions, i.e., $\mathcal{P}(\boldsymbol{x}(t)|\boldsymbol{\theta}) \neq \mathcal{P}(\boldsymbol{x}(t)|\boldsymbol{\theta}')$ for $\boldsymbol{\theta} \not\equiv \boldsymbol{\theta}'$ [HN96].

Let ρ denotes the Kruskal rank [SS07, Kru77] of the co-array manifold matrix $\breve{\boldsymbol{V}}$, i.e., ρ is the largest integer such that the columns of the matrix $\breve{\boldsymbol{V}}([\theta_1, \ldots, \theta_\rho]^T)$ are linearly independent for any vector $[\theta_1, \ldots, \theta_\rho]^T$ with pairwise different DOAs. Based on ρ, we provide a sufficient condition for the unique identifiability of L sources in the following theorem.

Theorem 4 (Sufficient condition for identifiability). *The L DOAs $\boldsymbol{\theta}$ can be uniquely identified from covariances $\boldsymbol{r} = \breve{\boldsymbol{V}}\boldsymbol{p}$ provided that*

$$L \leq \lfloor \frac{\rho}{2} \rfloor, \tag{5.15}$$

where ρ is the Kruskal rank of the co-array manifold $\breve{\boldsymbol{V}}$.

Proof. See Appendix A.4. □

Denote by $\breve{\boldsymbol{b}}_{k,i,j}$ the (i,j)th covariance lag of the kth subarray, i.e., $\breve{\boldsymbol{b}}_{k,i,j} = \boldsymbol{\zeta}'_{k,j} - \boldsymbol{\zeta}'_{k,i}$, where $\boldsymbol{\zeta}'_{k,i}$ is the location of the ith sensor of the kth subarray with respect to its first sensor (refer to Section 2.1), and let $\breve{\mathcal{B}}_k$ denotes the set of all different covariance lags of the kth subarray, i.e.,

$$\breve{\mathcal{B}}_k = \{\breve{\boldsymbol{b}}_{k,i,j}, i,j = 1, \ldots, M_k\}. \tag{5.16}$$

Further, let $\breve{\mathcal{B}}$ denotes the set of different covariance lags of the whole array, i.e.,

$$\breve{\mathcal{B}} = \bigcup_{k=1}^{K} \breve{\mathcal{B}}_k. \tag{5.17}$$

The Kruskal rank ρ of the matrix $\breve{\boldsymbol{V}}$ is bounded by the number of covariance lags in the set $\breve{\mathcal{B}}$. This observation yields the following result.

Corollary 1. *The number of sources which can be uniquely identified from covariances* $\boldsymbol{r}$ *is smaller than* $\lfloor \mathrm{card}(\breve{\mathcal{B}})/2 \rfloor$*, where* $\mathrm{card}(\breve{\mathcal{B}})$ *is the cardinality of the set* $\breve{\mathcal{B}}$*.*

Corollary 1 implies that the number of uniquely identifiable sources using non-coherent processing can be increased by designing the subarrays with different covariance lags. Note that in the special case where all subarrays admit the same covariance lags, e.g., if the subarrays are identical, then the number of uniquely identifiable sources by the whole array is not larger than the number identifiable by one individual subarray. The following example provides further insight.

Example

Consider an array composed of $K = 3$ identically oriented linear subarrays where the kth subarray includes $M_k = 2$ sensors. The relative positions between the successive sensors in the subarrays are assumed to be $d_1 = 1$, $d_2 = 2$ and $d_3 = 3$ half-wavelength, respectively, see Fig. 5.1. For coherent processing the maximum number of identifiable sources using this array is generally $M - K = 3$ (see [PGW02]). Note that coherent processing scenario represents an upper bound on the number of uniquely identifiable sources using non-coherent processing, since more covariance lags are available for coherent processing, namely, the covariance lags corresponding to the relative position of two sensors belonging to different subarrays. Thus, $L \leq 3$ is a necessary condition for identifying the sources using non-coherent processing. In the following, based on Theorem 4, it is shown that $L \leq 3$ is a sufficient condition for identifying the sources in the considered array example.

The subarray response vectors in (2.4) are reduced to $\boldsymbol{v}_k(\theta) = [1, e^{\jmath d_k \pi \sin\theta}]^T$, for $k = 1, \ldots, 3$, in this example. Thus, the matrix $\breve{\boldsymbol{V}}$ has the same rank as the matrix

$$\breve{\boldsymbol{M}} = \begin{pmatrix} e^{-3\jmath\pi \sin\theta_1} & \ldots & e^{-3\jmath\pi \sin\theta_L} \\ e^{-2\jmath\pi \sin\theta_1} & \ldots & e^{-2\jmath\pi \sin\theta_L} \\ \vdots & \vdots & \vdots \\ e^{3\jmath\pi \sin\theta_1} & \ldots & e^{3\jmath\pi \sin\theta_L} \end{pmatrix}, \tag{5.18}$$

where rows are rearranged and duplicated rows are deleted from $\breve{\boldsymbol{V}}$ to get $\breve{\boldsymbol{M}}$. The matrix $\breve{\boldsymbol{M}}$ is a Vandermonde matrix with 7 rows. Consequently, $\rho = 7$ and $\lfloor \frac{\rho}{2} \rfloor = 3$, i.e., up to $L = 3$ sources can be identified assuming non-coherent processing in this example. Thus, regarding identifiability non-coherent processing is equivalent to coherent processing in this scenario. Moreover, observe that where each subarray is able to identify one source locally (since each subarray consists of 2 sensors [WZ89]), using

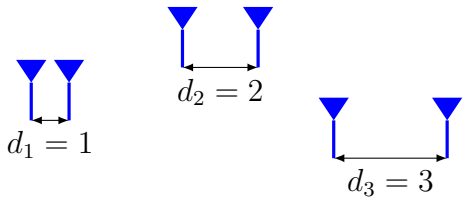

Fig. 5.1. Array composed of $K = 3$ subarrays.

non-coherent processing, the number of identifiable sources is increased up to $L = 3$ sources. This increase results from the fact that the three subarrays have different covariance lags.

5.3.2 The Maximum Likelihood Estimator

In Section 2.2.2, the MLE for coherent arrays is reviewed. In this section, the MLE for DOA estimation using non-coherent processing is derived considering uncorrelated sources.

In the scenario considered in this chapter, the PC receives the sample covariance matrices from the subarrays. These matrices follow a Wishart distribution [SS10, p. 49] with probability density function

$$\mathcal{P}(\hat{\boldsymbol{R}}_k) = \frac{|N\hat{\boldsymbol{R}}_k|^{N-M_k}}{\Gamma^c_{M_k}(N)|\boldsymbol{R}_k|^N} \exp\left(-N\mathrm{tr}\left(\boldsymbol{R}_k^{-1}\hat{\boldsymbol{R}}_k\right)\right) \tag{5.19}$$

where $\Gamma^c_{M_k}(N) = \pi^{M_k(M_k-1)/2} \prod_{i=1}^{M_k} \prod_{j=1}^{N-i} j$ and $\boldsymbol{R}_k$ is given in (5.3). Ignoring the constant term in (5.19), the negative log-likelihood function is written as[7]

$$\mathcal{L}(\boldsymbol{R}_1, \ldots, \boldsymbol{R}_K) = \sum_{k=1}^{K} N\left(\log|\boldsymbol{R}_k| + \mathrm{tr}\left(\boldsymbol{R}_k^{-1}\hat{\boldsymbol{R}}_k\right)\right). \tag{5.20}$$

The function $\mathcal{L}(\boldsymbol{R}_1, \ldots, \boldsymbol{R}_K)$ is valid under the assumption of correlated sources as well as uncorrelated sources. Where only the structure of the measurement covariance matrices $\boldsymbol{R}_1, \ldots, \boldsymbol{R}_K$ depends on the source correlations. For uncorrelated sources the measurement covariance matrix of the kth subarray $\boldsymbol{R}_k$ reduces to (5.5), i.e., $\boldsymbol{R}_k$ depends on the DOAs $\boldsymbol{\theta}$, the source powers $\boldsymbol{p}$, and the noise variance σ^2. Thus, the

[7] A concentrated expression of the MLE for non-coherent processing, similar to that in (2.22) for coherent and fully calibrated arrays, does not exist, since we assumed that the subarrays can not estimate the sources individually [SW99]

DOAs, the power of the sources, and the noise variance are estimated by solving the minimization problem

$$\begin{aligned} &\min_{\boldsymbol{\theta},\boldsymbol{p},\sigma^2} \mathcal{L}(\boldsymbol{\theta},\boldsymbol{p},\sigma^2) \\ &\text{s.t.} \quad \boldsymbol{p} > \mathbf{0}_L, \\ &\qquad \sigma^2 > 0. \end{aligned} \tag{5.21}$$

The function $\mathcal{L}(\boldsymbol{\theta},\boldsymbol{p},\sigma^2)$ in (5.21) is nonconvex [BV04]. Therefore, a good initial solution is essential for the MLE.

5.3.3 The Cramér-Rao Bound

The derivation of the CRB follows, as for the coherent case [SLG01], from the Fisher information matrix (FIM). Since for non-coherent processing the measurements of different subarrays are not correlated, the FIM matrix for this scenario can be written as [SW99]

$$\text{FIM} = \sum_{k=1}^{K} \text{FIM}_k. \tag{5.22}$$

Using (5.22) and following the steps of [SW99,SWW97], the CRB corresponding to the direction parameters $\boldsymbol{\theta}$ can be written as

$$\text{CRB}_{\boldsymbol{\theta}} = \left(\boldsymbol{\Delta}_1^H \left(\breve{\boldsymbol{R}} - \breve{\boldsymbol{R}}\boldsymbol{\Delta}_2 \left(\boldsymbol{\Delta}_2^H \breve{\boldsymbol{R}} \boldsymbol{\Delta}_2 \right)^{-1} \boldsymbol{\Delta}_2^H \breve{\boldsymbol{R}} \right) \boldsymbol{\Delta}_1 \right)^{-1}. \tag{5.23}$$

where

$$\boldsymbol{\Delta}_1 = [\frac{d\boldsymbol{r}}{d\boldsymbol{\theta}^T}], \quad \boldsymbol{\Delta}_2 = [\frac{d\boldsymbol{r}}{d\boldsymbol{p}^T}, \frac{d\boldsymbol{r}}{d\sigma^2}], \tag{5.24}$$

are the matrices which represent the derivatives of $\boldsymbol{r}$ with respect to $\boldsymbol{\theta}$, $\boldsymbol{p}$, and σ^2, respectively,

$$\breve{\boldsymbol{R}} = \text{blkdiag}\left(\breve{\boldsymbol{R}}_1, \ldots, \breve{\boldsymbol{R}}_K \right), \tag{5.25}$$

and $\breve{\boldsymbol{R}}_k = N(\boldsymbol{R}_k^{-T} \otimes \boldsymbol{R}_k^{-1})$. In the sequel, the behaviour of CRB at high SNR is demonstrated by simulation then it is analyzed.

Consider the following two scenarios:

S1: $M_1 = \cdots = M_K \leq L$, i.e., the FIM for each individual subarray is not invertible, whereas the overall FIM, defined in (5.22), is invertible.

S2: $M_1 > L$ and $M_k \leq L$, for $k = 2, \ldots, K$, i.e., the FIM of the first subarray FIM_1 is invertible whereas the FIM of the remaining subarrays, i.e., FIM_k, for $k = 2, \ldots, K$ are not invertible.

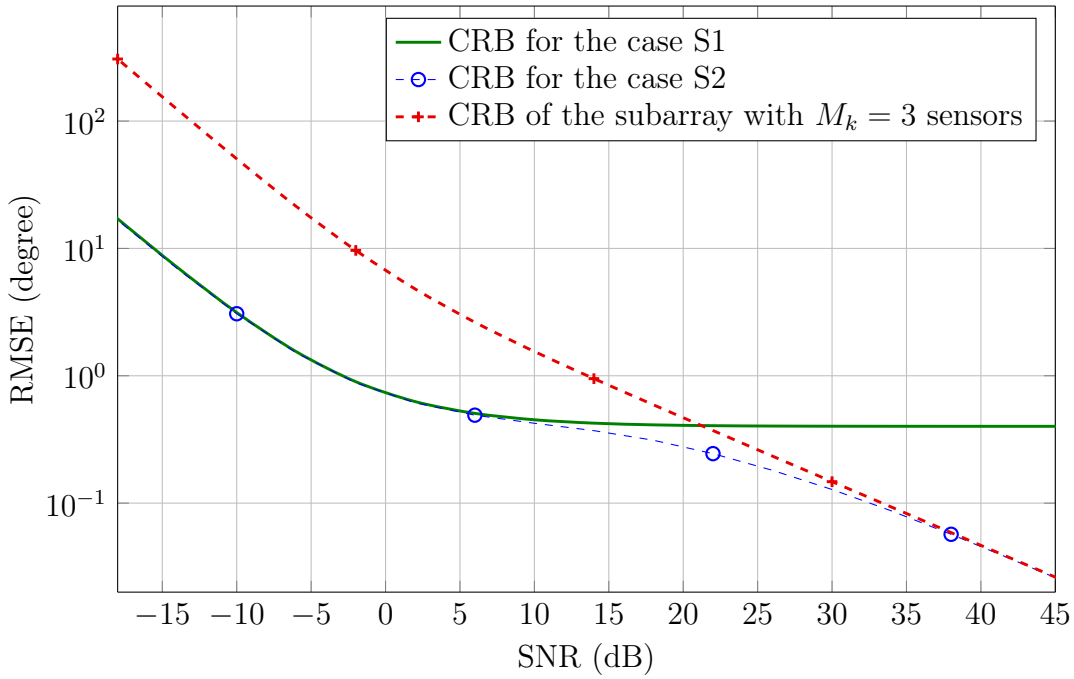

Fig. 5.2. The CRB for the cases where 1) none of the subarrays is able to identify the sources individually 2) one subarray can identify the sources. Also the CRB for the subarray with 3 sensors in case 2) is shown.

In Fig. 5.2, the CRB for $K = 12$ subarrays and $L = 2$ uncorrelated equal-powered sources is displayed for two array configurations which represent the aforementioned scenarios S1 and S2[8]. The two configurations are:

1. $M_1 = \cdots = M_K = 2 = L$, which represents S1.
2. $M_1 = 3 > L$ and $M_2 = \cdots = M_{K-1} = 2 = L$, which represents S2.

Moreover, in Fig. 5.2, the CRB of the first subarray with $M_1 = 3$ sensors in the second configuration is displayed. It can be observed from Fig. 5.2 that in the scenario S1, the CRB does not approach zero as the SNR approaches infinity rather it remains unchanged at high SNR (in Fig. 5.2, the CRB remains almost unchanged for SNR above 10 dB). In the scenario S2, the CRB is almost identical to that of the scenario S1 when the SNR is less than 10 dB. However, it continues to decrease for SNR larger than 10 dB and the performance at high SNR in this case is determined by the performance of the first subarray. Thus, at high SNR, DOA estimation can be performed using only those subarrays which are able to identify and estimate the DOAs individually,

[8]For the details on the array geometry parameters please refer the array setup described in Section 5.5.

if such subarrays exist. In [LKM90], the authors suggested to include only subarrays which can individually identify all the sources in the DOA estimation algorithm. This approach is justified at high SNR, however, at low SNR using all the subarray yields the better estimation performance, as demonstrated by the CRB in Fig. 5.2.

In the following, the aforementioned behaviour of the CRB is analyzed at high SNR in the two scenarios S1 and S2. Thus, let L uncorrelated equally-powered sources in the high SNR region, i.e., $p_1 = \cdots = p_L = p$, where $p_1, \ldots, p_L$ are the power of the sources whose directions are denoted by $\theta_1, \ldots, \theta_L$, respectively, and $p \gg \sigma^2$, refer to (5.5). Further, let $\boldsymbol{r}|_{p\gg\sigma} = p\left(\breve{\boldsymbol{V}}\mathbf{1}_L + \frac{\sigma^2}{p}\boldsymbol{i}\right)|_{p\gg\sigma} \approx p\breve{\boldsymbol{V}}\mathbf{1}_L$ denotes the high SNR approximation of the vectorized covariance matrices. Consequently, the derivative matrices $\boldsymbol{\Delta}_1$ and $\boldsymbol{\Delta}_2$ in (5.24) reduce to $\boldsymbol{\Delta}_1|_{p\gg\sigma} = p[\frac{d(\breve{\boldsymbol{V}})\mathbf{1}_L}{d\theta_1}, \ldots, \frac{d(\breve{\boldsymbol{V}})\mathbf{1}_L}{d\theta_L}]$ and $\boldsymbol{\Delta}_2|_{p\gg\sigma} = [\breve{\boldsymbol{V}}\mathbf{1}_L, \boldsymbol{i}]$. Similarly, denote $\breve{\boldsymbol{R}}$ at high SNR by $\breve{\boldsymbol{R}}|_{p\gg\sigma} \approx p^{-2}N\overline{\boldsymbol{V}}$, where $\overline{\boldsymbol{V}} = \text{blkdiag}\left(\overline{\boldsymbol{V}}_1, \ldots, \overline{\boldsymbol{V}}_K\right)$ and $\overline{\boldsymbol{V}}_k = (\boldsymbol{V}_k\boldsymbol{V}_k^H)^{-T} \otimes (\boldsymbol{V}_k\boldsymbol{V}_k^H)^{-1}$. Substituting $\boldsymbol{\Delta}_1|_{p\gg\sigma}$, $\boldsymbol{\Delta}_2|_{p\gg\sigma}$, and $\breve{\boldsymbol{R}}|_{p\gg\sigma}$ in (5.23), the CRB in the high SNR region reduces to

$$\text{CRB}_{\boldsymbol{\theta}}^{-1}|_{p\gg\sigma} \approx N[\frac{d(\breve{\boldsymbol{V}})\mathbf{1}_L}{d\theta_1}, \ldots, \frac{d(\breve{\boldsymbol{V}})\mathbf{1}_L}{d\theta_L}]^H\, \overline{\boldsymbol{V}}\, [\frac{d(\breve{\boldsymbol{V}})\mathbf{1}_L}{d\theta_1}, \ldots, \frac{d(\breve{\boldsymbol{V}})\mathbf{1}_L}{d\theta_L}]. \tag{5.26}$$

Interestingly, observe from (5.26) that at high SNR, the expression for $\text{CRB}_{\boldsymbol{\theta}}^{-1}|_{p\gg\sigma}$ depend neither on p nor on σ^2 but only on the DOAs $\theta_1, \ldots, \theta_L$. Next, consider how the expression for $\text{CRB}_{\boldsymbol{\theta}}^{-1}|_{p\gg\sigma}$ changes in the two scenarios S1 and S2.

Let $\overline{\rho}_k$ denote the rank of the matrix $\overline{\boldsymbol{V}}_k^{-1}$. Since the rank of the Kronecker product is the product of the ranks of its operand matrices [AM05], the rank $\overline{\rho}_k$, for $k = 1, \ldots, K$, takes the value $\overline{\rho}_k = M_k^2$ in both scenarios S1 and S2[9], except for the first subarray with sensors larger than L in S2 whose rank is $L^2 < M_1^2$. Thus, the following behaviour of the block diagonal matrix $\overline{\boldsymbol{V}}^{-1}$ is observable:

- In the scenario S1, $\overline{\boldsymbol{V}}^{-1}$ is full rank.
- In the scenario S2, $\overline{\boldsymbol{V}}^{-1}$ is rank deficient. More precisely, the first block of $\overline{\boldsymbol{V}}^{-1}$, which corresponds to the first subarray is rank deficient.

