KU LEUVEN

ARENBERG DOCTORAL SCHOOL
Faculty of Engineering Science
Department of Electrical Engineering

Distributed Signal Processing Algorithms for Multi-Task Wireless Acoustic Sensor Networks

Amin Hassani

Supervisor:
Prof. dr. ir. M. Moonen
Co-supervisor:
Prof. dr. ir. A. Bertrand

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor of Engineering Science (PhD): Electrical Engineering

October 2017

Distributed Signal Processing Algorithms for Multi-Task Wireless Acoustic Sensor Networks

Amin HASSANI

Examination committee:
Em. prof. dr. ir. H. Hens, chair
Prof. dr. ir. M. Moonen, supervisor
Prof. dr. ir. A. Bertrand, co-supervisor
Prof. dr. ir. H. Van hamme
Prof. dr. ir. J. Suykens
Prof. dr. ir. T. van Waterschoot
Prof. dr. ir. P. A. Naylor
(Imperial College London)

Dissertation presented in partial fulfillment of the requirements for the degree of Doctor in Engineering Science

October 2017

© 2017 KU Leuven – Faculty of Engineering Science
Uitgegeven in eigen beheer, Amin Hassani, Kasteelpark Arenberg 10, bus 2446, B-3001 Leuven (Belgium)

Alle rechten voorbehouden. Niets uit deze uitgave mag worden vermenigvuldigd en/of openbaar gemaakt worden door middel van druk, fotokopie, microfilm, elektronisch of op welke andere wijze ook zonder voorafgaande schriftelijke toestemming van de uitgever.

All rights reserved. No part of the publication may be reproduced in any form by print, photoprint, microfilm, electronic or any other means without written permission from the publisher.

Abstract

Recent technological advances in analogue and digital electronics as well as in hardware miniaturization have taken wireless sensing devices to another level by introducing low-power communication protocols, improved digital signal processing capabilities and compact sensors. When these devices perform a certain pre-defined signal processing (SP) task such as the estimation or detection of phenomena of interest, a cooperative scheme through wireless connections can significantly enhance the overall performance, especially in adverse conditions. The resulting network consisting of such connected devices (or nodes) is referred to as a wireless sensor network (WSN). In short, the advantage of WSNs compared to conventional fixed sensor arrays is related to the fact that they provide access to more physically-distributed sensors, yielding a more informative spatial sampling of the phenomena of interest and therefore leading to a superior performance of the SP task. In this thesis, we focus specifically on *estimation* tasks for signals, subspaces or parameters within such WSNs.

In acoustical applications (e.g., speech enhancement) a variant of WSNs, called wireless acoustic sensor networks (WASNs) can be employed in which the sensing unit at each node consists of a single microphone or a microphone array. The nodes of such a WASN can then cooperate to perform a multi-channel acoustic SP task, such as multi-channel noise reduction, echo cancellation, dereverberation, active noise control (ANC), or source localization. In general, WASNs deal with the acquisition and estimation of audio content for which data has to be sampled, processed and transmitted at a higher rate than in traditional low-power and low-rate WSNs. Hence, nodes of a WASN indeed demand greater processing power, communication resources and therefore they exhibit higher power consumption as compared to most other types of WSNs. Therefore, it is critical to design highly efficient SP algorithms under which nodes of a WASN can cooperate and enhance their signal, subspace, or parameter estimation performance, subject to constraints in bandwidth, computational complexity, or energy consumption.

WASNs typically assume a setting in which all the nodes are of the same type and cooperate to solve a single network-wide SP task. Recently, however, WASNs have started to emerge in which the nodes cooperate with each other to solve *multiple node-specific* SP tasks, i.e., one (different) task for each node. These types of WASNs are referred to as multi-task WASNs in which each node is interested in estimating a different set of signals or parameters as observed by its own reference sensors, leading to different node-specific SP tasks which are somehow related since the observed signals are often highly correlated across the sensors of different nodes.

This thesis aims at developing novel *distributed* SP algorithms for signal, parameter and subspace estimation in such multi-task WASNs. Distributed processing provides an attractive alternative to centralized processing, since for the latter case all the uncompressed sensor signals of the entire WASN have to be aggregated and processed in one place (e.g., in a fusion center), which demands a large communication bandwidth and therefore consumes a great deal of energy. In general, the distributed SP algorithms developed in this thesis aim at letting each node of a multi-task WASN obtain the centralized solution of its corresponding node-specific SP task, although nodes cooperate with a significantly reduced-bandwidth signal transmission relying on compressive *filter-and-sum* operations.

The first part of the thesis focuses on designing distributed algorithms for multi-task WASNs where the node-specific SP tasks all rely on the *same* basic SP technique, which can include signal enhancement, beamforming, spectrum estimation, subspace estimation, or direction of arrival (DOA) estimation. Such multi-task WASNs are classified as *homogeneous* multi-task WASNs, since all these nodes locally apply the same basic SP technique to their sensor signal observations (as part of their pre-defined routine operations). For instance, a homogeneous multi-task WASN can be established inside an auditorium where multiple hearing aids wish to cooperate with each other through wireless links. In this scenario, the hearing aids locally apply the same basic noise reduction technique to solve their node-specific noise reduction tasks.

The second part of the thesis develops distributed algorithms for multi-task WASNs where the node-specific SP tasks rely on *different* basic SP techniques. Such multi-task WASNs are classified as *heterogeneous* multi-task WASNs. The resulting distributed SP algorithms of this part attempt to provide a framework under which daily-life heterogeneous devices running particular SP techniques also become capable to readily exchange signals and enhance their estimation performance, without relying on a rigid set of pre-defined routine operations and even without having any prior knowledge about the SP techniques other nodes use to solve their SP tasks. For instance, a heterogeneous multi-task WASN in this category can be established in an environment where several multimedia

devices such as smartphones, laptops, tablets, ANC headphones, or hearing aids cooperate and share fused microphone signals to enhance their own estimation performance using different node-specific SP techniques such as multi-channel Wiener filtering, minimum variance beamforming, or subspace-based DOA estimation.

The third part of the thesis, provides a real-time experimental validation for the developed distributed algorithm in a fully adaptive and realistic speech enhancement scenario. This scenario is created using an acoustic sensor network with three collaborating microphone arrays. The output signals of the distributed algorithm are assessed by means of instrumental measures for both speech intelligibility and speech quality.

Finally, the last chapter provides the conclusions, summarizes the contributions of this thesis and further discusses possible future research directions.

Beknopte samenvatting

De huidige technologische vooruitgang in analoge en digitale elektronica, gecombineerd met de miniaturisatie van hardware, hebben geleid tot een verbetering in draadloze sensorapparatuur. Dit mede door de introductie van laag-vermogen communicatieprotocollen, verbeterde digitale signaalverwerkingsmogelijkheden en compactere sensoren. Wanneer deze apparaten een bepaalde signaalverwerkingstaak ('*signal processing*' of SP) uitvoeren, zoals het schatten of detecteren van bepaalde fysische fenomenen, kan een coöperatief schema waarin de apparaten informatie uitwisselen over draadloze verbindingen de algemene performantie aanzienlijk verbeteren. Deze verbetering gebeurt voornamelijk in ongunstige omstandigheden. Het resulterende netwerk, bestaande uit verbonden sensorapparaten (of knooppunten), wordt een draadloos sensornetwerk ('*wireless sensor network*' of WSN) genoemd. Het voordeel van WSN's in vergelijking met conventionele vaste sensorroosters, is het feit dat ze toegang bieden tot meer verspreide sensoren, resulterend in een betere ruimtelijke bemonstering van de fenomenen. In dit proefschrift richten we ons specifiek op taken gerelateerd aan het schatten van signalen, deelruimten of parameters aan de hand van dergelijke WSN's.

WSN's die gebruikt worden in akoestische toepassingen zoals spaakverbetering worden typisch draadloze akoestische sensornetwerken ('*wireless acoustic sensor networks*' of WASNs) genoemd. Ieder knooppunt in een WASN bestaat uit een microfoon of een lokaal microfoonrooster. De knooppunten van een WASN kunnen dan samenwerken om een meerkanaals akoestische SP taak uit te voeren, zoals meerkanaals ruisonderdrukking, echocancellatie, dereverberatie, actieve ruisonderdrukking ('*active noise control*' of ANC) of bronlokalisatie. Over het algemeen werken WASNs bij de acquisitie en het schatten van spraaksignalen op een hogere snelheid dan in traditionele laag-vermogen en lage-snelheid WSNs. Hierdoor vereisen knooppunten van een WASN meer rekenkracht en communicatiemiddelen, waardoor ze een hoger energieverbruik vertonen ten opzichte van de meeste andere typen WSNs. Daarom is het van essentieel belang om efficiënte SP algoritmen te ontwerpen waarin de

knooppunten van de WASN samenwerken om hun signaal-, deelruimte- of parameterschattingsprestaties verbeteren, rekening houdend met beperkingen in bandbreedte, rekencomplexiteit en/of energieverbruik.