Consequently, in the scenario S1, the matrix $\overline{\boldsymbol{V}}$ has finite entries (and eigenvalues) leading to a finite non-zero CRB. Whereas, in the scenario S2, the matrix $\overline{\boldsymbol{V}}$ has infinitely large eigenvalues which drive the corresponding CRB to zero. Moreover, in the scenario S2, $\text{CRB}_{\boldsymbol{\theta}}^{-1}|_{p\gg\sigma}$ in (5.26) can be approximated by

[9] Using the well-known inversion identity $(\boldsymbol{A} \otimes \boldsymbol{B})^{-1} = \boldsymbol{A}^{-1} \otimes \boldsymbol{B}^{-1}$ [AM05].

$N[\frac{d(\breve{\boldsymbol{V}}_1)\mathbf{1}_L}{d\theta_1},\ldots,\frac{d(\breve{\boldsymbol{V}}_1)\mathbf{1}_L}{d\theta_L}]^H\ \overline{\boldsymbol{V}}_1\ [\frac{d(\breve{\boldsymbol{V}}_1)\mathbf{1}_L}{d\theta_1},\ldots,\frac{d(\breve{\boldsymbol{V}}_1)\mathbf{1}_L}{d\theta_L}]$ since the entries of $\overline{\boldsymbol{V}}_2,\ldots,\overline{\boldsymbol{V}}_K$ are negligible compared to the entries of $\overline{\boldsymbol{V}}_1$. This means that in the scenario S2, at high SNR, the CRB of the whole array can be approximated by the CRB of the first subarray, which is observable in Fig. 5.2 for SNR larger than 20 dBs. A behaviour of the CRB similar to that of scenario S1 at high SNR has been observed in [SW99, SWW97] for DOA estimation using fewer receivers and "it is shown to be typical in scenarios where a signal subspace is nonexistent". However, in [SW99, SWW97] the scenario S2 has not been considered. Moreover, in [AGGS98, Fig. 1], a similar behaviour to the scenario S1 is observed in DOA estimation using fully augmentable sparse linear arrays when the number of sources is larger than the number of the sensors in the array but smaller than the available covariance lags.

Regarding the number of samples N, note that the CRB approaches zero in both scenarios S1 and S2 when N approaches infinity, as it can be observed from (5.26).

5.3.4 DOA Estimation Using Sparse Signal Representation

The SPICE optimization problem, reviewed in Section 2.2.3.2, is presented in [SBL11] for fully calibrated arrays. In the SPICE algorithm, DOA estimation is formulated as a convex optimization problem, which can be efficiently solved. In the sequel, the SPICE optimization problem is rewritten for the non-coherent processing scheme, considered in this section, using partly calibrated arrays. The difference between both SPICE formulations is highlighted. For the non-coherent processing scenario in the case of perfectly uncorrelated sources, the SPICE optimization problem is written as

$$\min_{\tilde{\boldsymbol{P}},\sigma} \quad \sum_{k=1}^{K} \mathrm{tr}\big(\tilde{\boldsymbol{R}}_k^{-1}\hat{\boldsymbol{R}}_k\big) \tag{5.27a}$$

$$\text{subject to} \quad \tilde{p}_i \geq 0, i=1,\ldots,G,\ \sigma \geq 0, \tag{5.27b}$$

$$\sum_{g=1}^{G} \omega_g \tilde{p}_g + \bar{\omega}\sigma^2 = 1 \tag{5.27c}$$

where $\tilde{\boldsymbol{R}}_k = \tilde{\boldsymbol{V}}_k\tilde{\boldsymbol{P}}\tilde{\boldsymbol{V}}_k^H + \sigma^2\boldsymbol{I}_{M_k}$, the $\breve{M}_k \times G$ overcomplete dictionary $\tilde{\boldsymbol{V}}_k$ is defined as

$$\tilde{\boldsymbol{V}}_k = [\boldsymbol{v}_k(\tilde{\theta}_1),\ldots,\boldsymbol{v}_k(\tilde{\theta}_G)], \tag{5.28}$$

the DOA grid $\tilde{\boldsymbol{\theta}}$ is defined in (2.24) and the sparse diagonal matrix $\tilde{\boldsymbol{P}}$ is defined in (2.32). The weights in (2.33c), for the non-coherent processing scenario, are written as

$$\omega_g = \frac{1}{M}\sum_{k=1}^{K} \boldsymbol{v}_k^H(\tilde{\theta}_g)\hat{\boldsymbol{R}}_k^{-1}\boldsymbol{v}_k(\tilde{\theta}_g), \tag{5.29}$$

and

$$\bar{\omega} = \frac{1}{M} \sum_{k=1}^{K} \mathrm{tr}\big(\hat{\boldsymbol{R}}_k^{-1}\big). \tag{5.30}$$

The basic difference between the original SPICE in (2.33) for coherent and fully calibrated arrays [SBL11] and the SPICE version for non-coherent processing using partly calibrated arrays, derived in this section, are:

1. For non-coherent processing, the overcomplete dictionaries $\tilde{\boldsymbol{V}}_k$, for $k = 1, \ldots, K$, are defined based on the subarray response $\boldsymbol{v}(\theta)$, whereas, for coherent processing, the overcomplete dictionary $\tilde{\boldsymbol{A}}$ is defined using the whole array response.

2. In non-coherent processing, the objective function in (5.27a) and the weights computation in (5.29) and (5.30) are written using a summation over the subarrays, whereas the corresponding definitions for coherent processing are defined for the whole array.

These differences appears since the SPICE approach in Section 2.2.3.2 is introduced for coherent and fully calibrated arrays. Thus, in this case, the array response is known as a function of θ and the overall array sample covariance matrix can be computed, which is not the case for non-coherent processing using partly calibrated arrays.

Problem (5.27) is semi-definite program [SBL11], thus it can be solved using, e.g., cvx [CR12]. Note that using SSR, the DOA estimation problem is reduced to the identification of the non-zero elements of the estimated diagonal sparse matrix $\hat{\tilde{\boldsymbol{P}}}$. The DOA estimates are the grid points, i.e., the elements of $\tilde{\boldsymbol{\theta}}$, which correspond to the L largest peaks of the diagonal of $\hat{\tilde{\boldsymbol{P}}}$, refer to Section 2.2.3.2.

5.4 Extension to Correlated Sources

In the previous section, it is assumed that the sources impinging onto the system of subarrays are uncorrelated. In that case, the phase matrices $\boldsymbol{\Phi}_k$ and $\boldsymbol{\Phi}_k^H$ cancels out, for $k = 1, \ldots, K$, as shown in (5.4). However, by dropping the assumption of uncorrelated sources, (5.4) is no longer valid, i.e., the phases matrices does not cancel out, since the matrix $\boldsymbol{P}$ is not diagonal. In this section, the MLE, the SSR approach, and the CRB, which have been introduced in the previous section for the case of uncorrelated sources, are extended to the case of correlated sources.

5.4.1 The MLE and SSR approaches for Correlated Sources

The derivation of the MLE in the correlated sources case is carried out as in the case of uncorrelated sources, which is introduced in Section 5.3.2. However, since in this case the phases matrices $\boldsymbol{\Phi}_k$ and $\boldsymbol{\Phi}_k^H$, for $k = 1, \ldots, K$, does not cancel out, the measurement covariance matrix $\boldsymbol{R}_k$ in (5.3) is used to derive the MLE. The negative log-likelihood in the presence of correlated sources, denoted as $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{P}, \sigma^2, \boldsymbol{\Phi}_2, \ldots, \boldsymbol{\Phi}_K)$, is defined in (5.20). However, for $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{P}, \sigma^2, \boldsymbol{\Phi}_2, \ldots, \boldsymbol{\Phi}_K)$ the subarray covariance matrices as defined in (5.3) are used since (5.5) is only valid for uncorrelated source. The DOAs can be estimated from the minimization problem

$$\begin{aligned} &\min_{\boldsymbol{\theta}, \boldsymbol{P}, \sigma^2, \boldsymbol{\Phi}_1, \ldots, \boldsymbol{\Phi}_K} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{P}, \sigma^2, \boldsymbol{\Phi}_1, \ldots, \boldsymbol{\Phi}_K) \\ &\text{s.t.} \quad \boldsymbol{P} \succeq 0, \\ &\qquad \sigma^2 > 0, \end{aligned} \tag{5.31}$$

where $\boldsymbol{P} \succeq 0$ denotes that the matrix $\boldsymbol{P}$ is positive semidefinite. The function $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{P}, \sigma^2, \boldsymbol{\Phi}_1, \ldots, \boldsymbol{\Phi}_K)$ in (5.31) is nonconvex [BV04]. Therefore, a good initial solution is essential for the MLE.

The SSR approach introduced in Section 5.3.4 for uncorrelated sources is robust to the assumption of uncorrelated sources. This robustness results from the fact that the SPICE method, which is the basis of the proposed SSR approach, is robust to the assumption of uncorrelated sources [SBL11, Section II]. Consequently, the SSR approach as introduced in Section 5.3.4 for uncorrelated sources is applicable in the case of correlated sources[10].

5.4.2 The CRB for Correlated Sources

The derivation of the CRB for the case of correlated sources is similar to that in the case of uncorrelated sources. The CRB for the case of correlated sources is written as in (5.23) with $\boldsymbol{\Delta}_2$ defined as

$$\boldsymbol{\Delta}_2 = [\frac{d\boldsymbol{r}}{d\mathfrak{p}^T}, \frac{d\boldsymbol{r}}{d\sigma^2}, \frac{d\boldsymbol{r}}{d\boldsymbol{\zeta}_2^T}, \ldots, \frac{d\boldsymbol{r}}{d\boldsymbol{\zeta}_K^T}], \tag{5.32}$$

where $\mathfrak{p}$ is a real vector of length L^2 which represents the unknown parameters of the source covariance matrix. More precisely $\mathfrak{p}$ contains the diagonal of $\boldsymbol{P}$ and the real and imaginary parts of the upper diagonal of the matrix $\boldsymbol{P}$. In the following, the behaviour of the CRB at high SNR is demonstrated by simulation and an asymptotic (for high SNR) analysis of this behaviour is carried out.

[10] This means that if correlated sources are present, the SPICE algorithm will simply ignore the source correlations and the unknown phases $\boldsymbol{\Phi}_k$, for $k = 1, \ldots, K$.

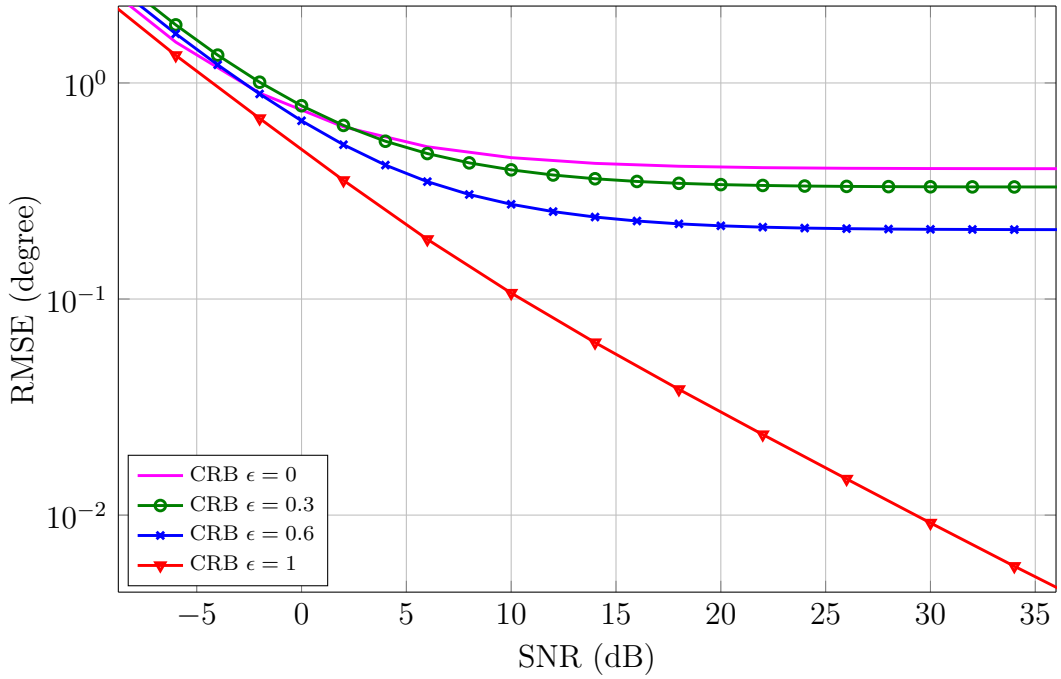

Fig. 5.3. The CRB in the case of correlated sources for different source correlation ϵ.

Example

In Fig. 5.3, the CRB is displayed for $K = 12$ subarrays each consists of two sensors and $L = 2$ equally-powered correlated sources,[11] i.e., the matrix FIM_k, for $k = 1, \ldots, K$, are not invertible. The source covariance matrix is written as

$$\boldsymbol{P} = p\,\boldsymbol{\Upsilon} \tag{5.33}$$

where $\boldsymbol{\Upsilon} = \begin{bmatrix} 1 & \epsilon \\ \epsilon^* & 1 \end{bmatrix}$, the correlation factor ϵ satisfies $0 \leq |\epsilon| \leq 1$, and p is the power of each of the two sources. In Fig. 5.3, the CRB is displayed for correlation factor ϵ of 0, 0.3, 0.6, and 1, where the latter correlation value indicates coherent sources. Observe in Fig. 5.3 that the CRB of the estimated DOAs for correlated sources behaves similar to the uncorrelated sources case of Fig. 5.2. However, the CRB decreases with the increase of ϵ. Interestingly, for coherent sources, i.e., for $\epsilon = 1$, the CRB approaches zero at high SNR, which is in exact contrast to the case of uncorrelated or partly correlated sources where the CRB does not vanish with SNR.

In the sequel, the aforementioned behaviour of the CRB is analyzed asymptotically for high SNR values. Following the steps of Section 5.3.3 for the case of uncorrelated

[11] The same configuration as in the case S1 in Section 5.3.1, except for the source correlation, is used. For the details on the array geometry parameters please refer the array setup described in Section 5.5.

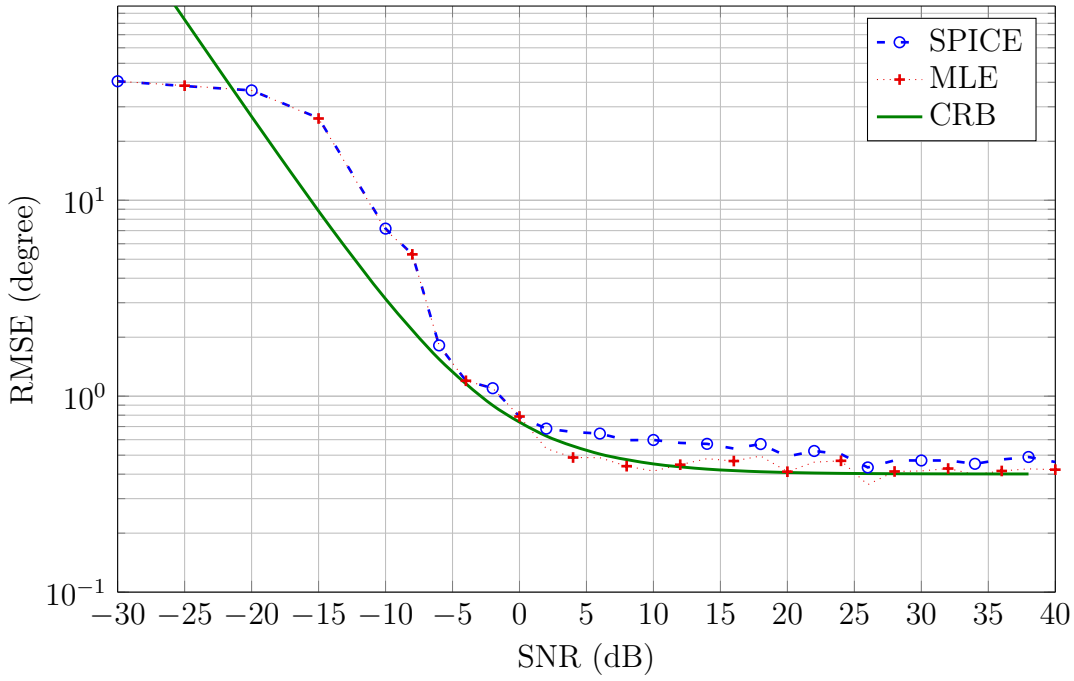

Fig. 5.4. The RMSE of the proposed non-coherent DOA estimation methods averaged over 500 realizations assuming uncorrelated sources versus SNR.

sources, in the correlated source case the CRB at high SNR is written as

$$\mathrm{CRB}_{\boldsymbol{\theta}}^{-1}|_{p\gg\sigma} \approx N[\frac{d\boldsymbol{\varphi}}{d\theta_1},\ldots,\frac{d\boldsymbol{\varphi}}{d\theta_L}]^H\,\overline{\overline{\boldsymbol{V}}}\,[\frac{d\boldsymbol{\varphi}}{d\theta_1},\ldots,\frac{d\boldsymbol{\varphi}}{d\theta_L}], \tag{5.34}$$

where $\boldsymbol{\varphi}=[\boldsymbol{\varphi}_1^T,\ldots,\boldsymbol{\varphi}_K^T]^T$, $\boldsymbol{\varphi}_k=\mathrm{vec}(\boldsymbol{V}_k\boldsymbol{\Phi}_k\boldsymbol{\Upsilon}\boldsymbol{\Phi}_k^H\boldsymbol{V}_k^H)$, $\overline{\overline{\boldsymbol{V}}}=\mathrm{blkdiag}\left(\overline{\overline{\boldsymbol{V}}}_1,\ldots,\overline{\overline{\boldsymbol{V}}}_K\right)$, and $\overline{\overline{\boldsymbol{V}}}_k=(\boldsymbol{V}_k\boldsymbol{\Phi}_k\boldsymbol{\Upsilon}\boldsymbol{\Phi}_k^H\boldsymbol{V}_k^H)^{-T}\otimes(\boldsymbol{V}_k\boldsymbol{\Phi}_k\boldsymbol{\Upsilon}\boldsymbol{\Phi}_k^H\boldsymbol{V}_k^H)^{-1}$. Note that $\mathrm{CRB}_{\boldsymbol{\theta}}^{-1}|_{p\gg\sigma}$ depends neither on p nor on σ^2. Thus, based on the rank of the matrix $\overline{\overline{\boldsymbol{V}}}_k^{-1}$, denoted as $\overline{\overline{\rho}}_k$, the following two cases are distinguished:

- The case when $|\epsilon|<1$ in which $\overline{\overline{\rho}}_k=M_k^2$, for $k=1,\ldots,k$, consequently, the matrix $\overline{\overline{\boldsymbol{V}}}^{-1}$ is full rank, and the CRB does not vanish at high SNR.

- The case when $|\epsilon|=1$, i.e., the sources are fully correlated, in which $\overline{\overline{\rho}}_1=\cdots\overline{\overline{\rho}}_K=1$, consequently, the matrix $\overline{\overline{\boldsymbol{V}}}^{-1}$ is rank deficient and drives the CRB to zero at high SNR.

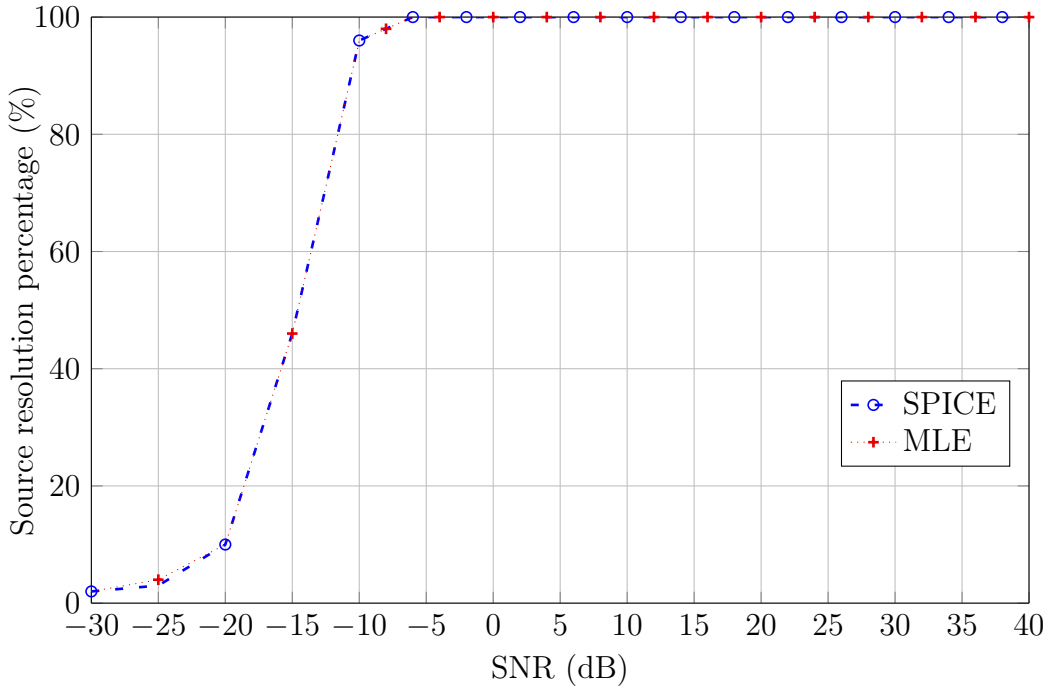

Fig. 5.5. The resolution percentage of the proposed non-coherent DOA estimation methods averaged over 500 realizations assuming uncorrelated sources versus SNR.

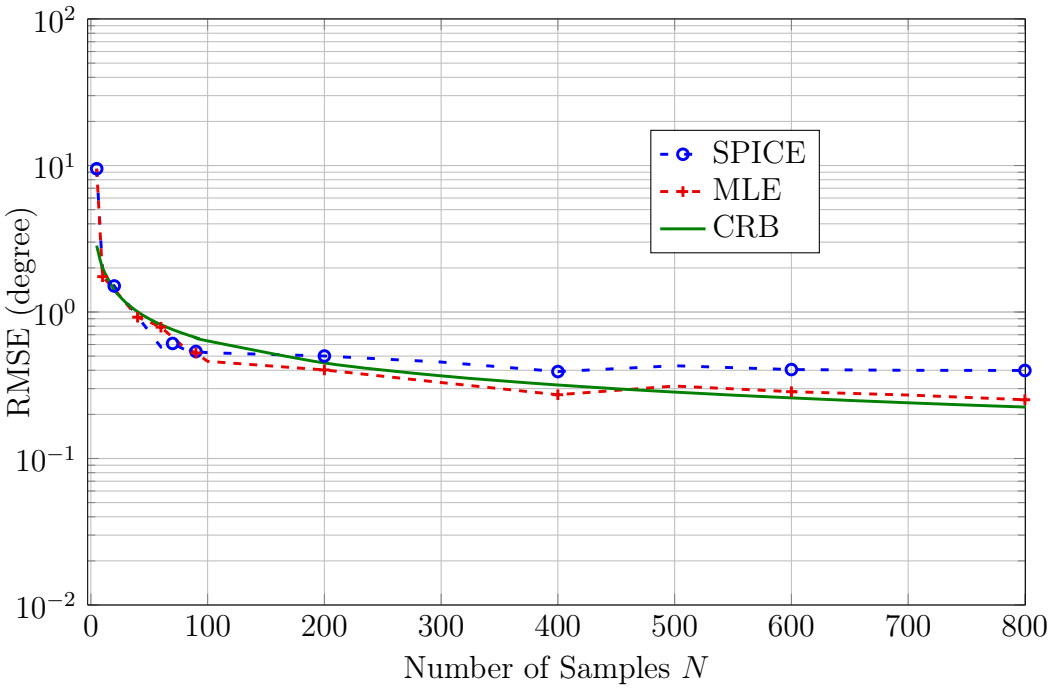

Fig. 5.6. The RMSE of the proposed non-coherent DOA estimation methods averaged over 500 realizations assuming uncorrelated sources versus the number of samples N.

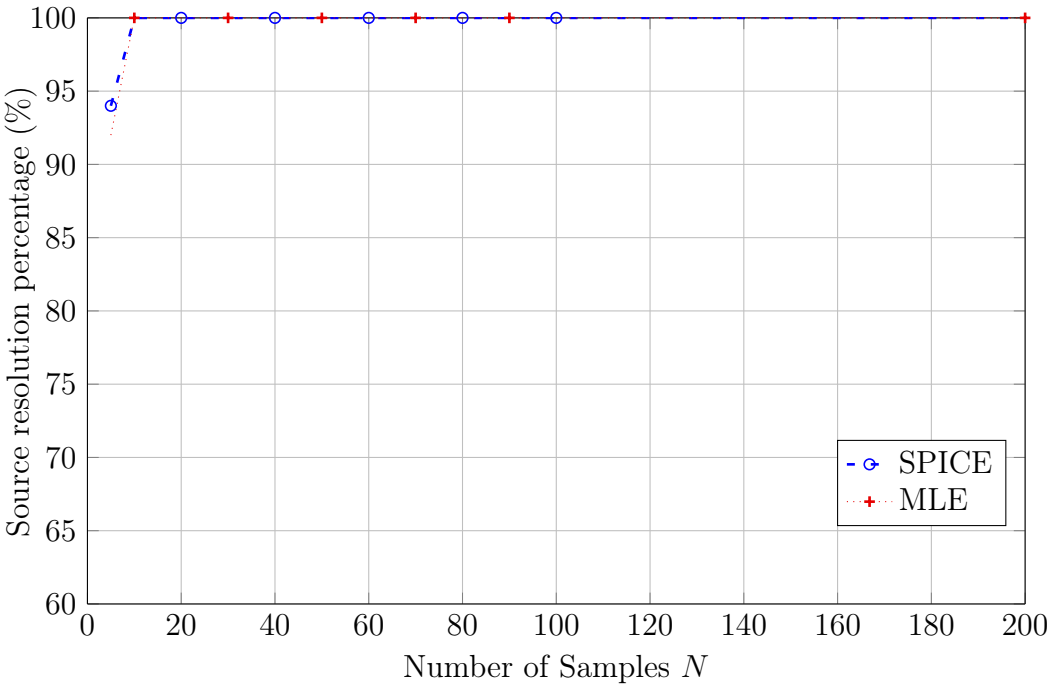

Fig. 5.7. The resolution percentage of the proposed non-coherent DOA estimation methods averaged over 500 realizations assuming uncorrelated sources versus the number of samples N.

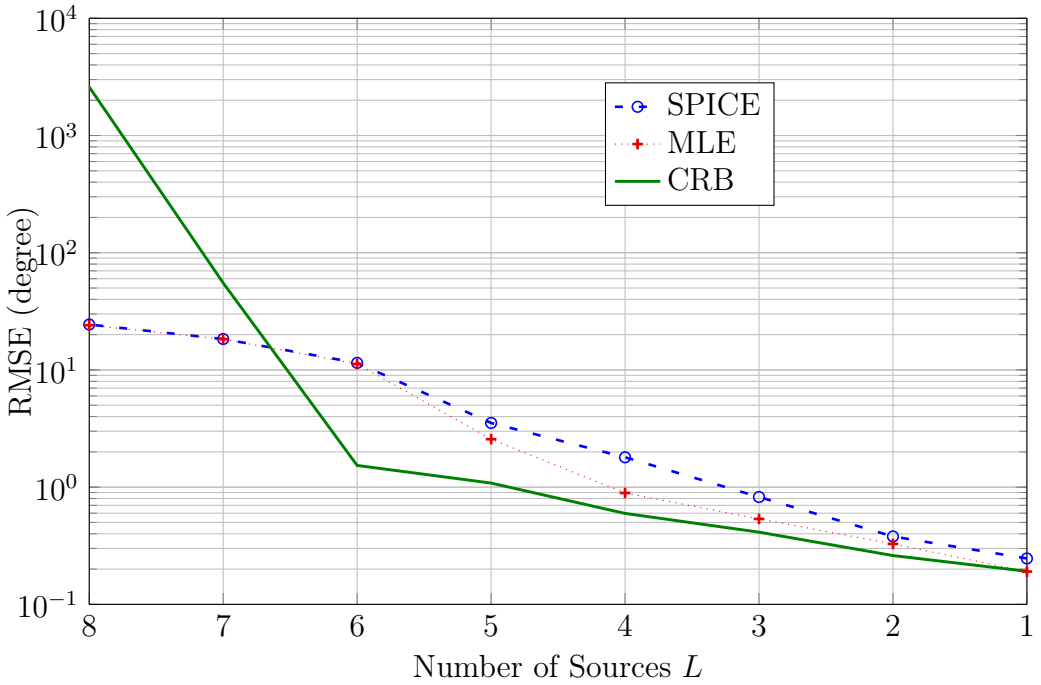

Fig. 5.8. The RMSE of the proposed non-coherent DOA estimation methods averaged over 500 realizations assuming uncorrelated sources versus the number of sources L.

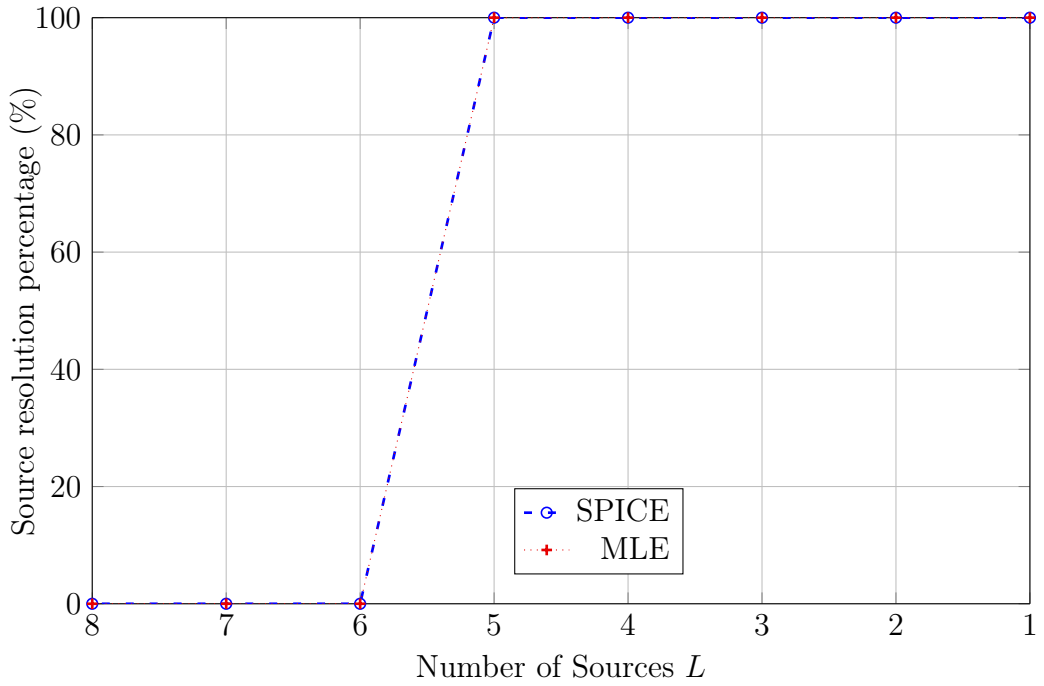

Fig. 5.9. The resolution percentage of the proposed non-coherent DOA estimation methods averaged over 500 realizations assuming uncorrelated sources versus the number of sources L.

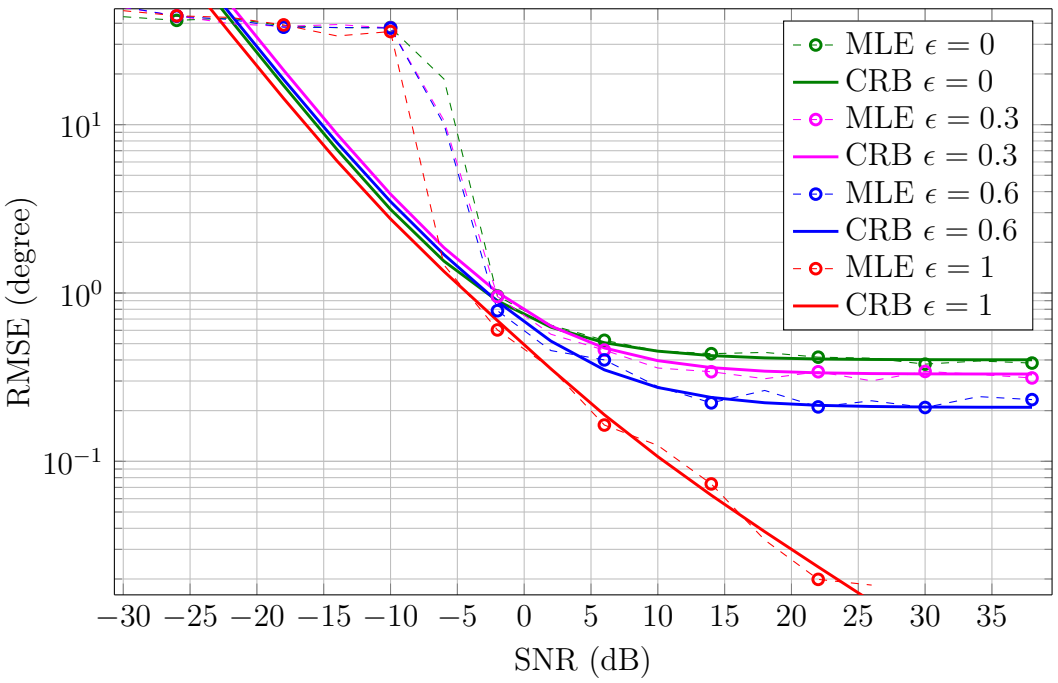

Fig. 5.10. The RMSE of the MLE for DOA estimation averaged over 500 realizations for correlated sources versus SNR.

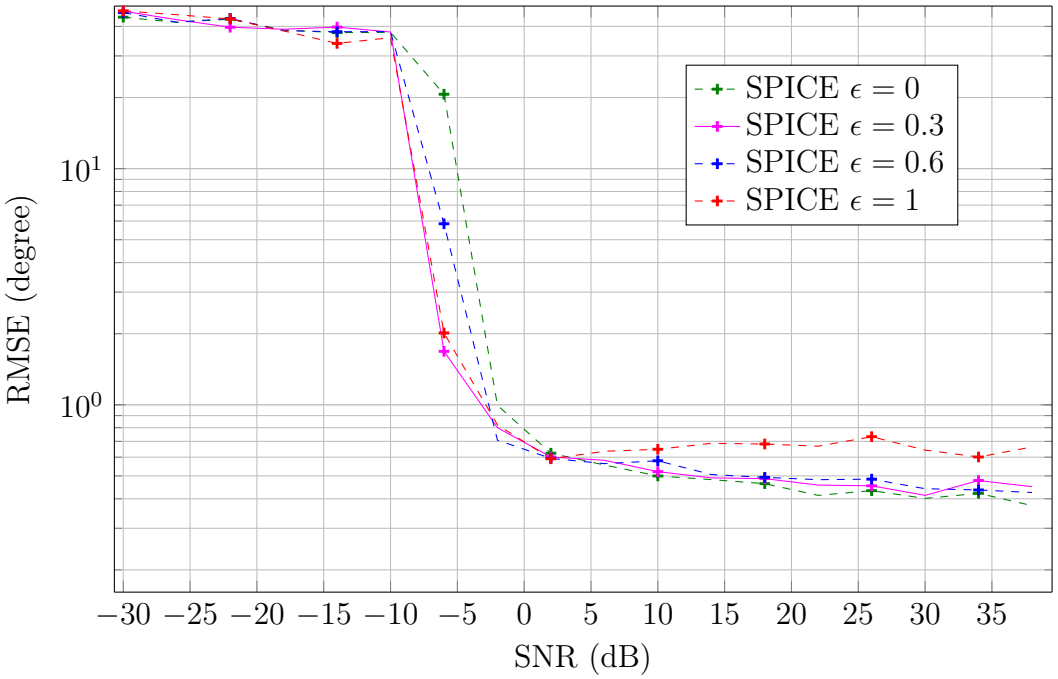

Fig. 5.11. The RMSE of the SPICE method for DOA estimation averaged over 500 realizations for correlated sources versus SNR.

5.5 Simulation Results

In this section, the performance of the presented MLE and SSR estimation methods is demonstrated using simulations for both cases uncorrelated sources and correlated sources. The cvx [CR12] framework is used to solve the SPICE optimization problem in (2.33), where the field-of-view is sampled every 0.1°. The MLE is initialized with the solution of the SPICE method and the MATLAB command *fmincon* is used to compute the MLE as presented in (5.21) and (5.31) for uncorrelated and correlated sources, respectively.

In the simulations, an array composed of $K = 12$ subarrays each is comprised of 2 sensors is considered. The location of the first sensors in the 12 subarrays measured in half-wavelength are $(0, 0)$, $(17.3, 6)$, $(-2.4, 6.2)$, $(10.5, -2)$, $(12.7, 2.1)$, $(4.6, -2.4)$, $(4.6, 4.5)$, $(4.5, 5.3)$, $(2.3, 9)$, $(10.2, 8.1)$, $(10.2, 4)$, and $(13.4, 6)$. These locations are considered to be unknown during the DOA estimation process. The locations of the second sensors in each subarray with respect to the first sensor in the corresponding subarray measured in half-wavelength are $(6.5, 0)$, $(4.4, 0)$, $(3.5, 0)$, $(2.6, 0)$, $(2.6, 0)$, $(2.5, 0)$, $(1.9, 0)$, $(1.5, 0)$, $(1.4, 0)$, $(1.3, 0)$, $(1, 0)$, and $(0.5, 0)$. These locations are considered to be known. Signals of two far-field equal-powered uncorrelated sources are impinging onto the subarrays from directions $-11.4°$ and $-1.1°$. In the simulations,

the RMSE for the estimated DOAs is computed over 500 realizations for the SPICE and the MLE approaches. The CRB for the considered scenario is displayed.

In Fig. 5.4, the averaged performance of the SPICE and the MLE for a fixed number of samples $N = 50$ is plotted against SNR. It can be observed in Fig. 5.4 that the MLE and the SPICE method achieve the CRB at high SNR. In Fig. 5.5, the source resolution percentage of the considered DOA estimation methods is plotted against the SNR, where two sources are considered to be resolved if the error in the estimated DOAs is less than the angular separation between the two sources [PPG11]. Observe that for SNR ≥ -8 dB, the MLE and SPICE method can always identify the sources and for SNR ≤ -20 dB the resolution percentage is almost zero.

In Fig. 5.6, the RMSE of DOA estimation using SPICE and MLE is plotted against the number of snapshots N for a fixed SNR $= -2$ dB. The MLE achieves the CRB for $N \geq 20$ samples, whereas the SPICE method is above the CRB because of the bias resulting from the nature of the SSR approaches [MÇW05]. In Fig. 5.7, the source resolution percentage is plotted against N. Observe that the SPICE and the MLE achieve 100% resolution percentage for $N \geq 20$.

In Fig. 5.8 and Fig. 5.9, for a fixed SNR of -2 dB and fixed number of samples $N = 50$, the number of sources L is changed. The source DOAs are chosen in order from the set $\{15°, -15°, 30°, -30°, 45°, -45°, 60°, -60°\}$. Observe in Fig. 5.8 that for small number of sources $L \leq 4$ the MLE and the SPICE achieves the CRB. In Fig. 5.9, it can be seen that for $L \leq 5$ both the SPICE and the MLE methods are always able to identify the sources. Note that since $M_k = 2$ for $k = 1, \ldots, 12$, none of the subarrays can individually identify more than one source, however, with the proposed methods, which exploit the diverse structure of the subarrays, up to $L = 5$ sources can be identified.