WASNs veronderstellen meestal dat alle knooppunten van hetzelfde type zijn en samenwerken om één enkele SP taak op te lossen over het gehele netwerk. Recentelijk zijn er echter WASNs opgedoken waarin de knooppunten samenwerken om meerdere knooppuntspecifieke SP taken op te lossen, i.e. één (verschillende) taak voor ieder knooppunt. Dit type WASN wordt een '*multi-task*' WASN genoemd, waarin ieder knooppunt geïnteresseerd is in het schatten van verschillende knooppuntsspecifieke signalen of parameters zoals waargenomen door zijn eigen referentiesensoren. Deze verschillen in interesses leiden tot verschillende knooppuntspecifieke SP taken die desalniettemin gerelateerd zijn, aangezien de waargenomen signalen van verschillende knooppunten vaak zeer sterk gecorreleerd zijn.

Dit proefschrift streeft naar het ontwikkelen van nieuwe gedistribueerde SP algoritmen voor signaal-, parameter- en deelruimte-schatting in dergelijke multi-task WASNs. Gedistribueerde SP biedt een aantrekkelijk alternatief voor gecentraliseerde SP, aangezien deze laatste vereist dat alle originele sensorsignalen van de gehele WASN op één plaats geaggregeerd en verwerkt moeten worden (e.g. in een fusiecentrum), wat een grote communicatiebandbreedte vergt en gepaard gaat met een hoog energieverbruik. In het algemeen zijn de voorgestelde gedistribueerde SP algoritmen bedoeld om, voor elk knooppunt van een multi-task WASN, de gecentraliseerde oplossing van zijn knooppuntspecifieke SP taak te bekomen en dit te verwezenlijken tegen een significant verminderde bandbreedte voor signaaloverdracht. Deze reductie wordt gerealiseerd aan de hand van comprimerende filter-en-som operaties.

Het eerste deel van het proefschrift richt zich op het ontwerpen van gedistribueerde algoritmen voor multi-task WASNs waarbij alle knooppuntspecifieke SP taken gebaseerd zijn op dezelfde fundamentele SP techniek, dewelke zowel signaalversterking als bundelsturing, spectrumschatting, deelruimte-schatting of '*direction of arrival*' (DOA)-schatting kan omvatten. Dergelijke multi-task WASNs worden geclassificeerd als homogene multi-task WASNs aangezien alle knooppunten lokaal dezelfde fundamentele SP techniek toepassen op hun signaalwaarnemingen (als deel van hun vooraf gedefinieerde routine-operaties). Zo kan een homogene multi-task WASN worden opgezet in een auditorium waar meerdere gehoorapparaten met elkaar samenwerken door middel van draadloze verbindingen. In dit scenario passen alle gehoorapparaten lokaal dezelfde geluidsreductietechniek toe om hun knooppuntspecifieke taken te vervullen.

Het tweede deel van het proefschrift ontwikkelt gedistribueerde algoritmen voor multi-task WASNs, waarbij de knooppuntspecifieke SP taken gebaseerd zijn

op verschillende fundamentele SP technieken. Dergelijke multi-task WASNs worden geclassificeerd als heterogene multi-task WASNs. De resulterende gedistribueerde SP algoritmen uit dit deel maken het mogelijk dat alledaagse heterogene apparaten, die specifieke SP taken uitvoeren, in staat zullen zijn om signalen eenvoudig uit te wisselen. Zo verbeteren ze hun eigen schattingsprestaties zonder uit te gaan van een starre reeks vooraf gedefinieerde routine-operaties en zelfs zonder voorafgaande kennis over de SP technieken die andere knooppunten gebruiken om hun SP taken te vervullen. Een heterogene multi-task WASN in deze categorie kan bijvoorbeeld gebruikt worden in een omgeving waar meerdere multimedia-apparaten zoals smartphones, laptops, tablets, ANC-hoofdtelefoons of gehoorapparaten samenwerken en gefuseerde microfoonsignalen delen om hun eigen schattingsprestatie te verbeteren met behulp van eigen knooppuntspecifieke SP technieken, zoals meerkanaals-Wiener filtering, minimum-variantie bundelsturing of deelruimtegebaseerde DOA-schatting.

Het derde deel van het proefschrift beschrijft een experimentele real-time validatie van het ontwikkelde gedistribueerde algoritme en dit in een volledig adaptief en realistisch spraakverbeteringsscenario. Dit scenario is gemaakt met behulp van een akoestisch sensor netwerk met drie samenwerkende microfoonroosters. De kwaliteit van de uitgangssignalen van het gedistribueerde algoritme wordt beoordeeld door middel van spraakverstaanbaarheids- en spraakkwaliteitsmaten.

Tenslotte geeft het laatste hoofdstuk de conclusies, een samenvatting van de bijdragen van dit proefschrift, alsook een bespreking van eventuele toekomstige onderzoeksmogelijkheden.

Glossary

Acronyms and Abbreviations

ANC	active noise canceling
ATF	acoustic transfer function
CI	cochlear implant
DACGEE	distributed adaptive covariance-matrix generalized eigenvector estimation
DANSE	distributed adaptive node-specific signal estimation
DB-MWF	distributed multi-channel Wiener filter
DFT	discrete Fourier transform
DOA	direction of arrival
DSP	digital signal processor/processing
EEG	electroencephalography
ESPRIT	estimation of signal parameters via the rotational invariance technique
EVD	eigenvalue decomposition
FC	fusion center
GEVC	generalized eigenvector
GEVD	generalized eigenvalue decomposition
GEVL	generalized eigenvalue

HA	hearing aid
IoT	internet of things
LC-DANSE	linearly constrained distributed adaptive node-specific signal estimation
LCMV	linearly constrained minimum variance
LMMSE	linear minimum mean squared error
LMS	least mean squares
MC	Monte Carlo
MDMT	multiple devices multiple tasks
MDST	multiple devices single task
MSE	mean square error
MUSIC	multiple signal classification
MVDR	minimum variance distortionless response
MWF	multi-channel Wiener filter
NSPE	node-specific parameter estimation
NSSE	node-specific signal estimation
RIR	room impulse response
RLS	recursive least squares
SDR	signal to distortion ratio
SINR	signal to interference plus noise ratio
SNR	signal-to-noise ratio
SP	signal processing
STFT	short-time Fourier transform
T-DANSE	distributed adaptive node-specific signal estimation in a tree topology
TLS	total least-squares
ULA	uniform linear array

VAD	voice activity detector
WASN	wireless acoustic sensor network
WBAN	wireless body area network
WMSN	wireless multimedia sensor network
WSN	wireless sensor network
a.k.a	also known as
cfr.	*conferatur*, compare with, see also
dB	decibel
e.g.	*exempli gratia*, for example
Hz	hertz
i.e.	*id est*, that is
i.i.d.	independent and identically distributed
s.t.	subject to
w.l.o.g.	without loss of generality

Mathematical Notation

a	scalar
$\mathbf{a}$	vector
$\mathbf{A}$	matrix
$\mathbf{I}$	identity matrix
$\mathbf{0}$	null matrix or vector
range$(\cdot)$	range or column space of a matrix
Tr$\{\cdot\}$	trace of a matrix
$\{\cdot\}^*$	scalar complex conjugate
$\{\cdot\}^H$	vector or matrix conjugate transpose
$\{\cdot\}^T$	vector or matrix transpose
$\{\cdot\}^{-1}$	matrix inverse

$E\{\cdot\}$	mathematical expectation
$\text{rank}(\mathbf{A})$	rank of a matrix $\mathbf{A}$
$\angle(\cdot)$	phase of a complex number
$\approx$	approximately equal to
$\exists$	there exists
$\forall$	for all
$\Leftrightarrow$	material equivalence
$\lim_{x \to a} f(x)$	limit of function f as x approaches a
$\lvert\cdot\rvert$	cardinality of a set or absolute value
$\lvert\lvert\cdot\rvert\rvert$	norm
$\lvert\lvert\cdot\rvert\rvert_F$	Frobenius norm
$\Rightarrow$	material implication
$\text{Blkdiag}\{\cdot\}$	block-diagonal matrix with arguments on block-diagonal
$\text{diag}\{\cdot\}$	diagonal matrix with arguments on diagonal
$\triangleq$	is defined as
$\arg\min_x$	argument of the minimum
$\underset{x}{\text{minimize}}$	minimize over x
mod	modular operator
$\setminus$	set-theoretic complement
$\cup$	set-theoretic union
$\in$	is member of

Contents

Abstract i

Beknopte samenvatting v

Glossary ix

Contents xiii

1 Introduction 1

1.1 Wireless sensor networks (WSNs) 2

1.1.1 Definition and motivation 2

1.1.2 Wireless acoustic sensor networks (WASNs) and applications 4

1.2 Signal processing in WSNs 5

1.2.1 Centralized vs. distributed processing 6

1.2.2 Signal vs. parameter exchange in distributed processing 7

1.2.3 Top-down vs. bottom-up distributed processing 10

1.3 Multi-task WSNs 12

1.3.1 Node-specific vs. network-wide estimation problems . . 12

1.3.2 Homogeneous vs. heterogeneous multi-task WSNs . . . 12

1.4 Challenges of distributed algorithm design for WSNs 14

1.4.1 Communication bandwidth 14
1.4.2 Efficient use of exchange signals for integrated tasks . . 15
1.4.3 Robustness to adverse scenarios 15
1.4.4 Per-iteration computational complexity 15
1.4.5 Adaptivity and convergence speed 16
1.4.6 Blindness 16
1.4.7 Network topology 17
1.5 Overview of the basic SP techniques 18
1.5.1 Beamforming 18
1.5.2 Multi-channel Wiener filter (MWF) 19
1.5.3 Direction of Arrival (DOA) and signal subspace estimation 19
1.6 Overview of the thesis 20
1.6.1 General overview 20
1.6.2 Thesis technical assumptions 21
1.6.3 Thesis outline 22

Part I Distributed signal processing algorithms for homogeneous multi-task WSNs 27

2 Distributed algorithms for GEVD-based signal subspace estimation 29
2.1 Introduction 30
2.2 Data model and problem statement 31
2.3 Centralized GEVD-based signal subspace estimation 33
2.4 Distributed GEVD-based signal subspace estimation 35
2.4.1 The fully-connected DACGEE algorithm 35
2.4.2 Extracting the signal subspace 36
2.5 Simulation results 41

2.6 Conclusion 42

3 GEVD-based DANSE for robust node-specific signal estimation 43

3.1 Introduction 45

3.2 Data model 47

3.3 Network-wide GEVD-based MWF 48

3.3.1 Network-wide MWF 48

3.3.2 Network-wide GEVD-based MWF 49

3.4 GEVD-based DANSE 51

3.4.1 Simplification $L = R$ 52

3.4.2 Algorithm assumptions 53

3.4.3 Algorithm description 53

3.4.4 Parameterization of the solution space 55

3.4.5 Communication cost and computational complexity . . 59

3.4.6 Convergence analysis 59

3.5 Simulation results 69

3.5.1 Batch-mode simulations 70

3.5.2 Finite-window simulations 76

3.6 Conclusion and future work 80

4 Distributed integrated noise reduction and node-specific DOA estimation 81

4.1 Introduction 83

4.2 Data model, problem statement and preview 85

4.3 Subspace-based DOA estimation 87

4.3.1 MUSIC 89

4.3.2 ESPRIT 90

4.4 Noise reduction preprocessing 91

4.4.1 Multi-channel Wiener filter 91
4.4.2 DANSE-based cooperative noise reduction 93
4.5 Cooperative integrated noise reduction and DOA estimation . . 95
4.5.1 Cooperative DOA estimation 95
4.5.2 Theoretical motivation 97
4.5.3 Shortcutting the noise reduction 100
4.6 Simulation Results 101
4.6.1 Evaluation aspects 101
4.6.2 Scenario 1 103
4.6.3 Scenario 2 106
4.6.4 Scenario 3: multi-source case 108
4.7 Conclusion 110

Part II Distributed signal processing algorithms for heterogeneous multi-task WSNs 111

5 Multi-task WSN for signal enhancement, MVDR beamforming and DOA estimation: single source case 113

5.1 Introduction 115
5.2 Data model and problem statement 117
5.3 Centralized estimation 117
5.3.1 MWF 118
5.3.2 MVDR 119
5.3.3 DOA estimation 120
5.4 Distributed MDMT-based algorithm 120
5.5 Numerical simulations 123
5.6 Conclusion 124

6 Multi-task WSN for signal enhancement, LCMV beamforming and DOA estimation: multi-source case with node-specific constraints 127

6.1 Introduction . . . 129

6.2 Overview, data model and problem statement . . . 131

6.2.1 Illustrative example . . . 131

6.2.2 Notation overview . . . 132

6.2.3 Data model and problem statement . . . 132

6.3 Centralized estimation algorithms . . . 135

6.3.1 Network-wide MWF . . . 135

6.3.2 Network-wide LCMV beamforming . . . 137

6.3.3 Network-wide DOA estimation . . . 139

6.4 Distributed MDMT algorithm . . . 140

6.4.1 Prelude: the homogeneous case . . . 142

6.4.2 The heterogeneous case . . . 147

6.5 Convergence and optimality . . . 148

6.5.1 Parameterization of the solution space . . . 149

6.5.2 Proof of convergence and optimality . . . 151

6.6 Simulation results . . . 153

6.7 Conclusions . . . 155

Part III Real-time experimental validation 159

7 Distributed GEVD-based Speech Enhancement with Collaborating Microphone Arrays: Real-Time Implementation and Instrumental Evaluation 161

7.1 Introduction . . . 161

7.2 Algorithms under evaluation . . . 163

7.3 Real-time signal processing platform . . . 164

7.4 Test setup description 166
7.4.1 Audio equipment 166
7.4.2 Acoustic test scenario 167
7.5 Instrumental measures 170
7.5.1 SI-weighted SNR (SNR_{SI}) 170
7.5.2 SI-weighted spectral distortion (SD_{SI}) 170
7.5.3 Short-time objective intelligibility (STOI) 171
7.5.4 Perceptual evaluation of speech quality (PESQ) 171
7.6 Graphical user interface (GUI) 172
7.7 Results and discussion 173
7.7.1 General specifications 173
7.7.2 Single target speaker scenario 176
7.7.3 Two target speakers scenario 180

8 Conclusion **185**

8.1 Summary and conclusions 185
8.2 Suggestions for future research 191

A Subspace projection for robust LCMV beamforming **195**

A.1 Introduction 197
A.2 Data model and problem statement 198
A.3 LCMV beamforming 199
A.4 Projection-based subspace estimation 200
A.5 Simulation results 202
A.5.1 Simulated scenario with narrowband source signals . . . 202
A.5.2 Multi-talker speech enhancement 203
A.6 Conclusion 205

B APPENDIX to Chapter 3 **207**

B.1 Proof of Theorem II 207

B.2 Algorithm fixes for special cases 212

C APPENDIX to Chapter 6 **215**

C.1 Distributed GEVD algorithm 215

C.2 Proof of Theorem I 216

Bibliography **221**

Curriculum Vitae **235**

Publication List **237**

Chapter 1

Introduction

Recent technological advances in analogue and digital electronics as well as in hardware miniaturization have taken wireless sensing devices to another level by introducing low-power communication protocols, improved digital signal processing capabilities and compact sensors. When these devices perform a certain pre-defined signal processing (SP) task such as the estimation or detection of phenomena of interest, a cooperative scheme through wireless connections can significantly enhance the overall performance, especially in adverse conditions. The resulting network consisting of such connected devices (or nodes) is referred to as a wireless sensor network (WSN). In acoustical applications, a variant of WSNs called wireless acoustic sensor networks (WASNs) is used in which the sensing unit at each node consists of a single microphone or a microphone array. For this particular type of WSNs, however, it is critical to design and apply highly efficient SP algorithms under which nodes can cooperate and enhance the output performance of their corresponding SP task, subject to constraints in bandwidth, computational complexity, or energy consumption. This thesis aims at developing novel efficient distributed SP algorithms for WASNs in which the nodes are tasked with estimation of a signal, subspace, or parameter of interest.

This introduction is organized as follows. In Section 1.1, WSNs and WASNs will be introduced in detail and the benefits that they typically offer in real-life scenarios will be discussed. In Section 1.2, different processing strategies will be explained and the advantages of having the entire computational burden of a WSN shared between the nodes inside the network (rather than in a central point) will be pointed out. Furthermore, the general problem statement of this thesis will be given in this section. In Sections 1.3 and 1.4, multi-task WSNs will first be introduced and then some prevailing challenges for designing

distributed SP algorithms for such networks will be reviewed. These sections will also provide the current state of the art for distributed signal and parameter estimation over WSNs. In Section 1.5, the multi-channel SP techniques applied in the later chapters of this thesis will briefly be explained. Finally, in Section 1.6, a chapter-by-chapter outline and the contributions of the thesis will be presented.

1.1 Wireless sensor networks (WSNs)

1.1.1 Definition and motivation

Sensor arrays have been successfully utilized in various practical SP applications for decades. Such arrays can be formed by a multitude of sensors such as microphones, image sensors, light sensors, or motion sensors. For instance, microphone arrays have been widely used to carry out audio and speech processing in portable devices (e.g., tablets, laptops, smartphones), wearable devices (e.g., smartwatches, hearing aids (HAs), Cochlear implants (CIs)) or mixed-reality devices (e.g., smartglasses) [2].

By effectively exploiting the spatial properties of the captured signals, multi-sensor (or multi-channel) SP techniques can deliver a significantly better performance when compared to single-sensor (or single-channel) SP techniques which are constrained by only temporal or spectral properties of the captured signals [3, 4]. For instance, multi-channel SP techniques enable detection and separation of the captured signals with respect to their originating direction, which is a major advantage in scenarios where non-stationary interfering or noise components are present. In these cases, the performance of multi-channel SP techniques typically improves with the number of sensors since more sensors provide more spatial information [3].