In the following, the performance of the MLE and SPICE considering $L = 2$ correlated sources is investigated. In Fig. 5.10, the number of samples is fixed to $N = 50$ and the RMSE for DOA estimation of the MLE is plotted against SNR for different values of the correlation factor $\epsilon = 0, 0.3, 0.6$, and $\epsilon = 1$, where ϵ is defined in (5.33). Note that the RMSE decreases by increasing ϵ. For coherent sources, i.e., $\epsilon = 1$, the RMSE approaches zero for high SNR, which is in correspondence to the discussion in Section 5.4.2. The averaged performance of the SPICE for the same scenario is shown in Fig. 5.11. Note that the SPICE method is robust against the assumption of correlated sources, i.e., the performance of SPICE does not degrade much with the increased correlation between the sources, see [SBL11].

5.6 Uniform Linear Subarrays with Large Number of Sensors

In this section, novel DOA estimation algorithms for non-coherent processing using partly calibrated arrays are proposed for the special case where:

1. The subarrays exhibit the uniform linear (UL) array geometry. Denote as $\overline{d}_k = \overline{n}_k \overline{d}$ the distance between two successive sensors in the kth subarray. The distance $\overline{d}_k$ is assumed to be an integer multiple $\overline{n}_k$ of a distance $\overline{d}$.

2. All subarrays can identify the sources, i.e., $M_k > L$, for $k = 1, \ldots, K$.

In this special case, search-free DOA estimation can be carried out, which results in fast and accurate DOA estimation. The proposed algorithms are based on finding the common roots (CR) of multiple univariate polynomials corresponding to the individual subarrays. These polynomials are the Root-MUSIC polynomials [Bar83]. Two algorithms are proposed using the generalized Sylvester matrix (GSM) [WW11] to find the CRs and to estimate the DOAs. Simulation results demonstrate that the proposed algorithms outperform existing decentralized methods and resolve many possible DOA estimation ambiguities caused by subarray geometries.

5.6.1 Computing the Local Polynomials

Since the number of sensors at the kth subarray M_k is larger than the number of sources L, the kth subarray can estimate the noise eigenvectors matrix from its sample covariance matrix $\hat{\boldsymbol{R}}_k$. Let $\hat{\boldsymbol{U}}_{\mathrm{n},k}$ denote the estimated noise eigenvectors matrix at the kth subarray. Then, the Root-MUSIC polynomial in (2.20) can be computed at the kth subarray from $\hat{\boldsymbol{U}}_{\mathrm{n},k}$, denote this polynomial as $\hat{\mathcal{F}}_k(z)$.

The polynomial $\hat{\mathcal{F}}_k(z)$ can be written as a multiplication of two polynomials

$$\hat{\mathcal{F}}_k(z) = \hat{\overline{\mathcal{F}}}_k(z)\hat{\underline{\mathcal{F}}}_k(z), \tag{5.35}$$

where $\hat{\overline{\mathcal{F}}}_k(z)$ and $\hat{\underline{\mathcal{F}}}_k(z)$ are formed, respectively, from the outside and the inside the unit circle (UC) roots of $\hat{\mathcal{F}}_k(z)$. Due to the conjugate reciprocal property of the root pairs, both polynomials contain the same spatial information. Therefore, one of the

polynomials is sufficient to estimate the DOAs. In the following, the polynomial $\hat{\overline{\mathcal{F}}}_k(z)$ is used to estimate the DOAs. Let

$$\hat{\overline{\mathcal{F}}}_k(z) = \hat{\kappa}_{k,0} + \hat{\kappa}_{k,1} z + \ldots + z^{D_k}, \quad k = 1, \ldots, K \tag{5.36}$$

where $\hat{\kappa}_{k,i}$, for $i = 0, \ldots, D_k - 1$, is the complex coefficient of $\hat{\overline{\mathcal{F}}}_k(z)$ scaled such that $\hat{\kappa}_{k,D_k} = 1$, and $D_k = \overline{n}_k(M_k - 1)$ is the degree of $\hat{\overline{\mathcal{F}}}_k(z)$. Only these coefficients are sent to the PC, thus, the communication cost per subarray is $D_k = n_k(M_k - 1)$. In centralized processing where all the measurements are sent to PC, the communication cost per subarray is $M_k N$. Thus, the proposed communication scheme reduces the communication cost, since normally $N \gg \overline{n}_k$.

Sorting the roots of $\hat{\overline{\mathcal{F}}}_k(z)$ such that $|\hat{z}_1| \leq \ldots \leq |\hat{z}_{D_k}|$, the smallest L roots are called the signal roots (containing the source DOA information) and the remaining $D_k - L$ roots are called the noise roots. Since each subarray can identify the L sources, all the subarrays share the same signal roots. Thus, the CRs of all local polynomials in the set $\mathscr{F} = \{\hat{\overline{\mathcal{F}}}_k(z)\}_{k=1}^{K}$ contain L signal roots, and can be used for DOA estimation.

Having received all the K local polynomials from the subarrays, the PC uses the GSM to estimate the CRs.

5.6.2 The Generalized Sylvester Matrix

Originally, the Sylvester matrix is defined for two polynomials [Akr93, CGTW95]. In [VS78, KFMH06, WW11] generalizations of the Sylvester matrix are introduced for more than two polynomials. In the following, the generalized matrix from [WW11] is used as it has the smallest size compared to the generalizations defined in [VS78] and [KFMH06].

Assume without loss of generality that $D_1 \geq D_2 \geq \ldots \geq D_K$. The GSM $\boldsymbol{\mathcal{S}}$ consists of K blocks, i.e.,

$$\boldsymbol{\mathcal{S}} = \left[\boldsymbol{\mathcal{S}}_1^T, \boldsymbol{\mathcal{S}}_2^T, \ldots, \boldsymbol{\mathcal{S}}_K^T\right]^T \in \mathbb{C}^{r \times h}, \tag{5.37}$$

where $r = (K-1)D_K + D_1$, and $h = D_1 + D_K$. The first $K - 1$ blocks correspond to the first $K - 1$ polynomials such that, for $k = 1, \ldots, K - 1$

$$\boldsymbol{\mathcal{S}}_k = \left[\boldsymbol{\varrho}_{k,D_K-1}^T, \boldsymbol{\varrho}_{k,D_K-2}^T, \ldots, \boldsymbol{\varrho}_{k,0}^T\right]^T \in \mathbb{C}^{D_K \times h}, \tag{5.38}$$

where each row $\boldsymbol{\varrho}_{k,m} = [0, \ldots, 0, \hat{\kappa}_{k,0}, \ldots, \hat{\kappa}_{k,D_k}, 0, \ldots, 0]$ contains zeros at the first $h - m - D_k - 1$ and the last m entries. The Kth block corresponds to the polynomial with the smallest degree D_K,

$$\boldsymbol{\varrho}_K = \left[\boldsymbol{\varrho}_{K,D_1-1}^T, \boldsymbol{\varrho}_{K,D_1-2}^T, \ldots, \boldsymbol{\varrho}_{K,0}^T\right]^T \in \mathbb{C}^{D_1 \times h}. \tag{5.39}$$

The matrix $\boldsymbol{\mathcal{S}}$ is of rank h and it drops rank if and only if the set of polynomials $\mathscr{F}$ has at least one CR [WW11]. More precisely, if the polynomials in the set $\mathscr{F}$ have n CRs then

$$\operatorname{rank}(\boldsymbol{\mathcal{S}}) = h - n. \tag{5.40}$$

If the polynomials in $\mathscr{F}$ were exact (or in array processing context N approaches infinity), then equation (5.40) implies that $\boldsymbol{\mathcal{S}}$ would have exactly n zero singular values. In such case, the CRs can be computed using triangularization [WW11]. However, the polynomials are not exact and the signal roots are only approximately similar, thus, $\boldsymbol{\mathcal{S}}$ will have n small but non-zero singular values. In the following, two algorithms to estimate the CRs of polynomials in $\mathscr{F}$ are proposed.

5.6.3 Algorithm I

Let z_i be one CR of the polynomials in $\mathscr{F}$ and $\mathcal{O}(\boldsymbol{\mathcal{S}})$ be the null space of $\boldsymbol{\mathcal{S}}$. Then, the Vandermonde vector $\boldsymbol{z}_i = \left[1, z_i, z_i^2, \ldots, z_i^{h-1}\right]^T$ belongs to $\mathcal{O}(\boldsymbol{\mathcal{S}})$, i.e., $\boldsymbol{\mathcal{S}}\,\boldsymbol{z}_i = \boldsymbol{0}$. Thus, for all the CRs $z_1, \ldots, z_n$, the matrix

$$\boldsymbol{Z}_n = [\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n] \in \mathbb{C}^{h \times n}, \tag{5.41}$$

forms a set of basis for $\mathcal{O}(\boldsymbol{\mathcal{S}})$. Let $\boldsymbol{e}_1, \ldots, \boldsymbol{e}_n$ be the first n right singular vectors of $\boldsymbol{\mathcal{S}}$ (corresponding to the n smallest singular values), then the matrix $\boldsymbol{E}_n = [\boldsymbol{e}_1, \ldots, \boldsymbol{e}_n]$ also forms a set of basis for $\mathcal{O}(\boldsymbol{\mathcal{S}})$. Therefore, from equation (5.40) it can be inferred that $\boldsymbol{E}_n$ and $\boldsymbol{Z}_n$ span the same subspace. Moreover, $\boldsymbol{Z}_n$ has a Vandermonde structure, thus the ESPRIT algorithm [RK89] can be used to estimate the n CRs from $\boldsymbol{E}_n$ [PPG12]. In analogy to [PPG12], two matrices $\overline{\boldsymbol{E}}_n$ and $\underline{\boldsymbol{E}}_n$ are formed by deleting the first and the last rows of $\boldsymbol{E}_n$, respectively. The n sought roots are, then, the eigenvalues of the matrix

$$\overline{\boldsymbol{\Psi}}_n = \left(\underline{\boldsymbol{E}}_n^H \underline{\boldsymbol{E}}_n\right)^{-1} \underline{\boldsymbol{E}}_n^H \overline{\boldsymbol{E}}_n. \tag{5.42}$$

Note that if the subarrays are not identical, then the polynomials in $\mathscr{F}$ have at least L CRs. However, in low SNRs some of the noise roots corresponding to the subarray(s) with the largest number of sensors may cause the matrix $\boldsymbol{\mathcal{S}}$ to be rank deficient. Consequently, the estimated $\mathcal{O}(\boldsymbol{\mathcal{S}})$ is expanded for $n = L, L+1, \ldots, h,$. For each value

of n, n roots are computed from the eigenvalues of $\overline{\boldsymbol{\Psi}}_n$ in equation (5.42). Let the remainders of the polynomials in $\mathscr{F}$ at each of the n roots be

$$\mathrm{Rem}_{n,i} = \sum_{k=1}^{K} \left| \hat{\overline{\mathcal{F}}}_k(z_i) \right|_2^2, \quad i = 1, \ldots, n. \tag{5.43}$$

$\mathrm{Rem}_{n,i}$ quantifies the quality of each estimated root for different values of n. The remainder Rem_n is defined as the sum of the L smallest $\{\mathrm{Rem}_{n,i}\}_{i=1}^{n}$ at each value of n. The L roots that minimize the remainder Rem_n are chosen as the estimates of the L common signal roots. Note that if all the subarrays are identical, then the set of CRs contains not only the signal roots but also the noise roots, i.e., $n = D_1 = D_2 = \ldots = D_K$. In this special case, the L roots which are closer to the UC are chosen. Having estimated the L signal roots, the DOAs can then be computed as in (2.21).

5.6.4 Algorithm II

Let $\boldsymbol{z} = \left[1, z, z^2, \ldots, z^{h-1}\right]^T$, then the following function

$$f(z) = \|\boldsymbol{\mathcal{S}}\,\boldsymbol{z}\|_2^2 = \boldsymbol{z}^H \boldsymbol{\mathcal{S}}^H \boldsymbol{\mathcal{S}}\,\boldsymbol{z} \tag{5.44}$$

is exactly zero whenever z is equal to one of the CRs of the polynomials in $\mathscr{F}$, since $\boldsymbol{\mathcal{S}}\,\boldsymbol{z}\,|_{z=z_i} = \boldsymbol{0}$ for $i = 1, \ldots, L$. Therefore, the CRs of the polynomials in $\mathscr{F}$ can be estimated by minimizing $f(z)$. This can be achieved by rooting $f(z)$. However, $f(z)$ has $2(h-1)$ roots, and these roots occur in pairs. Therefore, the L signal roots are chosen in two steps. First, the $h-1$ roots of $f(z)$ which are outside the UC is selected. Second, the remainder defined in equation (5.43) is calculated for the $h-1$ roots, and the L roots with the smallest remainder are chosen. Note that choosing the L roots closest to the UC yield the same performance. Having estimated the L signal roots, the DOAs can be estimated from equation (2.21).

5.6.5 Simulation Results

An array composed of 6 uniform linear subarrays with 4, 5, 6, 7, 7, and 7 sensors is considered. The positions of the reference sensors of the 6 subarrays measured in half-wavelength are $(0, 0)$, $(0.3, 0.5)$, $(-0.4, 0.4)$, $(1.1, 0.91)$, $(1.2, 0.61)$, and $(1.5, 0.9)$, respectively. The inter-sensor spacing in all the subarrays is taken to be the signal half-wavelength. Three uncorrelated Gaussian equal-powered sources impinge onto the array from directions $-3.32°$, $1.41°$, and $17.85°$. A number of $N = 50$ snapshots

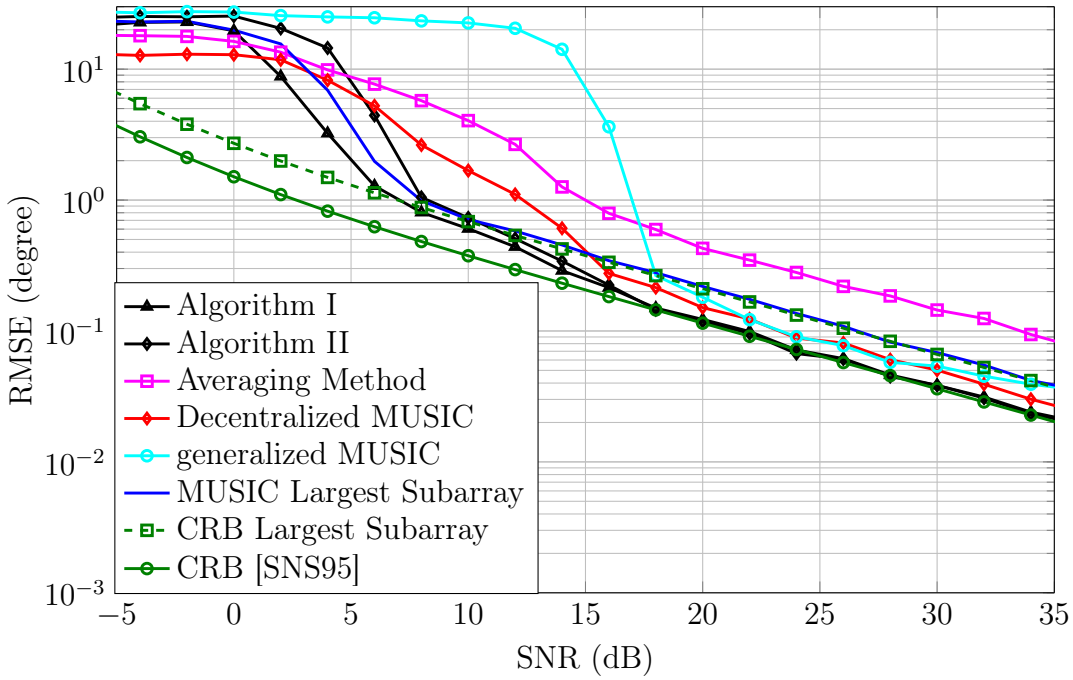

Fig. 5.12. DOA estimation performance (RMSE) vs. SNR in unambiguous scenario.

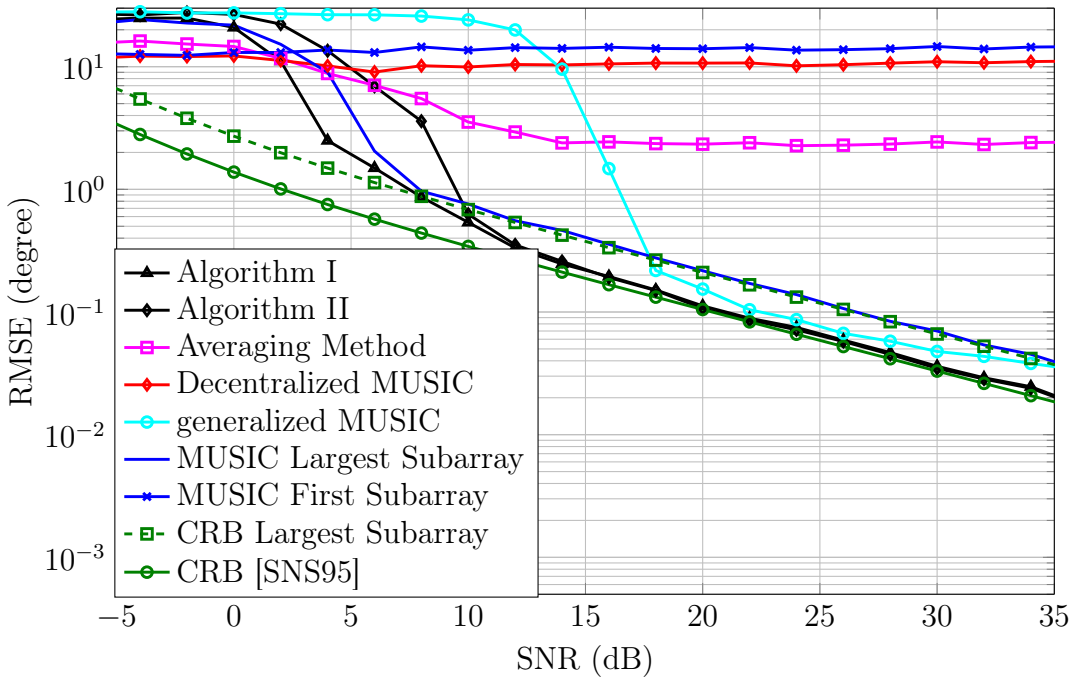

Fig. 5.13. DOA estimation performance (RMSE) vs. SNR in ambiguous scenario.

are collected at each subarray. The two proposed algorithms are compared with the following algorithms: (1) averaging method, where the subarrays use the Root-MUSIC method to estimate the DOAs and the PC average these estimates, (2) the decentralized MUSIC method of [SS92] which uses averaging of the local estimates weighted by their estimated variances as defined in [SS92], (3) the generalized spectral MUSIC method in [RF04] using a search grid with granularity of 0.1°. All algorithms are compared to the CRB for non-coherent processing defined as CRB $= \left(\sum_{k=1}^{K} \boldsymbol{C}_k^{-1}\right)^{-1}$ [SS92], where the matrix $\boldsymbol{C}_k$ is the CRB matrix corresponding to the kth subarray as defined in [SA89]. In the simulations, 500 Monte Carlo runs are used.

Fig. 5.12 shows the RMSE of the DOA estimates versus SNR. It is obvious that the performance of the two proposed algorithms stay close to the CRB and outperform the three competing methods at high SNR as well. The proposed methods also have better threshold performance when compared to the other algorithms.

Fig. 5.13 shows the RMSE for the same setup as described above except that the inter-sensor distance at the first subarray is set to one wavelength. Thus, the first subarray is unable to identify the sources unambiguously. As it can be seen in this figure, the ambiguous estimation of the first subarray affects the performance of averaging and decentralized MUSIC methods. However, the proposed algorithms are still able to resolve the sources, since the CRs of all the subarray polynomials remain unchanged.

5.7 Summary

In this chapter, non-coherent DOA estimation using partly calibrated arrays has been considered. We have focused our presentation and contributions on the case where none of the subarrays is able to individually identify all the sources. A bound on the maximum number of uncorrelated sources that can be estimated using non-coherent processing has been presented. Moreover, the CRB for non-coherent processing has been derived and its behaviour at high SNR has been analyzed. Two methods, namely, the MLE and the SSR-based approach, have been proposed to estimate the DOAs from the sample covariance matrices received from all subarrays. Finally, the special case when all the subarrays are UL and can identify the sources has been considered. Non-coherent DOA estimation algorithms, based on finding the CRs of multiple univariate polynomials corresponding to the individual subarrays, have been proposed and shown to outperform the state-of-the-art DOA estimation methods by simulations.

Chapter 6
Conclusions and Outlook

In this work two approaches towards decentralized DOA estimation are considered, namely, coherent and non-coherent processing. DOA estimators and the corresponding performance bounds are derived.

For coherent processing, decentralized DOA estimation is carried out using the DESPRIT and decentralized Root-MUSIC algorithms, which are based on the DPM for the eigendecomposition of the sample covariance matrix. An analytical expressions for the second order statistics of the eigenvectors and eigenvalues of the sample covariance matrix computed using the DPM are derived. These simplified expressions can be combined easily with the MSE expressions of the conventional estimators e.g., ESPRIT and Root-MUSIC algorithms, to obtain the MSE of the corresponding decentralized implementations, e.g., DESPRIT and decentralized Root-MUSIC algorithms as presented in the thesis. Our analytical expressions show that the AC errors in the DPM and the DESPRIT algorithm are dominant when the number of samples is very large or the SNR is very high. Further, based on this expressions, the minimum number of AC iterations, which are required to obtain DOA estimates with accuracy comparable to that of the centralized case, can be computed for a certain number of sample and SNR. The DESPRIT algorithm requires: 1) a shift-invariant array geometry and 2) the number of sources to be known beforehand. These two aspects of the DESPRIT algorithm are considered where: 1) the DESPRIT algorithm is extended to arbitrary array geometries using interpolation and 2) an algorithm for detecting the number of sources is presented, which decides that a source is present even before computing the eigenvalue and eigenvector corresponding to this source. Thus, it results in a huge reduction of computational (and in decentralized scenarios communication) cost.

The DPM requires a sufficiently large number of PM iterations to converge for each eigenvector, which results in a large communication cost. Further, the DPM is a batch processing algorithm, i.e., in the DPM, first the individual subarrays collect and store measurements, then, they perform decentralized estimation assuming that the sources are stationary in the batch of the collected samples. However, if the sources are moving the DOAs of the signals of sources must be updated online, i.e., at each sample. The aforementioned shortcomings of the DPM are considered where:

1. The decentralized Lanczos method for eigendecomposition of the sample covariance matrix, which yields reduced communication cost compared to the DPM, is introduced.

2. Online generalized eigendecomposition is implemented in a decentralized fashion. Further, DOA tracking, where the DOA estimates are updated for each acquired sample, is performed using the online decentralized generalized eigendecomposition, whereas the DPM yields only batch-based eigendecomposition and thus can be used only for batch-based DOA estimation.

The usage of the two aforementioned eigendecomposition and generalized eigendecomposition algorithms is not restricted to DOA estimation. Rather, they can be used in any eigendecomposition application, e.g., spectrum sensing, source enumeration, and linear discriminant analysis.

The proposed decentralized implementations of the ESPRIT and Root-MUSIC algorithms are applicable in the case of uncorrelated sources. The performance of the proposed algorithms degrades with increased source correlation and they completely fail to identify the sources if the sources are fully correlated. Developing a decentralized DOA estimation algorithm, which is able to identify fully correlated sources still an open problem. Moreover, the proposed decentralized source enumeration algorithm is derived assuming a large number of AC iterations and a heuristic is used to approximate the statistical distribution of the sample covariance matrix projected on some of its eigenvectors. Although the proposed algorithm is shown to perform well using a moderate number of AC iterations and moderately high SNR, developing a better performing detector which fully uses the true distribution of the projected covariance matrix and takes into consideration the finite number of AC iterations still an open problem.

For non-coherent processing, DOA estimation is achieved using the MLE and a computationally simpler approach based on SSR. Assuming uncorrelated sources, a sufficient condition for the unique identifiability of the sources in the aforementioned non-coherent processing scheme is presented and it is shown that under mild conditions, with the non-coherent system of subarrays, it is possible to identify more sources than identifiable by each individual subarray. Moreover, the CRB for the measurement model is derived and used to assess the presented DOA estimators. The behaviour of the CRB at high SNR is analyzed. In contrast to coherent processing, it is shown that the CRB approaches zero at high SNR only if at least one subarray can identify the sources individually.

The identifiability condition introduced in this thesis is only valid for the case of uncorrelated sources. In the case of correlated sources, the identifiability is still an open problem. Moreover, the SSR approach used in this work is shown to be robust against

the assumption of uncorrelated sources. However, the CRB predicts that the DOA estimation performance can be enhanced by exploiting the source coherence. Developing a SSR-based DOA estimation algorithm which exploits the correlation is still an open problem.

Appendix

A.1 Proof of Theorem 1

The proof of Theorem 1 is achieved in two parts. In the first part, we prove that $\{\check{\boldsymbol{u}}_i(P)\}_{i=1}^{M}$ are the eigenvectors of the matrix $\check{\boldsymbol{R}}(P)$. In the second part, we prove that $\{\check{\lambda}_i(P)\}_{i=1}^{M}$ are the eigenvalues of the matrix $\check{\boldsymbol{R}}(P)$.

Part I: $\{\check{\boldsymbol{u}}_i(P)\}_{i=1}^{M}$ are the eigenvectors of the matrix $\check{\boldsymbol{R}}(P)$

Proof. This part of Theorem 1 is proved by by induction. Thus, first the vector $\check{\boldsymbol{u}}_1(P)$, which is computed using the DPM, is proven to be the principal eigenvector of the matrix $\check{\boldsymbol{R}}(P)$. Then, assuming that the vectors $\{\check{\boldsymbol{u}}_i(P)\}_{i=1}^{l-1}$ are the principal $(l-1)$ eigenvectors of the matrix $\check{\boldsymbol{R}}(P)$, it is proven that $\check{\boldsymbol{u}}_l(P)$ is the lth eigenvector of the matrix $\check{\boldsymbol{R}}(P)$. For notation convenience, the dependency on P is dropped from $\check{\boldsymbol{R}}(P)$ and $\check{\boldsymbol{u}}_l(P)$, throughout the derivations.

Note that when the DPM is used to compute the vector $\check{\boldsymbol{u}}_1$, then (2.61) reduces to

$$\check{\boldsymbol{u}}_1^{(q)} = \frac{1}{N}\sum_{t=1}^{N}\begin{bmatrix}\check{x}_{t,1,[1]}^{(q)}\boldsymbol{x}_1^T(t)\\ \vdots \\ \check{x}_{t,1,[K]}^{(q)}\boldsymbol{x}_K^T(t)\end{bmatrix}, \tag{A.1}$$

since the matrix $\check{\boldsymbol{U}}_0 = \boldsymbol{0}$. Let

$$\tilde{\boldsymbol{X}}(t) = \begin{pmatrix}\boldsymbol{x}_1(t) & \boldsymbol{0}_{M_1} & \cdots & \boldsymbol{0}_{M_1}\\ \boldsymbol{0}_{M_2} & \boldsymbol{x}_2(t) & \cdots & \vdots \\ \vdots & \vdots & \ddots & \boldsymbol{0}_{M_{K-1}}\\ \boldsymbol{0}_{M_K} & \boldsymbol{0}_{M_K} & \cdots & \boldsymbol{x}_K(t)\end{pmatrix},$$

where $\boldsymbol{x}_k(t)$ is defined in (2.5). Then, (A.1) is written as

$$\begin{aligned}\check{\boldsymbol{u}}_1^{(q)} &= \left(\frac{K}{N}\sum_{t=1}^{N}\tilde{\boldsymbol{X}}(t)\boldsymbol{W}^P\tilde{\boldsymbol{X}}^H(t)\right)\check{\boldsymbol{u}}_1^{(q-1)}\\ &= \left(\frac{K}{N}\sum_{t=1}^{N}\sum_{k=1}^{K}\alpha_k^P\tilde{\boldsymbol{X}}(t)\boldsymbol{\beta}_k\boldsymbol{\beta}_k^H\tilde{\boldsymbol{X}}^H(t)\right)\check{\boldsymbol{u}}_1^{(q-1)},\end{aligned} \tag{A.2}$$

where the eigendecomposition of the AC weighting matrix $\boldsymbol{W}$ in (2.57) is used and P is the number of AC iterations used to compute the scalars $\{\check{x}_{t,1,[k]}^{(q)}\}_{t=1}^{N}$. The product $\tilde{\boldsymbol{X}}(t)\boldsymbol{\beta}_k$ can be rewritten as

$$\begin{aligned}\tilde{\boldsymbol{X}}(t)\boldsymbol{\beta}_k &= \left[\beta_{k,1}\boldsymbol{x}_1^T(t),\ldots,\beta_{k,K}\boldsymbol{x}_K^T(t)\right]^T \\ &= (\boldsymbol{T}\boldsymbol{\beta}_k)\odot\boldsymbol{x}(t),\end{aligned} \tag{A.3}$$

where $\boldsymbol{\beta}_k=[\beta_{k,1},\ldots,\beta_{k,K}]^T$. Substituting (A.3) into (A.2), yields

$$\begin{aligned}\check{\boldsymbol{u}}_1^{(q)} &= \left(\frac{K}{N}\sum_{t=1}^{N}\sum_{k=1}^{K}\alpha_k^P\left((\boldsymbol{T}\boldsymbol{\beta}_k)\odot\boldsymbol{x}(t)\right)\left((\boldsymbol{T}\boldsymbol{\beta}_k)\odot\boldsymbol{x}(t)\right)^H\right)\check{\boldsymbol{u}}_1^{(q-1)} \\ &= \left(\frac{K}{N}\sum_{t=1}^{N}\sum_{k=1}^{K}\alpha_k^P\left(\boldsymbol{T}\boldsymbol{\beta}_k\boldsymbol{\beta}_k^H\boldsymbol{T}^T\right)\odot\left(\boldsymbol{x}(t)\boldsymbol{x}^H(t)\right)\right)\check{\boldsymbol{u}}_1^{(q-1)} \\ &= \left(K\sum_{k=1}^{K}\alpha_k^P\left(\boldsymbol{T}\boldsymbol{\beta}_k\boldsymbol{\beta}_k^H\boldsymbol{T}^T\right)\odot\frac{1}{N}\sum_{t=1}^{N}\left(\boldsymbol{x}(t)\boldsymbol{x}^H(t)\right)\right)\check{\boldsymbol{u}}_1^{(q-1)} \\ &= \left(K\left(\boldsymbol{T}\boldsymbol{W}^P\boldsymbol{T}^T\right)\odot\hat{\boldsymbol{R}}\right)\check{\boldsymbol{u}}_1^{(q-1)}.\end{aligned}$$

Thus, the decentralized computation of $\check{\boldsymbol{u}}_1$ using the DPM can be written as the following iteration

$$\check{\boldsymbol{u}}_1^{(q)} = \check{\boldsymbol{R}}\,\check{\boldsymbol{u}}_1^{(q-1)}, \tag{A.4}$$

where $\check{\boldsymbol{R}}$ is defined in (3.11). Note that (A.4) corresponds to the update procedure of the conventional PM applied to the matrix $\check{\boldsymbol{R}}$. Thus, after a sufficiently large number of PM iterations Q, the resulting vector $\check{\boldsymbol{u}}_1^{(Q)}$ converges (if normalized) to the principal eigenvector of the matrix $\check{\boldsymbol{R}}$. It follows from Assumption A1 that the decentralized normalization of $\check{\boldsymbol{u}}_1^{(Q)}$ is accurate. Thus, under Assumption A1, the vector resulting from applying the DPM to the sample covariance matrix $\hat{\boldsymbol{R}}$ is the principal eigenvector of the matrix $\check{\boldsymbol{R}}$ computed using the conventional PM. This concludes the first part of the induction.

For the second part of the induction, the vectors $\{\check{\boldsymbol{u}}_i\}_{i=1}^{l-1}$ computed using the DPM are assumed to be the first $(l-1)$ eigenvectors of the matrix $\check{\boldsymbol{R}}$. Then, the induction is proved for the vector $\check{\boldsymbol{u}}_l$.

The computation of the vector $\check{\boldsymbol{u}}_l$ using the DPM is achieved as follows. First, the vector $\check{\boldsymbol{u}}_l'^{(q)}$, which is defined in (2.60) is computed in a decentralized fashion. In analogy to (A.1), the vector $\check{\boldsymbol{u}}_l'^{(q)}$ can be rewritten as

$$\check{\boldsymbol{u}}_l'^{(q)} = \check{\boldsymbol{R}}\,\check{\boldsymbol{u}}_l^{(q-1)}. \tag{A.5}$$

Second, the scalars $\{\check{u}_{i,l}^{\prime(q)}\}_{i=1}^{l-1}$ are computed in a decentralized fashion. Since under Assumption A1 the AC errors resulting from this computation are negligible, the decentralized iteration used to compute the vector $\check{\boldsymbol{u}}_l$ is reduced to

$$\check{\boldsymbol{u}}_l^{(q)} = (\boldsymbol{I}_M - \check{\boldsymbol{U}}_{l-1}\check{\boldsymbol{U}}_{l-1}^H)\check{\boldsymbol{R}}\,\check{\boldsymbol{u}}_l^{(q-1)}. \tag{A.6}$$

Note that (A.6) is equivalent to the conventional PM iteration (2.51) applied to compute the lth eigenvector of the matrix $\check{\boldsymbol{R}}$. After Q iterations of the DPM, the resulting vector $\check{\boldsymbol{u}}_l^{(Q)}$ is normalized. Again under Assumption A1, the normalization is accurate, thus, the resulting normalized vector $\check{\boldsymbol{u}}_l$ is the lth eigenvector of the matrix $\check{\boldsymbol{R}}$ computed using the conventional PM. □

Part II: $\{\check{\lambda}_i(P)\}_{i=1}^M$ are the eigenvalues of the matrix $\check{\boldsymbol{R}}(P)$

Proof. In order to prove this part of Theorem 1, the results of Part I is used to show that the decentralized computation of the eigenvalue $\check{\lambda}_l(P)$ as defined in (2.69) yields the lth eigenvalue of the matrix $\check{\boldsymbol{R}}(P)$. For notation convenience, the dependency on P is dropped from $\check{\boldsymbol{R}}(P)$, $\check{\boldsymbol{u}}_l(P)$, and $\check{\lambda}_l(P)$ throughout the derivations.

The eigenvalue $\check{\lambda}_l$ using the DPM is computed as follows. First, the vector $\check{\boldsymbol{u}}_l'$, as defined in 2.70, is computed in decentralized fashion using P AC iterations. Thus, in analogy to (A.1) and (A.5), this vector can be rewritten as $\check{\boldsymbol{u}}_l' = \check{\boldsymbol{R}}\,\check{\boldsymbol{u}}_l$. Secondly, the scalar product in (2.71) is computed in decentralized fashion using P_3 AC iterations. According to Assumption A1 this computation is accurate. Consequently, the computation of $\check{\lambda}_l$ using the DPM can be written as

$$\check{\lambda}_l = \check{\boldsymbol{u}}_l^H \check{\boldsymbol{R}} \check{\boldsymbol{u}}_l. \tag{A.7}$$

Since the vector $\check{\boldsymbol{u}}_l$ is the lth eigenvector of the matrix $\check{\boldsymbol{R}}$, the right-hand-side of (A.7) is the lth eigenvalue of $\check{\boldsymbol{R}}$ which concludes the proof. □

A.2 Proof of Theorem 2

Proof. In order to prove Theorem 2, the matrix $\check{\boldsymbol{R}}(P)$ is written in terms of $\hat{\boldsymbol{R}}$ and $\boldsymbol{W}$. Then, a first order analysis is carried out. For convenience, the dependency on P is droped from $\check{\boldsymbol{R}}(P)$, $\check{\boldsymbol{u}}_l(P)$, $\check{\lambda}_l(P)$, and $\boldsymbol{\kappa}_l(P)$ throughout the proof.

The largest eigenvalue of the AC weighting matrix $\boldsymbol{W}$ is $\alpha_1 = 1$ and its corresponding eigenvector is $\boldsymbol{\beta}_1 = \frac{1}{\sqrt{K}}\mathbf{1}_K$, see Section 2.4.2, thus

$$\begin{aligned}\check{\boldsymbol{R}} &= K\left(\boldsymbol{T}\boldsymbol{W}^P\boldsymbol{T}^T\right)\odot\hat{\boldsymbol{R}}\\ &= K\left(\boldsymbol{T}\Big(\sum_{k=1}^{K}\alpha_k^P\boldsymbol{\beta}_k\boldsymbol{\beta}_k^H\Big)\boldsymbol{T}^T\right)\odot\hat{\boldsymbol{R}}\\ &= \hat{\boldsymbol{R}} + K\left(\sum_{k=2}^{K}\alpha_k^P\boldsymbol{T}\boldsymbol{\beta}_k\boldsymbol{\beta}_k^H\boldsymbol{T}^T\right)\odot\hat{\boldsymbol{R}}\\ &= \hat{\boldsymbol{R}} + K\sum_{k=2}^{K}\alpha_k^P\tilde{\boldsymbol{T}}_k\,\hat{\boldsymbol{R}}\,\tilde{\boldsymbol{T}}_k^H,\end{aligned} \tag{A.8}$$

where $\tilde{\boldsymbol{T}}_k = \operatorname{diag}\big(\boldsymbol{T}\boldsymbol{\beta}_k\big)$ and, for the last equality, the rank one Hadamard product property [Joh90, p. 104] is used. Note that the second term in (A.8) accounts for the errors resulting from the finite number of AC iterations $P < \infty$, and that $\lim_{P\to\infty}\check{\boldsymbol{R}} \to \hat{\boldsymbol{R}}$. Substituting (3.16) in (A.8) yields

$$\begin{aligned}\check{\boldsymbol{R}} &= \boldsymbol{R} + \delta\boldsymbol{R} + K\sum_{k=2}^{K}\alpha_k^P\tilde{\boldsymbol{T}}_k\left(\boldsymbol{R}+\delta\boldsymbol{R}\right)\tilde{\boldsymbol{T}}_k^H\\ &\approx \boldsymbol{R} + \delta\boldsymbol{R} + K\sum_{k=2}^{K}\alpha_k^P\tilde{\boldsymbol{T}}_k\boldsymbol{R}\tilde{\boldsymbol{T}}_k^H\end{aligned} \tag{A.9}$$

where the term $\sum_{k=2}^{K}\alpha_k^P\,\tilde{\boldsymbol{T}}_k\,\delta\boldsymbol{R}\,\tilde{\boldsymbol{T}}_k^H$ is neglected in the approximation since the magnitudes of $\alpha_2,\ldots,\alpha_K$ are all strictly smaller than one (see Section 2.4.2) and they are multiplied with the small variation $\delta\boldsymbol{R}$.