However, sensors that form a conventional array are typically placed at *fixed* locations and arranged in a certain *compact* geometry with a known position with respect to a reference sensor or point. Consequently, two inevitable constraints come along with these compact, stand-alone sensor arrays which could significantly affect the performance of their associated SP techniques. Firstly, due to size constraints, it may not be possible to include more sensors in the array (e.g., in HAs or CIs). Alternatively if sensors can be included, they will be forced to be placed so close together that a reasonable compromise between the number of added sensors and the amount of enhanced spatial information cannot be attained. This therefore results in a rather local sampling of the spatial field. Secondly, the target sources and the interfering sources may

both come from the same direction, or the target sources may be located far away from the sensor array. In these cases, it has been shown that including signals of other nearby sensors which are placed in strategically better locations (e.g., closer to the target sources) can significantly increase the performance, as the combination indeed provides more spatial information. For instance in audio and speech enhancement applications, binaural (two-sided) HAs deliver a better performance in terms of speech intelligibility, enhanced localization and improved quality of listening, when compared to monaural (single-sided) HAs [5–7]. Furthermore, adding an external microphone - which can be worn or placed close to the desired speaker - to a binaural HA noise reduction system can further improve the output signal to noise ratio (SNR) of each device simultaneously and even lead to a better preservation of the binaural noise cues [8, 9]. Further performance improvements that can be achieved by adding more of such external microphones to a binaural HA noise reduction system has also been investigated in [10].

These physically distributed sensor arrays could then be simply connected via wires, forming a so-called wired sensor network. The resulting sensor network, however, would not work when the sensor arrays are mobile. This therefore poses a limitation on the locations where the sensor arrays can be placed and often results in a rather high installation and maintenance cost [11], especially in applications where such a wired sensor network needs to be established in a large enclosure, e.g., in applications in [12–14].

Alternatively, a wireless communication can be utilized for data exchange between these physically distributed sensor arrays. This requires each device to be equipped with a wireless transmitter and receiver, and a power source (often battery). The network consisting of a multitude of such cooperating devices, referred to as *nodes*, is then called a **wireless sensor network (WSN)** [15–17]. WSNs can be organized in a hierarchical structure where multiple master nodes can be defined, which collect sensor observations from multiple slave nodes in their neighborhood. Although WSNs solve many of the practical issues posed by a wired sensor network, they inevitably come with their own challenges such as bandwidth and power constraints, which has to be well addressed when designing suitable algorithms. Section 1.4 provides a discussion about the design challenges of such algorithms.

WSNs can be classified as either *homogeneous* or *heterogeneous*. In homogeneous WSNs, nodes are often identical in terms of the processing abilities, embedded algorithms, communication range and power budget, while in heterogeneous WSNs different types of nodes are being present. Therefore, when it comes to the design of algorithms for heterogeneous WSNs, discrepancies between the nodes should be taken into account, and further exploited when possible.

The envisaged WSNs of this thesis contain several nodes where each one includes a sensor array of which the position and orientation is unknown. In addition, different nodes can have different local sensor arrangements.

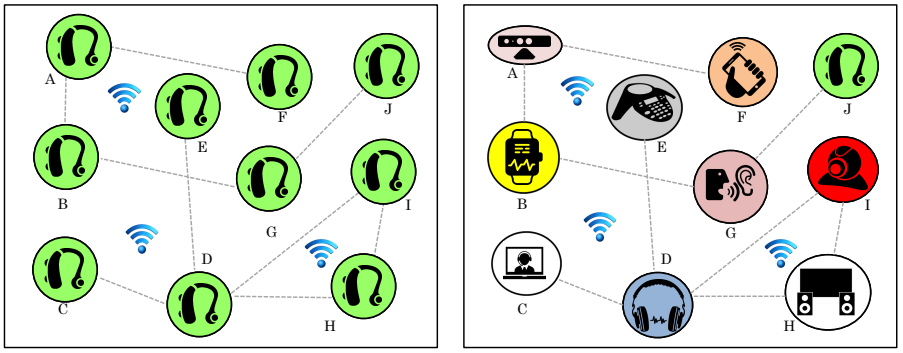
1.1.2 Wireless acoustic sensor networks (WASNs) and applications

WSNs have been applied to a large variety of applications [18], such as event detection applications [19–21], surveillance and navigation systems [22, 23], and enviromental and civilian applications in urban areas [24]. Depending on the context in which they are used, different subclasses of WSNs such as wireless body area networks (WBANs) [25, 26] or wireless multimedia sensor networks (WMSNs) [27, 28] have emerged over the last few years.

WMSNs are designed to deal with the acquisition and processing of multimedia content for which data typically has to be sampled, processed and transmitted at a higher rate than in traditional low-power and low-rate WSNs. Hence, nodes of this particular type of WSNs indeed demand greater processing power, communication resources and therefore they exhibit higher power consumption as compared to most other types of WSNs. WMSNs have been used in applications such as image enhancement [29] and acoustical applications, where in the latter a particular type of WMSNs, namely **wireless acoustic sensor networks (WASNs)** [10, 30, 31] is used. In WASNs, the sensing unit at each node consists of a single microphone or a microphone array to let the nodes apply a multi-channel acoustic SP technique in a cooperative fashion. Such SP techniques may include acoustic multi-channel noise reduction [10, 32–37], echo cancellation [38, 39], signal dereverberation [40, 41], estimation and equalization of room acoustics [42], source localization and beamforming [43–48], source separation [49–52], or active noise control and equalization [53–58]. In real-life scenarios, WASNs can be formed by connecting heterogeneous devices such as HAs, CIs, portable devices, ANC headphones, etc. An example of a homogeneous and heterogeneous WASN is illustrated in Figure 1.1.

In this thesis, we focus on designing distributed algorithms for real-time signal[1], parameter and subspace estimation in WASNs formed by either homogeneous or heterogeneous nodes. The term distributed here means that the whole processing task of the WSN is intended to be shared between the nodes, rather than in a central point (see also Section 1.2.1). To achieve these goals, the

[1]To keep up with the actual time in the case of real-time signal estimation, for each acquired block of sensor observations, each node needs to estimate a new block of samples of its desired signals in real-time.



(a) Homogeneous WASN consisting of a multitide of similar HAs with the same embedded algorithms.

(b) Heterogeneous WASN consisting of different devices, e.g., A: Kinect sensor, B: smartwatch, C: laptop, D: ANC headphone, E: conference phone, etc. with different embedded algorithms.

Figure 1.1: Example of a WASN which is formed by either (1.1a) homogeneous or (1.1b) heterogeneous nodes. In both cases, nodes share their microphone signals with each other to improve their local estimates.

devised distributed algorithms require the nodes to exchange their *compressed* sensor signal observations to be able to simultaneously enhance their estimates, while only transmitting each acquired block of sensor signal observations once. This will be discussed in more details in Section 1.2.2.

1.2 Signal processing in WSNs

In order to estimate the signals, parameters or subspaces of interest, the nodes of a WSN may transmit their sensor signal observations to a dedicated (remote) facility, often called a fusion center (FC). This FC then collects, combines and processes[2] all sensor signal observations of the entire WSN to estimate the desired signals, parameters or subspaces. However, it is common that such an FC is either unavailable or too far away from some nodes so that it leads to a significant increase in the energy consumption at each node, which is destructive to WSNs, especially to WASNs operating at high sampling rates. Moreover, such an FC-based processing scheme suffers from a single point failure, i.e., if the FC fails then the whole information processing of the WSN will fail.

[2]This sequence of collecting, combining and processing of data is often referred to as *data fusion*.

Alternatively, an *in-network* processing is often more desired, since in this setting the entire signal processing of the WSN is carried out by the nodes inside the network, rather than in an FC. This Section explains the variations of in-network processing and discusses the possible design strategies for realizing such a processing scheme.

1.2.1 Centralized vs. distributed processing

To contribute to the estimation process of the desired signals, parameters or subspaces in a WSN leveraging an in-network processing, each node could transmit *all* its sensor signal observations to all the other nodes. This method of processing is referred to as *centralized processing* and results in the *best* feasible estimation performance at each node, since its corresponding SP technique will efficiently exploit the full coherence between all the available sensor signal observations of a WSN. As a result, the estimates obtained from the centralized processing are considered as the *optimal* solution in terms of the estimation performance and hence often used as an upper-bound benchmark for evaluation purposes. An example of a centralized processing in a binaural HA system, i.e., a 2-node WASN, is depicted in Figure 1.2, in which both nodes compute their centralized spatial filter, namely CF1 and CF2, based on the union of all the microphone signals.

Although the centralized processing results in the best estimation performance, it unfortunately exhibits the worst case performance in terms of communication bandwidth efficiency, as nodes need to transmit all their sensor signal observations. This means that centralized processing is not scalable in terms of communication bandwidth requirements and computational complexity in that adding an extra node has a direct impact on the computational load and data traffic of the WSN. An extreme case is that the total number of channels including in the network-wide signal becomes too large such that a node (or an FC) is unable to handle the whole processing burden.