Multiplying (A.9) from the right with $\check{\boldsymbol{u}}_l$ and keeping the first order terms, results in

$$\begin{aligned}\check{\boldsymbol{R}}\,\check{\boldsymbol{u}}_l &\approx \left(\boldsymbol{R} + \delta\boldsymbol{R} + K\sum_{k=2}^{K}\alpha_k^P\tilde{\boldsymbol{T}}_k\boldsymbol{R}\tilde{\boldsymbol{T}}_k^H\right)\left(\boldsymbol{u}_l + \delta\check{\boldsymbol{u}}\right)\\ &\approx \boldsymbol{R}\boldsymbol{u}_l + \delta\boldsymbol{R}\boldsymbol{u}_l + \boldsymbol{\kappa}_l + \boldsymbol{R}\delta\check{\boldsymbol{u}},\end{aligned} \tag{A.10}$$

where $\boldsymbol{\kappa}_l$ is defined in (3.20). The left hand side of (A.10) can be written as

$$\check{\boldsymbol{R}}\,\check{\boldsymbol{u}}_l = \check{\lambda}_l\,\check{\boldsymbol{u}}_l, \tag{A.11}$$

where $\check{\lambda}_l$ is the lth eigenvalue of $\check{\boldsymbol{R}}$. Substituting (3.15) and (3.13) in (A.11) yields

$$\begin{aligned}\check{\boldsymbol{R}}\,\check{\boldsymbol{u}}_l &= \left(\lambda_l + \delta\check{\lambda}_l\right)\left(\boldsymbol{u}_l + \delta\check{\boldsymbol{u}}_l\right)\\ &\approx \lambda_l\boldsymbol{u}_l + \delta\check{\lambda}_l\boldsymbol{u}_l + \lambda_l\delta\check{\boldsymbol{u}}_l,\end{aligned} \tag{A.12}$$

where only first order terms are kept. Substituting (A.12) in (A.10) results in

$$\left(\boldsymbol{R} - \lambda_l\boldsymbol{I}_M\right)\delta\check{\boldsymbol{u}}_l \approx \delta\check{\lambda}_l\boldsymbol{u}_l - \delta\boldsymbol{R}\boldsymbol{u}_l - \boldsymbol{\kappa}_l. \tag{A.13}$$

The matrix $(\boldsymbol{R} - \lambda_l \boldsymbol{I}_M)$ can be written as

$$\begin{aligned} \boldsymbol{R} - \lambda_l \boldsymbol{I}_M &= \sum_{k=1, k\neq l}^{M} (\lambda_k - \lambda_l)\, \boldsymbol{u}_l \boldsymbol{u}_k^H \\ &= \boldsymbol{U}_{-l} \boldsymbol{\Gamma}_{-l} \boldsymbol{U}_{-l}^H \end{aligned} \tag{A.14}$$

where $\boldsymbol{U}_{-l}$ and $\boldsymbol{\Gamma}_{-l}$ are defined in (2.13). Thus multiplying (A.13) by $\boldsymbol{\Xi}_l = \boldsymbol{U}_{-l} \boldsymbol{\Gamma}_{-l}^{-1} \boldsymbol{U}_{-l}^H$ yields

$$\begin{aligned} \delta\check{\boldsymbol{u}}_l &= \boldsymbol{\Xi}_l \left(\delta\check{\lambda}_l \boldsymbol{u}_l - \delta\boldsymbol{R}\boldsymbol{u}_l - \boldsymbol{\kappa}_l \right) \\ &= -\boldsymbol{\Xi}_l \left(\delta\boldsymbol{R}\boldsymbol{u}_l + \boldsymbol{\kappa}_l \right), \end{aligned} \tag{A.15}$$

which proves (3.18).

In the following, (3.19) is proven. Substituting (A.14) in (A.13) yields

$$\boldsymbol{U}_{-l} \boldsymbol{\Gamma}_{-l} \boldsymbol{U}_{-l}^H \delta\check{\boldsymbol{u}}_l = \delta\check{\lambda}_l \boldsymbol{u}_l - \delta\boldsymbol{R}\boldsymbol{u}_l - \boldsymbol{\kappa}_l. \tag{A.16}$$

Multiplying (A.16) with $\check{\boldsymbol{u}}_l^H$ from the left and noting that $\boldsymbol{u}_l^H \boldsymbol{U}_{-l} = \boldsymbol{0}$ gives

$$\delta\check{\lambda}_l = \boldsymbol{u}_l^H \delta\boldsymbol{R}\boldsymbol{u}_l + \boldsymbol{u}_l^H \boldsymbol{\kappa}_l, \tag{A.17}$$

which concludes the proof of Theorem 2

□

A.3 Proof of Theorem 3

Proof. In order to prove Theorem 3, we compute $\mathbb{E}\Big(\delta\check{\boldsymbol{u}}_l(P)\delta\check{\boldsymbol{u}}_m^H(P)\Big)$, $\mathbb{E}\Big(\delta\check{\boldsymbol{u}}_l(P)\delta\check{\boldsymbol{u}}_m^T(P)\Big)$, and $\mathbb{E}\Big(\delta\check{\lambda}_l(P)\delta\check{\lambda}_m^*(P)\Big)$ based on the expressions of $\delta\check{\boldsymbol{u}}_l(P)$ $\delta\check{\lambda}_l(P)$ which are derived in Theorem 2. Then, results from [Bri81] are used to simplify the computed expression. For notation convenience, the dependency on P is dropped from $\check{\boldsymbol{u}}_l(P)$,$\check{\lambda}_l(P)$ and $\boldsymbol{\kappa}_l(P)$ throughout the proof.

Using the result from Theorem 2, $\mathbb{E}\Big(\delta\check{\boldsymbol{u}}_l(P)\delta\check{\boldsymbol{u}}_m^H(P)\Big)$, $\mathbb{E}\Big(\delta\check{\boldsymbol{u}}_l(P)\delta\check{\boldsymbol{u}}_m^T(P)\Big)$, and $\mathbb{E}\Big(\delta\check{\lambda}_l(P)\delta\check{\lambda}_m^*(P)\Big)$ are written as

$$\begin{aligned} \mathbb{E}\Big(\delta\check{\boldsymbol{u}}_l(P)\delta\check{\boldsymbol{u}}_m^H(P)\Big) &= \boldsymbol{\Xi}_l \mathbb{E}\Big((\delta\boldsymbol{R}\boldsymbol{u}_l + \boldsymbol{\kappa}_l)(\boldsymbol{u}_m^H \delta\boldsymbol{R} + \boldsymbol{\kappa}_m^H)\Big)\boldsymbol{\Xi}_m^H \\ &= \mathbb{E}\big(\boldsymbol{\Xi}_l \delta\boldsymbol{R}\boldsymbol{u}_l \boldsymbol{u}_m^H \delta\boldsymbol{R}\boldsymbol{\Xi}_m^H\big) + \boldsymbol{\Xi}_l \boldsymbol{\kappa}_l \boldsymbol{\kappa}_m^H \boldsymbol{\Xi}_m^H, \end{aligned} \tag{A.18}$$

$$\begin{aligned}\mathbb{E}\Big(\delta\check{\boldsymbol{u}}_l(P)\delta\check{\boldsymbol{u}}_m^T(P)\Big) &= \boldsymbol{\Xi}_l\mathbb{E}\Big((\delta\boldsymbol{R}\boldsymbol{u}_l+\boldsymbol{\kappa}_l)(\boldsymbol{u}_m^T\delta\boldsymbol{R}+\boldsymbol{\kappa}_m^T)\Big)\boldsymbol{\Xi}_m^T \\ &= \mathbb{E}\big(\boldsymbol{\Xi}_l\delta\boldsymbol{R}\boldsymbol{u}_l\boldsymbol{u}_m^T\delta\boldsymbol{R}\boldsymbol{\Xi}_m^T\big)+\boldsymbol{\Xi}_l\boldsymbol{\kappa}_l\boldsymbol{\kappa}_m^T\boldsymbol{\Xi}_m^T,\end{aligned} \tag{A.19}$$

and

$$\mathbb{E}\Big(\delta\check{\lambda}_l(P)\delta\check{\lambda}_m^*(P)\Big) = \mathbb{E}\Big(\boldsymbol{u}_l^H\delta\boldsymbol{R}\boldsymbol{u}_l\boldsymbol{u}_m^H\delta\boldsymbol{R}\boldsymbol{u}_m\Big)+\boldsymbol{u}_l^H\boldsymbol{\kappa}_l\boldsymbol{\kappa}_m^H\boldsymbol{u}_m. \tag{A.20}$$

Using the following results from Theorem 9.2.2 in [Bri81][1]

$$\mathbb{E}\big(\boldsymbol{\Xi}_l\delta\boldsymbol{R}\boldsymbol{u}_l\boldsymbol{u}_m^H\delta\boldsymbol{R}\boldsymbol{\Xi}_m^H\big) = \frac{\lambda_l}{N}\sum_{\substack{i=1\\ i\neq l}}^{M}\frac{\lambda_i}{(\lambda_l-\lambda_i)^2}\boldsymbol{u}_l\boldsymbol{u}_l^H\delta_{l,m}, \tag{A.21}$$

$$\mathbb{E}\big(\boldsymbol{\Xi}_l\delta\boldsymbol{R}\boldsymbol{u}_l\boldsymbol{u}_m^T\delta\boldsymbol{R}\boldsymbol{\Xi}_m^T\big) = \frac{\lambda_l\lambda_m}{N}\frac{\boldsymbol{u}_l\boldsymbol{u}_m^T}{(\lambda_l-\lambda_m)^2}(\delta_{l,m}-1), \tag{A.22}$$

and

$$\mathbb{E}\big(\boldsymbol{u}_l^H\delta\boldsymbol{R}\boldsymbol{u}_l\boldsymbol{u}_m^H\delta\boldsymbol{R}\boldsymbol{u}_m\big) = \frac{\lambda_l^2}{N}\delta_{l,m} \tag{A.23}$$

in (A.18), (A.19), and (A.20) proves the theorem. □

A.4 Proof of Theorem 4

Proof. The proof of Theorem 4 consists in showing the sufficiency of the condition (5.15). For fully calibrated arrays using coherent processing a bound on the maximum number of identifiable sources is introduced in [HN96]. This bound is not applicable in the case considered in this dissertation since in [HN96] the covariance matrix of the whole array is assumed to be available and thus the bound is introduced using the rank of the matrix $\boldsymbol{V}$ and not $\check{\boldsymbol{V}}$ as considered here. The proof of the bound is similar in spirit to that of [HN96].

In this section, it is proven that if $\check{\boldsymbol{V}}(\boldsymbol{\theta})\boldsymbol{p} = \check{\boldsymbol{V}}(\boldsymbol{\theta}')\boldsymbol{p}'$ and $L \leq \lfloor\frac{\rho}{2}\rfloor$ then $\boldsymbol{\theta} = \boldsymbol{\theta}'$. Assume that there are $q \leq L \leq \lfloor\frac{\rho}{2}\rfloor$ entries which occur in both DOA vectors $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$. Then, $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$ can be partitioned as $\boldsymbol{\theta} = [\boldsymbol{\theta}_1^T, \boldsymbol{\theta}_2^T]^T$ and $\boldsymbol{\theta}' = [\boldsymbol{\theta}_1'^T, \boldsymbol{\theta}_2'^T]^T$ such that $\boldsymbol{\theta}_1 = \boldsymbol{\theta}_1' \in \mathbb{R}^{q\times 1}$ and that the DOAs $\boldsymbol{\theta}_2$ and $\boldsymbol{\theta}_2'$ are all different. Moreover, let $\boldsymbol{p} = [\boldsymbol{p}_1^T, \boldsymbol{p}_2^T]^T$ and $\boldsymbol{p}' = [\boldsymbol{p}_1'^T, \boldsymbol{p}_2'^T]^T$, where $\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_1'$, and $\boldsymbol{p}_2'$ contain the power of the sources corresponding to the DOAs $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \boldsymbol{\theta}_1'$ and $\boldsymbol{\theta}_2'$, respectively. Thus, the assumption that $\check{\boldsymbol{V}}(\boldsymbol{\theta})\boldsymbol{p} = \check{\boldsymbol{V}}(\boldsymbol{\theta}')\boldsymbol{p}'$ can be written as

$$[\check{\boldsymbol{V}}(\boldsymbol{\theta}_1), \check{\boldsymbol{V}}(\boldsymbol{\theta}_2)][\boldsymbol{p}_1^T, \boldsymbol{p}_2^T]^T = [\check{\boldsymbol{V}}(\boldsymbol{\theta}_1'), \check{\boldsymbol{V}}(\boldsymbol{\theta}_2')][\boldsymbol{p}_1'^T, \boldsymbol{p}_2'^T]^T. \tag{A.24}$$

[1] See also the proof of the Theorem 9.2.2 [Bri81, p. 454].

Since $\breve{\boldsymbol{V}}(\boldsymbol{\theta}_1) = \breve{\boldsymbol{V}}(\boldsymbol{\theta}'_1)$, (A.24) can be rearranged as

$$[\breve{\boldsymbol{V}}(\boldsymbol{\theta}_1), \breve{\boldsymbol{V}}(\boldsymbol{\theta}_2), \breve{\boldsymbol{V}}(\boldsymbol{\theta}'_2)][\boldsymbol{p}_1^T - \boldsymbol{p}_1'^T, \boldsymbol{p}_2^T, -\boldsymbol{p}_2'^T]^T = 0. \tag{A.25}$$

Next, the following two cases are distinguished:

1. $q = L$: In this case $\boldsymbol{\theta}_1 = \boldsymbol{\theta} = \boldsymbol{\theta}'$ and $\boldsymbol{p} = \boldsymbol{p}'$ is a unique solution to (A.25), i.e., in this case the DOAs are uniquely identifiable.

2. $q < L$: In this case, the matrix $[\breve{\boldsymbol{V}}(\boldsymbol{\theta}_1), \breve{\boldsymbol{V}}(\boldsymbol{\theta}_2), \breve{\boldsymbol{V}}(\boldsymbol{\theta}'_2)]$ contains $2L - q$ columns corresponding to different DOAs. Since $q < L$ and $L \leq \lfloor \frac{\rho}{2} \rfloor$ the inequality $2L - q \leq 2L \leq \rho$ holds. Consequently, the matrix $[\breve{\boldsymbol{V}}(\boldsymbol{\theta}_1), \breve{\boldsymbol{V}}(\boldsymbol{\theta}_2), \breve{\boldsymbol{V}}(\boldsymbol{\theta}'_2)]$ is full rank and (A.25) can only be satisfied, in this case, if $[\boldsymbol{p}_1^T - \boldsymbol{p}_1'^T, \boldsymbol{p}_2^T, -\boldsymbol{p}_2'^T]^T = 0$. However, this is not possible since it implies that $\boldsymbol{p}'_2 = \boldsymbol{p}_2 = 0$, i.e., the sources corresponding to the DOAs $\boldsymbol{\theta}_2$ and $\boldsymbol{\theta}'_2$ have zero power.

Thus, (A.25) can only be satisfied in case 1) which proves the theorem.

□

List of Acronyms

AC	Averaging consensus
CR	Common root
CRB	The Cramér-Rao Bound
cvx	MATLAB software for disciplined convex programming [CR12]
DESPRIT	The decentralized ESPRIT algorithm in Chapter 3
DGESPRIT	The decentralized generalized eigendecomposition-based ESPRIT algorithms in Chapter 4
D-Lanczos	The decentralized Lanczos method in Chapter 4
DOA	Direction-of-Arrival
DOF	Degree of freedom
DPM	The decentralized power method in Chapter 2
ED	Energy detector
ESPRIT	Estimation of signal parameters via rotational invariance techniques [RK89]
FAR	False alarm rate
FIM	Fisher information matrix
GESPRIT	Generalized eigendecomposition-based ESPRIT algorithms in Chapter 4
GLRT	Generalized likelihood ratio test
GSM	Generalized Sylvester matrix
IDESPRIT	Interpolated decentralized ESPRIT in Chapter 4
LDESPRIT	Decentralized Lanczos-based ESPRIT algorithm in Chapter 4
LS	Least Squares
MDL	Minimum description length [WK85b]
MLE	Maximum Likelihood estimator

MODE Method of direction estimation [SN90]

MSE Mean Square Error

MUSIC The MUltiple SIgnal Classification algorithm [Sch86]

NP2 The natural power method a second implementation [HXC+99]

PC Processing Center

PM The power method in Chapter 2

RARE RAnk Reduction Estimator [PGW02]

RMSE Root Mean Square Error

Root-MUSIC The rooting based MUSIC algorithm [Bar83]

SN Sensing node

SNR Signal to noise ration

SPICE The SParse Iterative Covariance-based Estimation algorithm [SBL11]

SSR Sparse signal representation

UC Unit circle

UL Uniform linear

ULA Uniform linear array

WSF Weighted subspace fitting [VOK91]

WSN Wireless sensor network

List of Symbols

Symbols and Operators Used Across the Thesis

$\hat{(\cdot)}$	Centralized estimate
$\check{(\cdot)}$	Decentralized estimate
$\check{(\cdot)}_k$	Decentralized estimate available only at the kth subarray
$\check{(\cdot)}_{[k]}$	Decentralized estimate at the kth subarray, other subarrays also estimate the same parameter, refer to Section 2.4.2
$(\cdot)^T$	Transpose operator
$(\cdot)^*$	Complex conjugate
$(\cdot)^H$	Hermitian operator
$(\cdot)^{-1}$	Inverse
$\lvert\cdot\rvert$	Magnitude of scalars or determinant of matrices
$\lVert\cdot\rVert_1$	ℓ_1 norm
$\lVert\cdot\rVert_2$	Euclidean norm
$\lVert\cdot\rVert_{\mathrm{F}}$	Frobenius matrix norm
$\lVert\cdot\rVert_{p,q}$	$\ell_{p,q}$ mixed norm
$\mathbb{R}$	The set of real numbers
$\mathbb{C}$	The set of complex numbers
$\jmath$	The imaginary unit
tr	Trace of a matrix
rank	Rank of a matrix
$\circ$	Katri-Rao product
$\otimes$	Kronecker product
$\odot$	Hadamard (element-wise) product
$\mathbb{E}$	The expectation operator
diag	Diagonal matrix
blkdiag	Block diagonal matrix
$(\cdot) \to \infty$	Approach infinity
$\lim_{i\to\infty}(\cdot)$	Limit when i approaches infinity
vec	Vectorization of a matrix
$\delta_{i,j}$	Kronecker delta function
card	Cardinality of a set
$\lfloor\cdot\rfloor$	The floor operator

$\boldsymbol{I}_i$	The identity matrix of size $i \times i$
$\mathbf{0}_i$	Zero vector of length i
$\mathbf{1}_i$	Ones vector of length i
$\mathcal{L}(\cdot)$	The negative log-likelihood function
$\frac{df(x)}{dx}$	Derivative of f with respect to x
$\mathcal{N}_k$	Set of neighbors of the kth subarray
K	The number of subarrays
L	The number of sources
M_k	The number of sensors at the kth subarray
M	The number of sensors at the whole array
N	The number of samples
$\boldsymbol{W}$	AC weighting matrix
$\alpha_i, \boldsymbol{\beta}_i$	The ith eigenvalue and eigenvector of $\boldsymbol{W}$
P, P_1, P_2, P_3	Number of AC iterations used in the DPM algorithm
P_4	Number of AC iterations used in the DESPRIT algorithm
$\boldsymbol{T}$	Sensor selection matrix
$\overline{\boldsymbol{J}}, \underline{\boldsymbol{J}}$	Upper and lower selection matrices defined in Chapter 2
$\boldsymbol{\theta}$	DOA of the sources
$\boldsymbol{\nu}(\theta)$	sin and cos of θ
$\tilde{\boldsymbol{\theta}}$	Directions grid
G	Length of $\tilde{\boldsymbol{\theta}}$
λ_c	Carrier frequency
$\boldsymbol{\zeta}_k$	Displacement of the first sensor of the kth subarray with respect to the first sensor in the first subarray
$\boldsymbol{\zeta}'_{k,i}$	Location of the ith sensor of the kth subarray with respect to its first sensor
$\overline{d}$	Uniform spacing between sensors of ULA
d	Distance between the upper and lower groups of sensors
$\boldsymbol{a}, \boldsymbol{A}, \overline{\boldsymbol{A}}, \underline{\boldsymbol{A}}$	Array manifold vector, array manifold matrix, upper- , and lower-array manifold matrices
$\tilde{\boldsymbol{A}}$	Overcomplete dictionary $\tilde{\boldsymbol{A}} = \boldsymbol{A}(\tilde{\boldsymbol{\theta}})$
$\boldsymbol{v}, \boldsymbol{V}$	Subarray manifold vector and matrix
$\tilde{\boldsymbol{V}}$	Overcomplete dictionary $\tilde{\boldsymbol{V}} = \boldsymbol{V}(\tilde{\boldsymbol{\theta}})$
$\phi(\theta_l, \boldsymbol{\zeta}_k)$ or $\phi_{k,l}$	Unknown phase at the kth subarray corresponding to the direction θ_l
$\boldsymbol{\Phi}_k(\boldsymbol{\theta}, \boldsymbol{\zeta}_k)$	Diagonal matrix $\mathrm{diag}(\phi(\theta_1, \boldsymbol{\zeta}_k), \ldots, \phi(\theta_L, \boldsymbol{\zeta}_k))$
$\boldsymbol{s}(t)$	Source signals at time t