Alternatively, an in-network processing can be realized via *distributed processing*, which is often more favorable since it significantly reduces the communication load and it lets the heavy computational burden of the centralized processing be shared between the physically-distributed nodes of a WSN (see Figure 1.3). To achieve such a distributed processing scheme, each node locally processes its own sensor signal observations, possibly compresses them with (compressed) signal observations received from its neighbors, and then shares the results with its neighboring nodes. Therefore, since this local processing compresses the exchange signals it allows nodes to communicate with reduced bandwidth and energy resources. Moreover, by relying on nearest-neighbor transmissions,

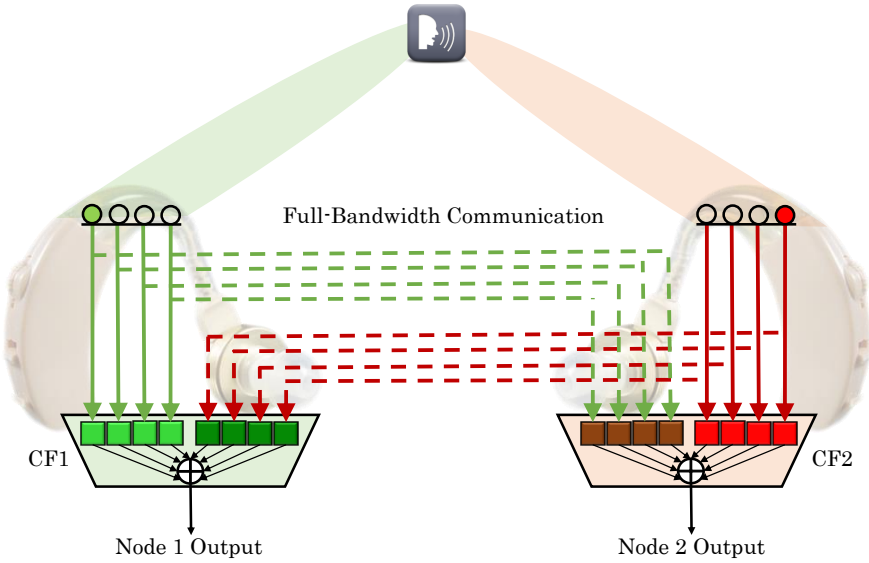


Figure 1.2: Example of centralized processing in a binaural HA system, leading to the best possible estimation performance but to the worst communication bandwidth efficiency.

distributed processing can allow for WSNs which are scalable in terms of the communication bandwidth requirements and computational complexity.

1.2.2 Signal vs. parameter exchange in distributed processing

In general, there exist two large classes of distributed estimation problems: *signal* estimation and *parameter* estimation. In parameter estimation problems, usually a slow-varying target vector containing few parameters needs to be estimated and tracked over time, whereas in signal estimation problems for each corresponding block of samples collected at the sensors, a new block of samples of the desired signals needs to be estimated. Therefore, in signal estimation problems the number of variables to be estimated grows linearly with the sample time.

This thesis aims at designing distributed algorithms for both real-time signal and parameter estimation so that the nodes are able to simultaneously enhance their estimates through collaboration. In order to achieve this, the developed distributed algorithms require the nodes to exchange their compressed sensor

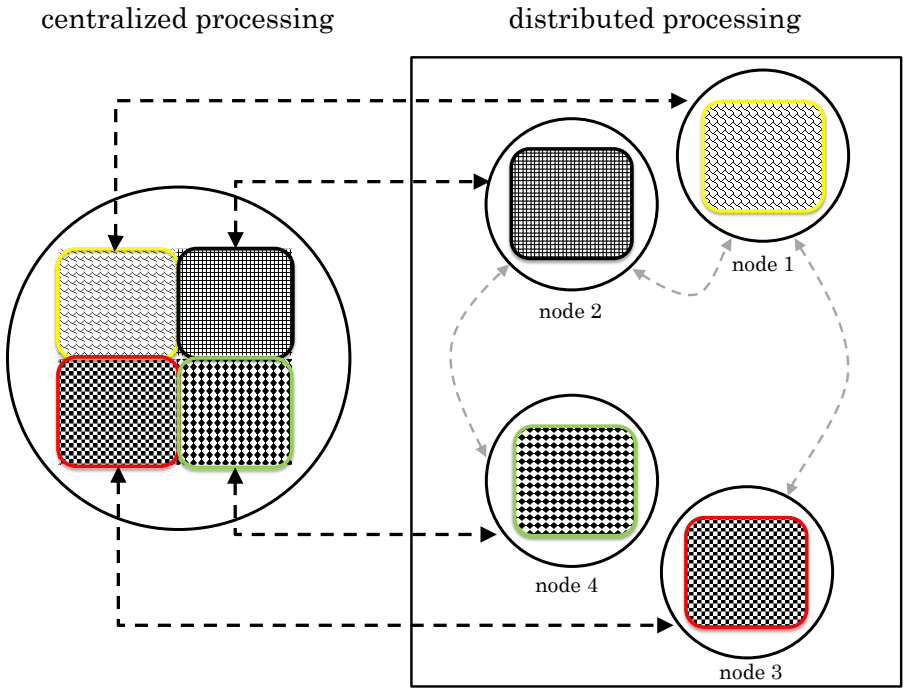


Figure 1.3: Distributed processing scheme (right) where the heavy computational burden of a centralized processing point (left) is shared between the nodes that are exchanging their processed signals with each other. The squares with the different patterns are associated with the computational power required by different nodes.

signal observations. Nevertheless, it is indeed important that this signal exchange adopts an efficient communication scheme and further allows to fulfill the real-time signal and parameter estimation requirements. In the distributed algorithms developed here, this is achieved by restricting the actual compressed sensor signal observations that are exchanged between the nodes to being transmitted *only once* for each new acquired block of sensor signal observations. Interestingly, this leads to distributed algorithms which iterate over *time* and not over the same block of sensor signal observations. It is emphasized that this is unlike cases where only derived parameters are required to be exchanged between the nodes, rather than the actual compressed sensor signal observations. The latter often happens in parameter estimation problems where algorithms inherently iterate over the same block of sensor signal observations, for which

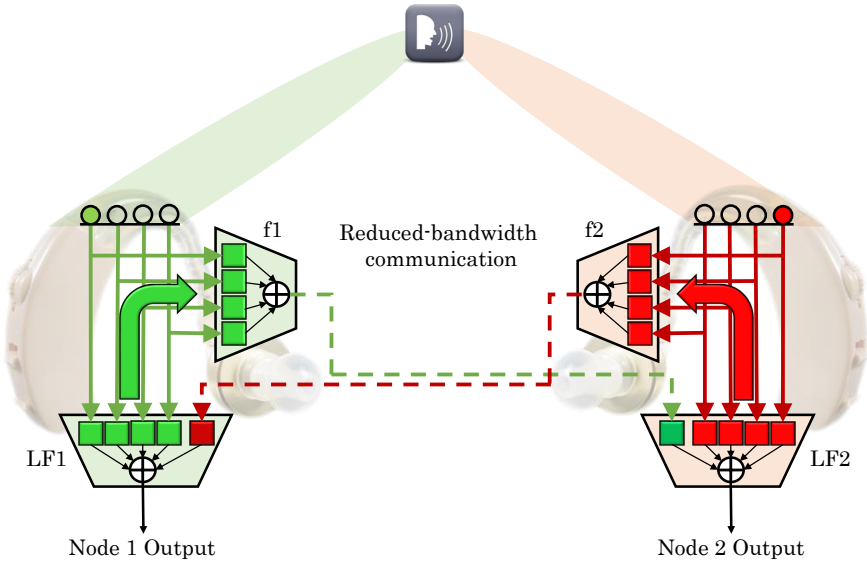


Figure 1.4: Example of a distributed algorithm adopting compressive filter-and-sum operations in a binaural HA system.

the nodes are required to re-transmit and re-estimate the parameters until they converge to an equilibrium. Therefore, local processing techniques that are often used in the context of distributed parameter estimation, such as consensus [59, 60], incremental [61], or diffusion [62–65] based algorithms, are often inefficient for signal estimation due to their inherent iterative re-estimation of the target vector. If such algorithms would be used for, e.g., speech enhancement, a new instance of this iterative algorithm would have to be started from scratch for each acquired block of sensor observations [35–37, 66]. Therefore, we here focus on distributed estimation techniques that update distributed subspaces or spatial filter coefficients over time, without iterating on the same block of sensor observations that are fed through these distributed quantities[3].

By relying on compressive *filter-and-sum* operations, the developed distributed algorithms let the nodes significantly reduce their required communication bandwidth. The per-node compressors are then recursively updated using the acquired blocks of sensor observations, allowing the iterations of the distributed algorithms to be spread out over time and avoiding re-transmission and re-

[3]In the case of real-time signal estimation, the fact that neither re-transmission nor re-estimation of each acquired block of sensor observations is involved enables the nodes to better satisfy the delay constraints.

estimation over the same block of sensor observations. Note that this linear compression is unlike what often occurs in typical encoding (compression) and decoding (decompression) schemes, in that after nodes compress and broadcast their sensor signal observations using this filter-and-sum compression scheme other receiving nodes do not apply any decompression to these compressed signals. An example of a distributed algorithm adopting such compressive filter-and-sum operations in a binaural HA system, i.e., a 2-node WASN, is depicted in Figure 1.4, where the filter-and-sum compression rules are depicted as $\mathbf{f}_1$ and $\mathbf{f}_2$ (compare to the example using centralized processing in Figure 1.2). Devising this type of compression rules for WSNs, however, in general poses several challenges to the design procedure, especially in the case of heterogeneous WSNs in which despite having different setups in different nodes, the goal of the distributed algorithm may still be to obtain the corresponding centralized estimates (more discussion in Section 1.2.3, 1.3). For certain homogeneous WSNs and under certain conditions, the per-node compression rules that allow to achieve this goal have been found as part of the local spatial filters of the nodes ([33, 34, 67–69]). This makes the per-node compression rules *algorithm-dependent*, since at each node it depends on the same SP technique that the node uses to solve its estimation problem. This is illustrated in the example of Figure 1.4, where the compression rules $\mathbf{f}_1$ and $\mathbf{f}_2$ are derived from the local spatial filters LF1 and LF2, respectively. The distributed algorithms of this thesis aim at leveraging similar compressive filter-and-sum strategies across further homogeneous and heterogeneous WSNs with different SP techniques to possibly reduce the required communication bandwidth, while maximizing the estimation performance at each node.

Dependency of these compression rules on the local spatial filters, however, leads to a chicken-and-egg problem, since the computation of the local spatial filters in turn requires the per-node compression rules. Therefore, the compression rules of the developed algorithms need to be first initialized with random entries and then, in a time-recursive fashion, iteratively updated (adapted) to allow better future estimates. As a result, successive updates of the local spatial filters (and hence the compression rules) use different blocks of sensor observations, where the updated filters will only be used for future observations. These updates can happen sequentially, i.e., only one node at a time (e.g., in [67]) or simultaneously at all nodes (e.g., in [68]).

1.2.3 Top-down vs. bottom-up distributed processing

Two different design strategies can be followed for designing distributed algorithms, referred to as *top-down* and *bottom-up* strategies. In a top-down strategy, first an ultimate centralized goal with respect to some a-priori known

design constraints is set and then the per-node operations are tailored accordingly such that the nodes can attain the same ultimate goal via distributed processing. Therefore this strategy requires some *upper-layer coordination* that needs to be established prior to operation of a WSN in order to effectively control and trigger the way nodes cooperate with each other. As a result, each node is fully aware of the SP technique being used by each and every other node and hence interacts according to a rigid, pre-defined routine. Examples of this can be found in [10, 33, 34, 67, 69–71], where in all cases the attempt is to design a set of pre-defined operations including the choice of the algorithm-dependent compression rules, and dimension of the exchange signals, in which case the nodes have been proven to iteratively converge to their centralized performance, i.e., to the same performance that they could achieve if they would have access to all the raw sensor signal observations of the entire WSN. However, it should be noted that it is not always possible to guarantee the convergence and optimality of a distributed algorithm, since the extent to which the solutions can reach the centralized performance heavily depends on the way the compression rules are chosen and on the nature of the SP technique that the nodes use to solve their estimation problems [72].

WSNs can alternatively be formed in an *ad-hoc* fashion, where no a-priori design constraints and/or upper-layer coordination are applicable or permitted. In such cases, a bottom-up strategy can be followed to allow the nodes to immediately start exchanging their signals, even though they are not provided with a set of pre-defined operations and they are not aware of the SP techniques being used by the other nodes. The design then deals with investigating which kind of application-dependent compression rules and how many compressed signals can facilitate the resulting distributed algorithm to deliver a performance which is *better* than the one that could be achieved by an uncontrolled strategy. Such uncontrolled strategies can for instance include cases where nodes would only share part of their raw sensor signal observations, or cases where nodes would use application-independent (e.g., a random) linear compression rules to simply achieve a reduced-bandwidth solution. These cases are often used as lower-bound benchmarks when evaluating the performance of bottom-up distributed algorithms. Note that in general such bottom-up distributed algorithms do not necessarily aim at achieving the centralized (optimal) performance[4]. Therefore, these algorithms often lead to heuristic or suboptimal solutions (as in e.g., [31, 49, 73, 74]), where the centralized (optimal) performance can be viewed as an upper-bound benchmark for evaluation purposes.

[4]It is possible that there exists no per-node operations with which nodes could achieve the centralized perfromance with respect to the design constraints, e.g., when the SP problems are not inter-related [72]. In such cases, a bottom-up strategy can be applied.

1.3 Multi-task WSNs

1.3.1 Node-specific vs. network-wide estimation problems

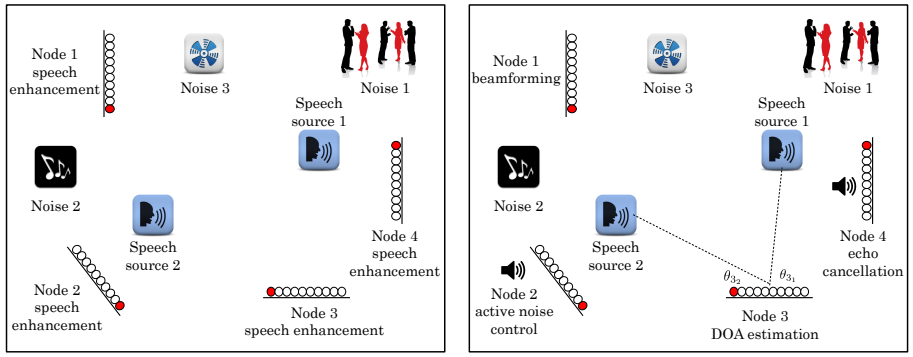
One class of distributed algorithms deals with WSNs in which all the nodes cooperate with each other to solve a *single* network-wide SP task. We will refer to such WSNs as single-task WSNs (sometimes referred to as 'multiple devices for a single task' (MDST)). Examples of distributed algorithms for single-task WSNs can be found in [16, 61, 75–78]. Single-task WSNs are not specifically addressed in this thesis.

Another class of distributed algorithms deals with WSNs in which all the nodes cooperate with each other to solve *multiple node-specific* SP tasks. We will refer to such WSNs as multi-task WSNs ('multiple devices for multiple tasks' (MDMT)). Each node of such multi-task WSNs is interested in estimating a different set of desired signals, leading to different node-specific tasks which are somehow related since the desired signals as well as the noise are often highly correlated across the sensors of different nodes[5]. This is of great importance in applications where the spatial information of the signals needs to be preserved, for example in WASNs where each node is required to estimate a different node-specific observation of the same audio signal(s) [10], or in binaural HAs or CIs where each device needs to preserve its spatial cues to allow directional hearing [34, 86], or when each node needs to run a localization on the denoised desired signal estimates of its own sensors.

1.3.2 Homogeneous vs. heterogeneous multi-task WSNs

Multi-task WSNs can comprise of cases where the node-specific tasks all rely on the *same* basic SP technique, such as signal enhancement, ANC, beamforming, spectrum estimation, subspace estimation, or DOA estimation. In the context of this thesis, such multi-task WSNs are classified as **homogeneous multi-task WSNs**, since all these nodes locally apply the same basic SP technique to their sensor signal observations (as part of their local SP operations). Moreover, each node of a multi-task WSN may simultaneously apply multiple SP techniques to accomplish its task (see Chapter 4). These cases are also considered as homogeneous multi-task WSNs, as long as the same SP techniques are identically applied at all nodes. Several distributed algorithms for homogeneous multi-task WSNs have been devised in which the basic SP technique is the Multi-channel Wiener Filter (MWF) [34, 67], the minimum variance distortionless response

[5] Also in the context of distributed parameter estimation, large body of works deal with node-specific estimation problems, e.g., [79–85].



(a) (b)

Figure 1.5: Example of a (1.5a) homogeneous multi-task WSN and (1.5b) heterogeneous multi-task WSN. Each node is interested in estimating the signal or parameter as observed by its local microphones.

(MVDR) beamformer [33], the linearly constrained minimum variance (LCMV) beamformer [69] or direction-of-arrival (DOA) estimation [87]. These basic tasks will be explained in Section 1.5.

Multi-task WSNs may however be formed by cases where the node-specific tasks rely on *different* basic SP techniques. In the context of this thesis, such multi-task WSNs are classified as **heterogeneous multi-task WSNs**. Devices that form such heterogeneous multi-task WSNs can have the same (e.g., WASN in Figure 1.1a) or diverse (e.g., WASN in Figure 1.1b) hardware capabilities and resources, similar to what is envisaged in the context of the Internet of things (IoT). For instance, a heterogeneous multi-task WASN in this category can be formed by wirelessly connecting devices such as HAs, laptops or smartphones, while each of these devices applies a different SP technique and/or performs a different task. The design of distributed algorithms for such heterogeneous multi-task WSNs often poses more challenges compared to the former cases.

Examples of multi-task WASNs for both cases where the node-specific tasks rely on the same basic SP technique and on different basic SP techniques are illustrated in Figure 1.5.

1.4 Challenges of distributed algorithm design for WSNs[6]

1.4.1 Communication bandwidth

The most challenging aspect of designing SP algorithms for WSNs is to efficiently allocate the limited power available at each node, since this directly affects the lifetime of the entire network. In fact, the most power-hungry operation at each node is known to be the wireless transmission of the signals, even when employing state-of-the-art low-power technologies [88, 89]. Therefore, designing distributed algorithms that efficiently reduce the required communication bandwidth with minimal impact on the estimation performance of the per-node tasks is of great importance.

To reduce transmitted data over wireless channels of a WSN, several general-purpose compressive schemes based on encoding (compression) and decoding (decompression) have been proposed [90, 91]. However, when these schemes are employed in distributed algorithms which rely on spatial filtering techniques, essential information and cues for the SP task may get lost, leading to reduced estimation performance and hence to suboptimal solutions [92–96]. Moreover, when such compressive schemes are used, nodes need to apply a decompression when they receive the compressed sensor signal observation from the other nodes. This often increases the per-node computational complexity and therefore demands more computational power and energy resources, which is not desired in WSNs.