$\tilde{\boldsymbol{s}}(t)$	Sparse representation of $\boldsymbol{s}(t)$
$\boldsymbol{n}(t)$	Additive noise at time t
σ^2	Variance of the noise
$\boldsymbol{x}, \overline{\boldsymbol{x}}, \underline{\boldsymbol{x}}$	Measurement-, upper-, lower-, measurement vectors
$\boldsymbol{P}$	Source covariance matrix
p_i	Power of the ith source
$\boldsymbol{R}$	True measurement covariance matrix
$\tilde{\boldsymbol{R}}$	Overcomplete parametrization of $\boldsymbol{R}$
$\lambda_i, \boldsymbol{u}_i$	The ith eigenvalue and the corresponding eigenvector of $\boldsymbol{R}$
$\boldsymbol{\Lambda}, \boldsymbol{U}$	All eigenvalues and eigenvectors of the matrix $\boldsymbol{R}$
$\boldsymbol{\Lambda}_\mathrm{s}, \boldsymbol{U}_\mathrm{s}$	Signal eigenvalues and eigenvectors
$\boldsymbol{\Lambda}_\mathrm{n}, \boldsymbol{U}_\mathrm{n}$	Noise eigenvalues and eigenvectors
$\overline{\boldsymbol{U}}_\mathrm{s}, \underline{\boldsymbol{U}}_\mathrm{s}$	Upper and lower parts of the signal eigenvectors
$\boldsymbol{\Pi}_\mathrm{s}, \boldsymbol{\Pi}_\mathrm{n}$	Projection into the signal and noise eigenvectors
$\hat{\boldsymbol{R}}$	Sample covariance matrix
$\hat{\lambda}_i, \hat{\boldsymbol{u}}_i, \hat{\boldsymbol{\Lambda}}, \hat{\boldsymbol{U}}$	Centralized sample estimate of $\lambda_i, \boldsymbol{u}_i, \boldsymbol{\Lambda}, \boldsymbol{U}$
$\hat{\boldsymbol{\Lambda}}_\mathrm{s}, \hat{\boldsymbol{U}}_\mathrm{s}, \hat{\boldsymbol{\Lambda}}_\mathrm{n}, \hat{\boldsymbol{U}}_\mathrm{n}$	Centralized sample estimate of $\boldsymbol{\Lambda}_\mathrm{s}, \boldsymbol{U}_\mathrm{s}, \boldsymbol{\Lambda}_\mathrm{n}, \boldsymbol{U}_\mathrm{n}$
$\hat{\overline{\boldsymbol{U}}}_\mathrm{s}, \hat{\underline{\boldsymbol{U}}}_\mathrm{s}$	Centralized sample estimate of $\overline{\boldsymbol{U}}_\mathrm{s}, \underline{\boldsymbol{U}}_\mathrm{s}$
$\hat{\boldsymbol{\Pi}}_\mathrm{s}, \hat{\boldsymbol{\Pi}}_\mathrm{n}$	Centralized sample estimate of $\boldsymbol{\Pi}_\mathrm{s}, \boldsymbol{\Pi}_\mathrm{n}$
$\hat{\boldsymbol{U}}_l$	Centralized estimates of the first l eigenvectors
$\hat{\mathcal{F}}(z)$	Centralized estimate of the Root-MUSIC polynomial
$\check{\lambda}_i, \check{\boldsymbol{u}}_i$	Decentralized sample estimate of $\lambda_i, \boldsymbol{u}_i$
$\check{\boldsymbol{U}}_\mathrm{s}, \check{\overline{\boldsymbol{U}}}_\mathrm{s}, \check{\underline{\boldsymbol{U}}}_\mathrm{s}$	Decentralized sample estimate of $\boldsymbol{U}_\mathrm{s}, \overline{\boldsymbol{U}}_\mathrm{s}, \underline{\boldsymbol{U}}_\mathrm{s}$
$\check{\boldsymbol{\Pi}}_\mathrm{n}$	Decentralized sample estimate of $\boldsymbol{\Pi}_\mathrm{n}$
$\check{\boldsymbol{U}}_l$	Decentralized sample estimates of the first l eigenvectors
$\check{\mathcal{F}}(z)$	Decentralized Root-MUSIC polynomial
$\boldsymbol{\Omega}$	Refer to (2.43)
$\mathcal{S}_\mathrm{MUSIC}, \mathcal{S}_{\ell_{p,q}}$	Spatial spectrum of MUSIC and $\ell_{p,q}$ mixed norm
z	Parameter of the Root-MUSIC polynomial
$\hat{z}_i$	The i root of the estimated Root-MUSIC polynomial
$\check{z}_i$	The i root of the decentralized computation of the Root-MUSIC polynomial
$\boldsymbol{X}$	Measurements matrix
$\tilde{\boldsymbol{S}}$	Sparse source signal matrix
$\boldsymbol{N}$	Noise matrix

$\tilde{\boldsymbol{P}}, \tilde{p}_i$	Sparse diagonal representation of $\boldsymbol{P}$, the ith diagonal element of $\tilde{\boldsymbol{P}}$
$\omega_i, \bar{\omega}$	Weighting factors for SPICE
$\boldsymbol{\Psi}$	Used in the ESPRIT algorithm, refer to (2.47)
$\psi_i, \overleftarrow{\boldsymbol{\psi}}_i, \overrightarrow{\boldsymbol{\psi}}_i$	The ith eigenvalue, left-, right-, eigenvector of $\boldsymbol{\Psi}$
$\breve{\boldsymbol{R}}$	Refer to (3.11)
$\delta\boldsymbol{u}_l, \delta\lambda_l, \delta\boldsymbol{R}$	Finite sample size errors in $\boldsymbol{u}_l, \lambda_l, \boldsymbol{R}$
$\delta\breve{\boldsymbol{u}}_l, \delta\breve{\lambda}_l, \delta\breve{\boldsymbol{R}}$	AC and finite sample size errors in $\boldsymbol{u}_l, \lambda_l, \boldsymbol{R}$
$\boldsymbol{\Xi}_i, \boldsymbol{\Gamma}_i, \boldsymbol{U}_{-i}$	Refer to (2.13)
$\tilde{\boldsymbol{T}}_k, \boldsymbol{\kappa}_l$	Refer to Theorem 2
$\boldsymbol{\gamma}_i, \boldsymbol{\mu}_i$	Refer to (3.27)
$\dot{\boldsymbol{a}}$	Derivative of $\boldsymbol{a}$ with respect to θ

Symbols Used in Chapter 3 and Chapter 4

$\overline{\boldsymbol{A}}$	Virtual array response
$\boldsymbol{\mathfrak{I}}$	Interpolation matrix
$\tilde{\boldsymbol{\Omega}}$	A grid representation of $\boldsymbol{\Omega}$
$\overline{\boldsymbol{u}}_i$	The ith eigenvector of the virtual array covariance matrix
$s(t)$	Signal of a single source at time t
$T_{\mathrm{GLRT}}, \eta_{\mathrm{GLRT}}$	Test statistics and threshold of the GLRT
$T_{\mathrm{ED}}, \eta_{\mathrm{ED}}$	Test statistics and threshold of the ED in centralized scheme
$T_{\mathrm{ED}}^{(n)}, \eta_{\mathrm{ED}}^{(n)}$	Test statistics and threshold for the nth source, refer to Algorithm 3
$T_{\mathrm{ED},k}$	Refer to 3.43
γ	FAR
$\mathcal{P}(E_1\|E_2)$	Probability of E_1 given E_2
$\breve{T}_{\mathrm{ED},[k]}$	Decentralized computation of T_{ED}
$\mathcal{H}_0, \mathcal{H}_1$	Null hypothesis, alternative hypothesis
$\breve{L}$	Number of iterations used in the Lanczos method
$\breve{\alpha}_i, \hat{\alpha}_i$	Scalars computed at the ith Lanczos iteration
$\breve{\boldsymbol{T}}$	Tridiagonal matrix defined in the Lanczos method
$\breve{t}_i, \breve{\boldsymbol{t}}_i$	the ith eigenvalue and the corresponding eigenvector of $\breve{\boldsymbol{T}}$
$\breve{\boldsymbol{U}}, \breve{\boldsymbol{u}}_i$	Intermediate variables in the Lanczos method
$\breve{P}, \breve{P}_1, \breve{P}_2$	Number of AC iterations used in D-Lanczos algorithm
$\overline{M}$	The number of sensors in the upper and lower groups
ε	Forgetting factor in NP2

$\overline{\boldsymbol{C}}, \underline{\boldsymbol{C}}$	Covariance matrices defined from the upper and lower measurements
$\overline{\boldsymbol{H}}, \overline{\boldsymbol{G}}$	Eigenvalues and eigenvectors of $\overline{\boldsymbol{C}}$
$\overline{g}_i, \overline{\boldsymbol{h}}_i$	The ith eigenvalue and the corresponding eigenvector of $\overline{\boldsymbol{C}}$
$\boldsymbol{G}, \boldsymbol{H}$	Eigenvalue and eigenvector matrices of $\overline{\boldsymbol{C}}^{-1}\underline{\boldsymbol{C}}$
$g_i, \boldsymbol{h}_i$	The ith eigenvalue and the corresponding eigenvector of $\overline{\boldsymbol{C}}^{-1}\underline{\boldsymbol{C}}$
$\hat{\overline{\boldsymbol{C}}}, \hat{\underline{\boldsymbol{C}}}$	Centralized estimate of $\overline{\boldsymbol{C}}, \underline{\boldsymbol{C}}$
$\hat{\overline{\boldsymbol{h}}}_i, \hat{\overline{\boldsymbol{G}}}, \hat{\overline{\boldsymbol{H}}}$	Centralized estimates of $\overline{\boldsymbol{h}}_i, \overline{\boldsymbol{G}}, \overline{\boldsymbol{H}}$
$\hat{\boldsymbol{G}}, \hat{\boldsymbol{H}}$	Centralized estimates of $\boldsymbol{G}, \boldsymbol{H}$
$\overline{\boldsymbol{F}}, \overline{\boldsymbol{Y}}, \overline{\boldsymbol{y}}, \overline{\boldsymbol{f}}$	Refer to (4.21), (4.22) and (4.24)
$\boldsymbol{F}, \boldsymbol{Y}$	Refer to (4.29)
$\underline{\boldsymbol{q}}, \boldsymbol{f}, \underline{\boldsymbol{y}}$	Refer to (4.30)
$\underline{\boldsymbol{n}}, \boldsymbol{q}$	Refer to (4.31)
$\eta_{\overline{\boldsymbol{x}}}(t), \eta_{\underline{\boldsymbol{q}}}(t)$	The scalar products $\overline{\boldsymbol{x}}^H(t)\overline{\boldsymbol{x}}(t)$ and $\underline{\boldsymbol{q}}^H(t)\underline{\boldsymbol{q}}(t)$
$\check{\overline{\boldsymbol{h}}}_i, \check{\overline{\boldsymbol{G}}}_{[k]}, \check{\overline{\boldsymbol{H}}}$	Decentralized estimate of $\overline{\boldsymbol{h}}_i, \overline{\boldsymbol{G}}, \overline{\boldsymbol{H}}$
$\check{\boldsymbol{G}}_{[k]}, \check{\boldsymbol{H}}$	Decentralized estimate of $\boldsymbol{G}$ and $\boldsymbol{H}$
$\check{\overline{\boldsymbol{F}}}_{[k]}, \check{\overline{\boldsymbol{Y}}}, \check{\overline{\boldsymbol{y}}}_{[k]}$	Decentralized estimate of $\overline{\boldsymbol{F}}$, $\overline{\boldsymbol{Y}}$, and $\overline{\boldsymbol{y}}$
$\check{\boldsymbol{F}}_{[k]}, \check{\boldsymbol{Y}}, \check{\overline{\boldsymbol{f}}}_{[k]}$	Decentralized estimate of $\boldsymbol{F}$, $\boldsymbol{Y}$, and $\overline{\boldsymbol{f}}$
$\check{\underline{\boldsymbol{q}}}, \check{\boldsymbol{f}}_{[k]}, \check{\underline{\boldsymbol{y}}}_{[k]}$	Decentralized estimate of $\underline{\boldsymbol{q}}$, $\boldsymbol{f}$, and $\underline{\boldsymbol{y}}$
$\check{\underline{\boldsymbol{n}}}_{[k]}, \check{\boldsymbol{q}}_{[k]}$	Decentralized estimate of $\underline{\boldsymbol{n}}$ and $\boldsymbol{q}$
$\check{\eta}_{\overline{\boldsymbol{x}},[k]}(t)$	Decentralized computation of $\overline{\boldsymbol{x}}^H(t)\overline{\boldsymbol{x}}(t)$
$\check{\eta}_{\underline{\boldsymbol{q}},[k]}(t)$	Decentralized computation of $\underline{\boldsymbol{q}}^H(t)\underline{\boldsymbol{q}}(t)$

Symbols and Operators Used in Chapter 5

τ_k	Time offset at the kth subarray
$\boldsymbol{r}$	Vectorized covariance matrix
$\check{M}$	Length of $\boldsymbol{r}$
$\hat{\boldsymbol{r}}$	Sample estimate of $\boldsymbol{r}$
$\boldsymbol{i}$	Vectorization of the identity matrices of the subarrays
$\check{\boldsymbol{V}}$	Co-array manifold
$\boldsymbol{p}$	The diagonal of the source covariance matrix $\boldsymbol{P}$
$\not\equiv$	Refer to Section 5.3.1
ρ	Kruskal rank of the co-array manifold matrix $\check{\boldsymbol{V}}$
$\check{\boldsymbol{R}}$	Block diagonal matrix in (5.25)
$\boldsymbol{\Delta}_1, \boldsymbol{\Delta}_2$	Derivative matrices used in the CRB

$\hat{\tilde{\boldsymbol{P}}}$	Sample estimate of $\tilde{\boldsymbol{P}}$
$\boldsymbol{\Upsilon}$	Source correlation matrix
ϵ	Source correlation factor
$\overline{\hat{\mathcal{F}}}, \underline{\hat{\mathcal{F}}}$	The two factors of the Root-MUSIC polynomial
$\hat{\kappa}_{k,i}$	Coefficients of the Root-MUSIC polynomial
$\mathscr{F}$	The set of subarray polynomials
$\boldsymbol{\mathcal{S}}$	The GSM
r, h	Dimensions of the GSM
$\mathcal{O}(\boldsymbol{\mathcal{S}})$	Null space of the GSM
$\boldsymbol{e}_i$	The ith right singular eigenvector of $\boldsymbol{\mathcal{S}}$
$\boldsymbol{E}_n$	Concatenation of the n principal eigenvectors of $\boldsymbol{\mathcal{S}}$
$\overline{\boldsymbol{E}}_n, \underline{\boldsymbol{E}}_n$	Upper and lower parts of $\boldsymbol{E}_n$
$\overline{\boldsymbol{\Psi}}_n$	Refer to (5.42)

Bibliography

[ABB+99] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen. (1999) LAPACK user's guide third edition. [Online]. Available: http://www.netlib.org/lapack/lug/node48.html

[AGGS98] Y. I. Abramovich, D. A. Gray, A. Y. Gorokhov, and N. K. Spencer, "Positive-definite Toeplitz completion in DOA estimation for nonuniform linear antenna arrays. I. Fully augmentable arrays," *IEEE Transactions on Signal Processing*, vol. 46, no. 9, pp. 2458–2471, 1998.

[AK12] M. Atashbar and M. H. Kahaei, "Direction-of-arrival estimation using AMLSS method," *IEEE Latin America Transactions*, vol. 10, no. 5, pp. 2053–2058, 2012.

[Akr93] A. G. Akritas, "Sylvester's forgotten form of the resultant," *Fibonacci Quarterly*, vol. 31, pp. 325–332, Feb. 1993.

[AM05] K. M. Abadir and J. R. Magnus, *Matrix algebra*. Cambridge University Press, 2005, vol. 1.

[ASG99] Y. I. Abramovich, N. K. Spencer, and A. Y. Gorokhov, "Resolving manifold ambiguities in direction-of-arrival estimation for nonuniform linear antenna arrays," *IEEE Transactions on Signal Processing*, vol. 47, no. 10, pp. 2629–2643, Oct. 1999.

[Bar83] A. Barabell, "Improving the resolution performance of eigenstructure-based direction-finding algorithms," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 8. IEEE, 1983, pp. 336–339.

[BDD+00] Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. van der Vorst, *Templates for the solution of algebraic eigenvalue problems: a practical guide*. SIAM, 2000, vol. 11.

[BM86] Y. Bresler and A. Macovski, "Exact maximum likelihood parameter estimation of superimposed exponential signals in noise," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 34, no. 5, pp. 1081–1089, 1986.

[Box54] G. E. Box, "Some theorems on quadratic forms applied in the study of analysis of variance problems, I. Effect of inequality of variance in the one-way classification," *The annals of mathematical statistics*, vol. 25, no. 2, pp. 290–302, 1954.

[BPB03] M. Bühren, M. Pesavento, and J. F. Böhme, "A new approach to array interpolation by generation of artificial shift invariances: interpolated ESPRIT," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 5, 2003, pp. V–205.

[Bri81] D. R. Brillinger, *Time series: data analysis and theory.* SIAM, 1981, vol. 36.

[BV04] S. P. Boyd and L. Vandenberghe, *Convex optimization.* Cambridge university press, 2004.

[BZP02] R. F. Brcich, A. M. Zoubir, and P. Pelin, "Detection of sources using bootstrap techniques," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 206–215, 2002.

[Cap69] J. Capon, "High-resolution frequency-wavenumber spectrum analysis," *Proceedings of the IEEE*, vol. 57, no. 8, pp. 1408–1418, 1969.

[CGTW95] R. M. Corless, P. M. Gianni, B. M. Trager, and S. M. Watt, "The singular value decomposition for polynomial systems," in *the 1995 international symposium on Symbolic and Algebraic Computation.* Montreal, Canada: ACM, July 1995, pp. 195–207.

[CMB04] D. Cabric, S. M. Mishra, and R. W. Brodersen, "Implementation issues in spectrum sensing for cognitive radios," in *Conference record of the thirty-eighth Asilomar conference on Signals, systems and computers*, vol. 1, 2004, pp. 772–776.

[CR12] I. CVX Research, "CVX: Matlab software for disciplined convex programming, version 2.0," http://cvxr.com/cvx, Aug. 2012.

[CSPC11] Y. Chi, L. L. Scharf, A. Pezeshki, and A. R. Calderbank, "Sensitivity to basis mismatch in compressed sensing," *IEEE Transactions on Signal Processing*, vol. 59, no. 5, pp. 2182–2195, 2011.