In this thesis an alternative compression strategy is adopted, whereby application-dependent compressive filter-and-sum operations are used to reduce the required communication bandwidth. This linear compressive method essentially *fuses* multi-sensor observations to obtain a signal with fewer channels than the original number of sensors. To achieve this while maximizing the performance of the per-node tasks at hand, the compression rule at each node is chosen as part of the local spatial filters obtained from the SP technique employed by the node to attain its own task (see figure 1.4). This compressive filter-and-sum scheme can either be lossless for the SP task at hand, i.e., the centralized solution can be reached, or it is lossy and delivers suboptimal solutions, in which case it allows the nodes to readily employ the received data

[6] To avoid providing an exhaustive list, this section only discusses the particular challenges that are often addressed in the developed distributed algorithms of this thesis. However, it is noted that multitude of implementation challenges such as sampling frequency synchronization, wireless communication link failures and delays, voice activity detection (VAD) issues, and privacy constraints are also often present in practical scenarios (see also Section 8.2).

without applying any decompression. Moreover, it is noted that this linear compression can be combined with other additional (lossy) coding schemes, as in [97].

1.4.2 Efficient use of exchange signals for integrated tasks

Many applications require that multiple basic SP techniques are simultaneously combined at each node to fulfill a certain task. For instance, the speech enhancement task at node 1 of the WASN in Figure 1.5a may need to combine spatial beamforming, ambient noise reduction, echo and feedback cancellation and dereverberation of its microphone signals to achieve a desirable performance. Such combination is indeed more favorable and efficient when these basic SP techniques at each node come as an integrated solution, rather than a concatenation of the individual SP techniques. Several integrated problems have been proposed in audio and speech enhancement applications, namely an integrated noise reduction and echo cancellation [98], an integrated active noise control and noise reduction [99], an integrated speech dereverberation and noise reduction [100, 101], and an integrated noise reduction and dynamic range compression [102]. When such integrated schemes are intended to be realized in nodes of a WSN, it is eminently more desirable that the compression rules of the distributed algorithms allow the exchange signals to simultaneously enhance the performance of all the basic SP techniques.

1.4.3 Robustness to adverse scenarios

Inevitable adverse conditions caused by physical surroundings and/or malfunction of other auxiliary algorithms that run in conjunction with a certain basic SP technique may hamper proper functionality of a distributed algorithm. Hence, when designing these algorithms, it is crucial to value the robustness aspects such that the resulting solutions can cope with such poor conditions to the utmost. For instance, in speech enhancement applications, the distributed algorithms which are more robust to microphone phase and gain mismatches, non-stationary noise and/or erroneous voice activity detectors (VADs) are indeed much more favorable [10, 103].

1.4.4 Per-iteration computational complexity

Since the processing power of each node of a WSN is often assumed to be limited, often only a *scaled down* or *reduced-dimension* version of the

centralized task can be processed and handled by each node at each iteration of a distributed algorithm. Therefore, the entire processing burden should be properly distributed between the nodes, as otherwise the per-node computational complexity may become too high leading to DSP overruns or real-time performance failure. In the distributed algorithms designed in this thesis, these per-iteration complexity reductions directly result from the filter-and-sum compression applied to the exchanged signals. The latter is due to the fact that in these distributed algorithms, the nodes have only access to a subset of the full sensor signal observations and hence carry out their operations on reduced-dimension matrices, when compared to the dimension of matrices in the centralized case.

1.4.5 Adaptivity and convergence speed

In dynamic scenarios, environmental parameters and physical characteristics of a WSN may slightly or even abruptly change over time. Hence distributed algorithms should be able to track and adapt to such changes. For instance, in audio and speech processing applications, position of the nodes and/or audio sources as well as the statistics of the signal components may change over time, hence the distributed algorithm should be able to adapt accordingly.

The adaptation is here achieved by effectively updating the spatial filters at the different nodes (and hence the compression rules) from iteration to iteration in a *data-driven* fashion, where the iterations are spread out over time in a time-recursive block-wise fashion. For this, it is also important that the nodes update their parameters with a sufficiently high frequency to be able to track and adapt to the changes in dynamic environments, e.g., by allowing simultaneous updating [68].

Along with the feature to adapt to environmental changes, in many applications it is also beneficial if a distributed algorithm can adapt to hardware discrepancies, e.g., sensor mismatches, or sensor failures, without requiring an explicit learning or calibration phase prior to its operation [104, 105].

1.4.6 Blindness

The positions and orientations of the nodes of a WSN are often unknown in practice, as nodes are often deployed randomly. Therefore, the relative geometry and the between-nodes signal coherence model cannot be effectively leveraged in many practical scenarios. Such cases therefore require *blind* distributed

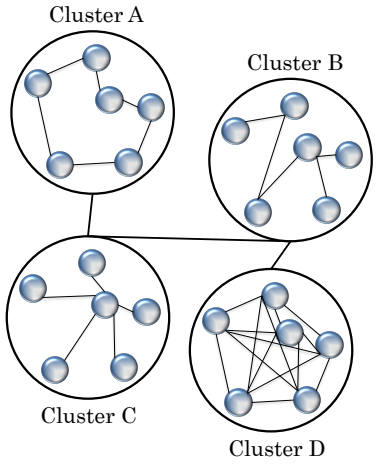


Figure 1.6: Typical WSN topologies: cluster A: ring topology, cluster B: tree topology, cluster C: star (centralized) topology, cluster D: fully-connected topology. The whole WSN can be viewed as an example of a mixed-topology case.

algorithms which do not rely on the knowledge of the positions and orientations of nodes for their routine operations.

1.4.7 Network topology

In essence, a network topology of a WSN is imposed by the set of communication links that its corresponding distributed algorithm needs to establish in order to perform its routine operations. Although this network topology can be chosen independently of the physical location of the nodes, in practice those which avoid communicating over long distances or through obstacles are more efficient in terms of link quality and energy consumption. Four typical basic topologies of WSNs, namely ring, tree, star and fully-connected, are shown in Figure 1.6. To enhance a particular aspect of a WSN in some conditions, such as convergence speed, distributed algorithms may operate on a combination of such basic topologies [106].

Distributed algorithms in which the topologies allow nodes to only communicate over a short range, a.k.a *nearest-neighbor* topologies, are often more preferred in practice. This is not only because of their promising energy efficiency, but also because they let the frequency spectrum become re-usable in different cells

or segments of WSNs. Nevertheless, when these nearest-neighbor topologies are realized within an uncontrolled ad-hoc WSN, they may likely comprise of several cycles in their network graph, i.e., paths that start and end in the same node. Contrary to the majority of (ad-hoc) distributed parameter estimation algorithms which require exchange of parameters between nodes [59, 60, 63, 107], the distributed algorithms which rely on compressive filter-and-sum operations to exchange compressed signal observations cannot handle such cycles. This is because cycles introduce feedback paths and causality problems which may hamper the convergence of such distributed signal estimation algorithms [71].

Many nearest-neighbor-based distributed signal estimation algorithms have been proposed where such feedback paths are somehow avoided by means of topology control, e.g., based on a tree topology [71], a hierarchical tree-like topology [106], or based on a topology-independent scheme [108] in which a different tree is used in each iteration.

For the sake of an easy exposition and tractability, all the algorithms developed in this thesis will be presented for a fully-connected WSN. However, it should be emphasized that all results can be extended to more efficient nearest-neighbor topologies using the same strategies provided in [71, 106]. Furthermore, throughout this thesis the communication links are assumed to be noise-free without exhibiting any random failures, i.e., they are assumed to be ideal. In practice, suitable strategies must be adopted to avoid instantaneous performance degradation in real-time signal estimation for cases when possible link failures or packet loss occure (e.g., as in [104]).

1.5 Overview of the basic SP techniques

1.5.1 Beamforming

Beamforming refers to a class of multi-channel SP techniques which perform linear filter-and-sum operations, where the filters are optimized under certain design constraints [109]. In particular, an adaptive beamformer continuously minimizes its total output power under constraints that control the beam pattern or directivity pattern, i.e., 'look' and 'blocking' directions of the beamformer. In this class, the minimum variance distortionless response (MVDR) beamformer constrains the response to preserve a single target direction, while suppressing the signals arriving from all other directions [110]. An extension to the MVDR beamformer is the linearly constrained minimum variance (LCMV) beamformer, which incorporates a set of linear constraints such that multiple look directions can be preserved and blocking directions can be imposed [3, 4, 111].

For instance, in the WASN depicted in Figure 1.5b, node 1 can apply an MVDR beamformer to extract the signal of the speech source 1, while treating the remaining signal contributions as ambient noise. Alternatively, if the objective of this node would be to extract the signal of the speech source 1 and fully cancel out the signal of the speech source 2 in a controlled fashion, then it can use an LCMV beamformer. A more detailed discussion on MVDR and LCMV beamforming techniques will be provided in Chapters 5 - 6 and Appendix A.