[CW02] J. K. Cullum and R. A. Willoughby, *Lanczos Algorithms for Large Symmetric Eigenvalue Computations: Volume 1, Theory.* SIAM, 2002, no. 41.

[CW08] E. J. Candès and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21–30, 2008.

[DeG74] M. H. DeGroot, "Reaching a consensus," *Journal of the American Statistical Association*, vol. 69, no. 345, pp. 118–121, 1974.

[DET06] D. L. Donoho, M. Elad, and V. N. Temlyakov, "Stable recovery of sparse overcomplete representations in the presence of noise," *IEEE Transactions on Information Theory*, vol. 52, no. 1, pp. 6–18, 2006.

[DH01] D. L. Donoho and X. Huo, "Uncertainty principles and ideal atomic decomposition," *IEEE Transactions on Information Theory*, vol. 47, no. 7, pp. 2845–2862, 2001.

[DT08] D. L. Donoho and Y. Tsaig, "Fast solution of-norm minimization problems when the solution may be sparse," *IEEE Transactions on Information Theory*, vol. 54, no. 11, pp. 4789–4812, 2008.

[Far84] R. Farebrother, "[algorithm AS 204] the distribution of a positive linear combination of χ^2 random variables," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 33, no. 3, pp. 332–339, 1984.

[Fri90] B. Friedlander, "Direction finding using an interpolated array," *IEEE Signal Process Letters*, vol. 5, pp. 2951–2954, 1990.

[FW98] B. Friedlander and A. J. Weiss, "On the second-order statistics of the eigenvectors of sample covariance matrices," *IEEE Transactions on Signal Processing*, vol. 46, no. 11, pp. 3136–3139, 1998.

[GR97] I. F. Gorodnitsky and B. D. Rao, "Sparse signal reconstruction from limited data using FOCUSS: a re-weighted minimum norm algorithm," *IEEE Transactions on Signal Processing*, vol. 45, no. 3, pp. 600–616, Mar. 1997.

[Gra81] A. Graham, *Kronecker products and matrix calculus: with applications*, ser. Ellis Horwood series in mathematics and its applications. Horwood, 1981.

[GS05] A. Ghasemi and E. S. Sousa, "Collaborative spectrum sensing for opportunistic access in fading environments," in *IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks. DySPAN*, 2005, pp. 131–136.

[GVL13] G. H. Golub and C. F. Van Loan, *Matrix computations*. Johns Hopkins University Press, Baltimore, fourth edition, 2013, vol. 3.

[HM10] M. M. Hyder and K. Mahata, "Direction-of-arrival estimation using a mixed norm approximation," *IEEE Transactions on Signal Processing*, vol. 58, no. 9, pp. 4646–4655, 2010.

[HN96] B. Hochwald and A. Nehorai, "Identifiability in array processing models with vector-sensor applications," *IEEE Transactions on Signal Processing*, vol. 44, no. 1, pp. 83–95, 1996.

[HXC+99] Y. Hua, Y. Xiang, T. Chen, K. Abed-Meraim, and Y. Miao, "A new look at the power method for fast subspace tracking," *Digital Signal Processing*, vol. 9, no. 4, pp. 297–314, 1999.

[Joh90] C. R. Johnson, *Matrix theory and applications*. Proceedings of Symposia in Applied Mathematic, American Mathematical Society, 1990, vol. 40.

[Jon83] D. Jones, "Statistical analysis of empirical models fitted by optimization," *Biometrika*, vol. 70, no. 1, pp. 67–88, 1983.

[KFMH06] N. Karcanias, S. Fatouros, M. Mitrouli, and G. H. Halikias, "Approximate greatest common divisor of many polynomials, generalised resultants, and strength of approximation," *Elsevier Computers & Mathematics with Applications*, vol. 51, no. 12, pp. 1817–1830, June 2006.

[KHE15] H. Kim, A. M. Haimovich, and Y. C. Eldar, "Non-coherent direction of arrival estimation from magnitude-only measurements," *IEEE Signal Processing Letters*, vol. 22, no. 7, pp. 925–929, 2015.

[Kow09] M. Kowalski, "Sparse regression using mixed norms," *Applied and Computational Harmonic Analysis*, vol. 27, no. 3, pp. 303–324, 2009.

[Kru77] J. B. Kruskal, "Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics," *Linear algebra and its applications*, vol. 18, no. 2, pp. 95–138, 1977.

[KV96] H. Krim and M. Viberg, "Two decades of array signal processing research: the parametric approach," *IEEE Signal Processing Magazine*, vol. 13, no. 4, pp. 67–94, July 1996.

[LKM90] D. D. Lee, R. L. Kashyap, and R. N. Madan, "Robust decentralized direction-of-arrival estimation in contaminated noise," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 38, no. 3, pp. 496–505, Mar. 1990.

[LZ13] Z. Lu and A. M. Zoubir, "Flexible detection criterion for source enumeration in array processing," *IEEE Transactions on Signal Processing*, vol. 61, no. 6, pp. 1303–1314, 2013.

[MÇW05] D. Malioutov, M. Çetin, and A. S. Willsky, "A sparse signal reconstruction perspective for source localization with sensor arrays," *IEEE Transactions on Signal Processing*, vol. 53, no. 8, pp. 3010–3022, 2005.

[Mit12] H. B. Mitchell, *Data fusion: concepts and ideas.* Springer Science & Business Media, 2012.

[Mui09] R. J. Muirhead, *Aspects of multivariate statistical theory.* John Wiley & Sons, 2009, vol. 197.

[OSFM07] R. Olfati-Saber, A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

[OSM04] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, 2004.

[Oui86] H. Ouibrahim, "A generalized approach to direction finding," Ph.D. dissertation, Syracuse University, 1986.

[PGH00] M. Pesavento, A. B. Gershman, and M. Haardt, "Unitary root-MUSIC with a real-valued eigendecomposition: A theoretical and experimental performance study," *IEEE Transactions on Signal Processing*, vol. 48, no. 5, pp. 1306–1314, 2000.

[PGL02] M. Pesavento, A. B. Gershman, and Z.-Q. Luo, "Robust array interpolation using second-order cone programming," *IEEE Signal Processing Letters*, vol. 9, no. 1, pp. 8–11, 2002.

[PGW02] M. Pesavento, A. B. Gershman, and K. M. Wong, "Direction finding in partly calibrated sensor arrays composed of multiple subarrays," *IEEE Transactions on Signal Processing*, vol. 50, no. 9, pp. 2103–2115, Sep. 2002.

[Pis73] V. F. Pisarenko, "The retrieval of harmonics from a covariance function," *Geophysical Journal International*, vol. 33, no. 3, pp. 347–366, 1973.

[PP11] P. Parvazi and M. Pesavento, "A new direction-of-arrival estimation and calibration method for arrays composed of multiple identical subarrays," in *IEEE International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, San Francisco, CA, USA, June 2011, pp. 171–175.

[PPG11] P. Parvazi, M. Pesavento, and A. B. Gershman, "Direction-of-arrival estimation and array calibration for partly-calibrated arrays," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Prague, Czech Republic, May 2011, pp. 2552–2555.

[PPG12] ——, "Rooting-based harmonic retrieval using multiple shift-invariances: The complete and the incomplete sample cases," *IEEE Trans. Signal Processing*, vol. 60, no. 4, pp. 1556–1570, Apr. 2012.

[PS12] F. Penna and S. Stanczak, "Decentralized largest eigenvalue test for multi-sensor signal detection," in *IEEE Global Communications Conference*, Dec. 2012, pp. 3893–3898.

[RF04] D. W. Rieken and D. R. Fuhrmann, "Generalizing MUSIC and MVDR for multiple noncoherent arrays," *IEEE Transactions on Signal Processing*, vol. 52, no. 9, pp. 2396–2406, Sep. 2004.

[RH89a] B. D. Rao and K. Hari, "Performance analysis of ESPRIT and TAM in determining the direction of arrival of plane waves in noise," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 37, no. 12, pp. 1990–1995, 1989.

[RH89b] ——, "Performance analysis of root-MUSIC," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 37, no. 12, pp. 1939–1949, 1989.

[RK89] R. Roy and T. Kailath, "ESPRIT-estimation of signal parameters via rotational invariance techniques," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 37, no. 7, pp. 984–995, Nov. 1989.

[RP49] H. Robbins and E. Pitman, "Application of the method of mixtures to quadratic forms in normal variates," *The annals of mathematical statistics*, pp. 552–560, 1949.

[SA89] P. Stoica and N. Arye, "MUSIC, maximum likelihood, and Cramer-Rao bound," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 37, no. 5, pp. 720–741, May 1989.

[SBL11] P. Stoica, P. Babu, and J. Li, "SPICE: A sparse covariance-based estimation method for array processing," *IEEE Transactions on Signal Processing*, vol. 59, no. 2, pp. 629–638, 2011.

[Sch86] R. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Transactions on Antennas and Propagation*, vol. 34, no. 3, pp. 276–280, Mar. 1986.

[SCN12] H. Salarian, K.-W. Chin, and F. Naghdy, "Coordination in wireless sensor–actuator networks: a survey," *Journal of Parallel and Distributed Computing*, vol. 72, no. 7, pp. 856–867, 2012.

[SG04] C. M. S. See and A. B. Gershman, "Direction-of-arrival estimation in partly calibrated subarray-based sensor arrays," *IEEE Transactions on Signal Processing*, vol. 52, no. 2, pp. 329–338, 2004.

[SLG01] P. Stoica, E. G. Larsson, and A. B. Gershman, "The stochastic CRB for array processing: a textbook derivation," *IEEE Signal Processing Letters*, vol. 8, no. 5, pp. 148–150, 2001.

[SN90] P. Stoica and A. Nehorai, "Performance study of conditional and unconditional direction-of-arrival estimation," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 38, no. 10, pp. 1783–1795, 1990.

[SNS95] P. Stoica, A. Nehorai, and T. Söderström, "Decentralized array processing using the MODE algorithm," *Circuits, Systems and Signal Processing*, vol. 14, no. 1, pp. 17–38, 1995.

[SP14] W. Suleiman and P. Parvazi, "Search-free decentralized direction-of-arrival estimation using common roots for non-coherent partly calibrated arrays," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014, pp. 2292–2296.

[SPK08] A. Scaglione, R. Pagliari, and H. Krim, "The decentralized estimation of the sample covariance," in *Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, CA, USA, Oct. 2008, pp. 1722–1726.

[SPP14] C. Steffens, P. Parvazi, and M. Pesavento, "Direction finding and array calibration based on sparse reconstruction in partly calibrated arrays," in *IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM)*, June 2014, pp. 21–24.

[SPP16] C. Steffens, M. Pesavento, and M. E. Pfetsch, "A compact formulation for the $\ell_{2,1}$ mixed-norm minimization problem," *arXiv:1606.07231*, 2016.

[SPPZ14] W. Suleiman, P. Parvazi, M. Pesavento, and A. M. Zoubir, "Decentralized direction finding using Lanczos method," in *IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM)*, June 2014, pp. 9–12.

[SPPZ17] ——, "Non-coherent direction-of-arrival estimation using partly calibrated arrays," *arXiv:1704.06000 [stat.AP]*, Apr. 2017.

[SPZ13] W. Suleiman, M. Pesavento, and A. M. Zoubir, "Decentralized direction finding using partly calibrated arrays," in *European signal processing conference (EUSIPCO)*, Sep. 2013, pp. 1–5.

[SPZ15a] ——, "Decentralized cooperative DOA tracking using non-hermitian generalized eigendecomposition," in *European signal processing conference (EUSIPCO)*, June 2015.

[SPZ15b] ——, "Performance analysis of direction-of-arrival estimation using the decentralized root-MUSIC," in *IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, Dec. 2015.

[SPZ16a] ——, "Decentralized cooperative detection based on averaging consensus," in *2016 IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM)*, July 2016, pp. 1–5.

[SPZ16b] ——, "Performance analysis of eigenvalue based distributed spectrum sensing," in *2016 International Symposium on Wireless Communication Systems (ISWCS)*, Sep. 2016, pp. 481–485.

[SPZ16c] ——, "Performance analysis of the decentralized eigendecomposition and ESPRIT algorithm," *IEEE Transactions on Signal Processing*, vol. 64, no. 9, pp. 2375–2386, May 2016.

[SS77] H. Solomon and M. A. Stephens, "Distribution of a sum of weighted chi-square variables," *Journal of the American Statistical Association*, vol. 72, no. 360a, pp. 881–885, 1977.

[SS90] P. Stoica and K. C. Sharman, "Maximum likelihood methods for direction-of-arrival estimation," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 38, no. 7, pp. 1132–1143, July 1990.

[SS92] T. Söderström and P. Stoica, "Statistical analysis of decentralized MUSIC," *Springer Circuits, Systems and Signal Processing*, vol. 11, no. 4, pp. 443–454, Dec. 1992.

[SS07] A. Stegeman and N. D. Sidiropoulos, "On Kruskal's uniqueness condition for the Candecomp/Parafac decomposition," *Linear Algebra and its applications*, vol. 420, no. 2, pp. 540–552, 2007.

[SS10] P. J. Schreier and L. L. Scharf, *Statistical signal processing of complex-valued data: the theory of improper and noncircular signals*. Cambridge University Press, 2010.

[Ste76] G. Stewart, "Simultaneous iteration for computing invariant subspaces of non-hermitian matrices," *Numerische Mathematik*, vol. 25, no. 2, pp. 123–136, 1976.

[Sto05] I. Stojmenovic, *Handbook of sensor networks: algorithms and architectures*. John Wiley & Sons, 2005, vol. 49.

[SVPZ16] W. Suleiman, A. A. Vaheed, M. Pesavento, and A. M. Zoubir, "Decentralized direction-of-arrival estimation for arbitrary array geometries," in *2016 24th European Signal Processing Conference (EUSIPCO)*, Aug. 2016, pp. 1921–1925.

[SW99] J. Sheinvald and M. Wax, "Direction finding with fewer receivers via time-varying preprocessing," *IEEE transactions on signal processing*, vol. 47, no. 1, pp. 2–9, 1999.

[SWW97] J. Sheinvald, M. Wax, and A. J. Weiss, "On the achievable localization accuracy of multiple sources at high snr," *IEEE Transactions on Signal Processing*, vol. 45, no. 7, pp. 1795–1799, July 1997.

[Tib96] R. Tibshirani, "Regression shrinkage and selection via the LASSO," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

[TM03] M. Tubaishat and S. Madria, "Sensor networks: an overview," *IEEE potentials*, vol. 22, no. 2, pp. 20–23, 2003.

[TNKG10] A. Taherpour, M. Nasiri-Kenari, and S. Gazor, "Multiple antenna spectrum sensing in cognitive radios," *IEEE Transactions on Wireless Communications*, vol. 9, no. 2, pp. 814–823, 2010.

[Tro06] J. A. Tropp, "Algorithms for simultaneous sparse approximation. Part II: Convex relaxation," *Signal Processing*, vol. 86, no. 3, pp. 589–602, 2006.

[TS05] R. Tandra and A. Sahai, "Fundamental limits on detection in low SNR under noise uncertainty," in *International Conference on Wireless Networks, Communications and Mobile Computing*, vol. 1, 2005, pp. 464–469.

[VOK91] M. Viberg, B. Ottersten, and T. Kailath, "Detection and estimation in sensor arrays using weighted subspace fitting," *IEEE Transactions on Signal Processing*, vol. 39, no. 11, pp. 2436–2449, Nov. 1991.

[VS78] A. I. G. Vardulakis and P. N. R. Stoyle, "Generalized resultant theorem," *IMA Journal of Applied Mathematics*, vol. 22, no. 3, pp. 331–335, Feb. 1978.

[VT02] H. L. Van Trees, *Detection, Estimation, and Modulation Theory - Part IV: Optimum Array Processing*. Wiley-Interscience, 2002.

[WG91] A. J. Weiss and M. Gavish, "The interpolated ESPRIT algorithm for direction finding," in *17th Convention of Electrical and Electronics Engineers in Israel*, 1991, pp. 361–364.

[WJ90] D. B. Williams and D. H. Johnson, "Using the sphericity test for source detection with narrow-band passive arrays," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 38, no. 11, pp. 2008–2014, 1990.

[WK85a] M. Wax and T. Kailath, "Decentralized processing in sensor arrays," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 33, no. 5, pp. 1123–1129, Oct. 1985.

[WK85b] ——, "Detection of signals by information theoretic criteria," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 33, no. 2, pp. 387–392, 1985.

[Woo89] A. T. Wood, "An F approximation to the distribution of a linear combination of Chi-squared variables," *Communications in Statistics-Simulation and Computation*, vol. 18, no. 4, pp. 1439–1456, 1989.

[WW11] M. Wiesinger-Widi, "Sylvester matrix and GCD for several univariate polynomials," RISC Report Series University of Linz, Austria, Tech. Rep., 2011.

[WZ89] M. Wax and I. Ziskind, "On unique localization of multiple sources by passive sensor arrays," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 37, no. 7, pp. 996–1000, 1989.

[XB04] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems & Control Letters*, vol. 53, no. 1, pp. 65–78, 2004.

[XBK07] L. Xiao, S. Boyd, and S.-J. Kim, "Distributed average consensus with least-mean-square deviation," *Journal of Parallel and Distributed Computing*, vol. 67, no. 1, pp. 33–46, 2007.

[XSRK94] G. Xu, S. Silverstein, R. Roy, and T. Kailath, "Beamspace ESPRIT," *IEEE Transactions on Signal Processing*, vol. 42, no. 2, pp. 349–356, 1994.

[YC11] J. Yin and T. Chen, "Direction-of-arrival estimation using a sparse representation of array covariance vectors," *IEEE Transactions on Signal Processing*, vol. 59, no. 9, pp. 4489–4493, 2011.

[YL06] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 68, no. 1, pp. 49–67, 2006.

Lebenslauf

Name:	Wassim Suleiman
Geburtsdatum:	05.02.1981
Geburtsort:	Syrien
Familienstand:	Verheiratet, ein Kind

Ausbildung

08.2012–06.2017	Promotion: Kooperative Richtungsschätzung (Direction of Arrival) in verteilten Sensornetzwerken, Fachgebiet Signalverarbeitung und Fachgebiet Nachrichtentechnische Systeme, TU Darmstadt, Deutschland.
10.2010–06.2012	Master Studium: Information and Communication Engineering (iCE), Fachbereich Elektrotechnik, TU Darmstadt, Masterarbeit: Super-Auflösung für Durchwand-Radar-Bildgebung basierend auf sparsen Signal-Schätzverfahren (Super-Resolution for Through-the-Wall Radar Imaging Based on Compressive Sensing).
09.1998–09.2004	Diplomingenieur Communications: Higher Institute for Applied Sciences and Technology (HIAST), Damaskus, Syrien, Diplomarbeit: Entwicklung und Implementierung eines Network Intrusion Detection System (NIDS).
09.1986–09.1998	Schulbildung: in Hama, Syrien.

Berufserfahrung

08.2012–laufend	Wissenschaftlicher Mitarbeiter, Fachgebiet Signalverarbeitung und Fachgebiet Nachrichtentechnische Systeme, TU Darmstadt, Deutschland.
10.2004–09.2010	Forschungsassistent, Fachgebiet Radar-Signalverarbeitung, HIAST, Syrien.
11.2007–08.2010	Teilzeit-Programmierer, Ebridge (http://www.ebridge-sy.com), Syrien.
07.2005–10.2007	Teilzeit-Programmierer, eData (http://www.edatame.com), Syrien.

Erklärung laut §9 der Promotionsordnung

Ich versichere hiermit, dass ich die vorliegende Dissertation allein und nur unter Verwendung der angegebenen Literatur verfasst habe. Die Arbeit hat bisher noch nicht zu Prüfungszwecken gedient.

Darmstadt, 01. Mai 2017,