1.5.2 Multi-channel Wiener filter (MWF)

Another class of multi-channel SP techniques consists of mean square error (MSE) based *signal estimators*. In this class, MWF is a well-known technique in which an optimal solution of a linear filter-and-sum operation is obtained based on a linear minimum mean squared error (LMMSE) between the desired signal and the MWF output signal [112, 113]. Since the desired signal in this case is defined as a mixture of the target source signals as they are observed at the array sensors, the MWF allows to preserve the spatial characteristics of the received signals. To achieve this, the MWF only relies on the second-order statistics of its sensor signal observations to estimate its desired signal, i.e., it only relies on the correlation between signals of different sensors. In addition, MWF does not require a-priori knowledge of the desired source locations and sensor characteristics (e.g., no gain or phase calibration is required a-priori) [105]. These favorable features, together with the fact that it often exhibits a more robust and efficient solution compared to many of its counterparts, have made MWF a very promising and successful SP technique in applications such as audio and speech enhancement [105, 113].

For instance, in the WASN depicted in Figure 1.5a, node 2 can apply an MWF to estimate the mixture of the signals of both speech sources as locally observed at its *reference* microphone, while suppressing the noise contributions. To estimate the required second-order statistics, a VAD may be applied in practice to distinguish the periods in which the speech signals are present from those in which they are absent (known as 'speech-and-noise' and 'noise-only' segments). A more detailed discussion about MWF will be provided in Chapter 3 and Chapter 4.

1.5.3 Direction of Arrival (DOA) and signal subspace estimation

A family of multi-channel SP techniques addresses the estimation of signal subspaces and direction of arrivals (DOAs). At a given sensor array, DOAs are

defined as the directions from which the desired signals impinge on the sensors, i.e., the directions or the locations of the target sources with respect to the sensor array. The estimated DOAs can then be used for localization purposes, or for defining the look and blocking directions when designing a beamformer. For instance, a DOA estimation technique can be utilized to let node 3 of the WASN of Figure 1.5b estimate its DOAs with respect to both of the speech sources.

In this thesis we consider *subspace-based* DOA estimation techniques, which rely on the estimation of a so-called signal and noise subspace [4]. By applying a data-driven scheme, a basis for the column space of these subspaces can be derived from the correlation matrices of the received signals [114, 115]. A more detailed discussion of DOA and signal subspace estimation will be provided in Chapter 2 and Chapter 4.

In this thesis, the required subspaces that have to be incorporated in the aforementioned SP techniques will always be estimated using a generalized eigenvalue decomposition (GEVD) of the sensor signal correlation matrices. This is essentially because of the attractive features that a GEVD-based approach exhibits in terms of the robustness in adverse conditions, when compared to e.g., an eigenvalue decomposition (EVD-) based approach. Furthermore, by incorporating a GEVD-based subspace estimation the resulting subspace estimate is immune to arbitrary sensor gains at different nodes. This is of great interest especially for heterogeneous WSNs, since it allows the resulting distributed GEVD-based algorithms to run without a need for prior gain calibration. More details will be provided in the following chapters. The resulting SP techniques will be referred to as GEVD-based MWF, GEVD-based signal subspace (or DOA) estimation and GEVD-based LCMV (or MVDR).

1.6 Overview of the thesis

1.6.1 General overview

This thesis contains the following three main parts:

The **first part** (Chapters 2 - 4) of the thesis presents distributed algorithms for homogeneous multi-task WSNs in which all the nodes apply the same basic SP technique to fulfill their tasks. In this part a GEVD-based signal subspace estimation, a GEVD-based DOA estimation and a GEVD-based MWF are particularly employed as basic SP techniques. Although in the case of distributed GEVD-based signal subspace estimation only a single centralized

estimate is defined for the entire WSN, since the subspace that each node locally estimates is node-specific due to the difference in spatial characteristics of its sensor signal observations, the resulting WSN falls into the category of multi-task WSNs. When a GEVD-based MWF or a GEVD-based DOA estimation is addressed, however, many centralized node-specific estimates are defined, i.e., one centralized node-specific estimate for each node. The design of all the distributed algorithms in this part follows a top-down strategy, under which it will be shown that the nodes iteratively (in a time-recursive fashion) obtain their centralized estimates, i.e., the estimates that they would obtain if they would have access to all sensor signal observations.

The **second part** (Chapters 5 - 6) of the thesis focuses on distributed algorithms for heterogeneous multi-task WSNs in which the node-specific tasks rely on different basic SP techniques across the different nodes. In particular, a GEVD-based DOA estimation, a GEVD-based MWF, and a GEVD-based beamformer (MVDR or LCMV) are utilized as basic SP techniques in the envisaged WSNs of this part. Contrary to Part I, the design of all the distributed algorithms in this part follows a bottom-up strategy, since nodes are not aware of each others' tasks. This in fact allows the nodes to immediately start exchanging their signals even though they are not provided with a set of pre-defined routines, e.g., for their compression rules.

The **third part** (chapter 7) of the thesis, provides a real-time experimental validation for the GEVD-based DANSE algorithms presented in Chapter 3 in speech enhancement applications.

The composition of the next chapters as well as their contributions is outlined in Section 1.6.3.

1.6.2 Thesis technical assumptions

The distributed algorithms developed in Part I and Part II of this thesis basically assume that 1) the sampling rates between the nodes are synchronized 2) the WSN has ideal communication links 3) the nodes have access to an ideal (multi-source) VAD 4) all the nodes observe the full subspace spanned by all the source signals, i.e., all the nodes of a WASN are able to hear all speech sources present in the acoustic enclosure 5) network topology is fully-connected, with the possibility of being extended to tree-topologies 6) the position of the nodes, as well as their relative geometry and hence the between-nodes signal coherence models are unknown. Moreover, different nodes can have different local sensor arrangements, while the local sensor arrangement at each node performing DOA estimation (in Chapters 4-6) is assumed to be known and 8) for theoretical convergence analyses, the assumed signal models are satisfied perfectly and that

the required signal correlation matrices can be estimated error-free using infinite observation windows. In practice, with finite observation windows (e.g., when the problem is solved in the short-time Fourier transform (STFT) domain), an asymptotic convergence should be expected for which the approximation accuracy improves when larger observation windows are used.

1.6.3 Thesis outline

Chapter 2 presents two distributed algorithms under which nodes of a WSN obtain their corresponding part of the centralized signal subspace estimate in a distributed fashion. In order to achieve robust estimates, the signal subspaces in both cases are extracted based on a GEVD-based method. The proposed distributed algorithms are shown to converge to the centralized solution, while reducing the per-node communication and computational cost. The results of this chapter will be used in Chapter 3 and Chapter 6.
Chapter 2 is a combination of the following two publications:

- **Hassani A.**, Bertrand A., Moonen M., "Distributed signal subspace estimation based on local generalized eigenvector matrix inversion", in Proc. of the European Signal Processing Conference (EUSIPCO), Nice, France, Sep. 2015, pp. 1401-1405.
- **Hassani A.**, Bertrand A., Moonen M., "Distributed GEVD-based signal subspace estimation in a fully-connected wireless sensor network", in Proc. of the European Signal Processing Conference (EUSIPCO), Lisbon, Portugal, Sep. 2014, pp. 1292-1296.

Chapter 3 addresses the problem of distributed estimation of desired signals from noisy sensor signal observations based on an MWF in which a GEVD-based low-rank approximation of signal correlation matrices is incorporated. In this setup, each node estimates its own node-specific desired signal as received by its own sensors (to preserve the spatial characteristics of its sensor signals), thereby the distributed algorithm addresses many (different) node-specific estimation problems. The attempt to realize this particular distributed algorithm starts from the original DANSE algorithm, following by including GEVD-based low-rank approximations in the per-node local MWFs. Compared to the original DANSE algorithm, the proposed GEVD-based DANSE algorithms brings the following extra benefits: (1) it provides more robust estimates in the presence of second-order statistics errors, e.g., due to non-stationarity of noise, low-SNR conditions or erroneous VADs, (2) the conditions under which it converges to the centralized solutions are less strict, (3) allows for further compression of the

signal observations transmitted between the nodes.
Chapter 3 has been published as:

- **Hassani A.**, Bertrand A., Moonen M., "GEVD-Based Low-Rank Approximation for Distributed Adaptive Node-specific Signal Estimation in Wireless Sensor Networks", IEEE Transactions on Signal Processing, vol. 64, no. 10, May 2016, pp. 2557-2572.

Chapter 4 provides a distributed algorithm for parameter estimation under which nodes cooperate to estimate their node-specific DOAs with respect to a set of target speech sources, within a WASN where localized undesired sources (such as interfering speakers) or diffuse noise components (such as multi-talker babble noise) obstruct the node-specific DOA estimation tasks. To achieve robust node-specific DOA estimates at each node in such conditions, the GEVD-based DANSE algorithm developed in Chapter 3 is first employed as a preprocessing step to denoise all the microphone signals of the entire WASN. While each node could then merely use its (optimally) denoised local microphone signals to estimate its own DOAs, it is shown that further inclusion of the compressed signals of the GEVD-based DANSE algorithm can significantly improve these node-specific DOA estimates. These per-node improvements are remarkable, since they are achieved even though (1) different nodes have different DOAs depending on their position, orientation and positions of target speech sources, (2) the position of the nodes, as well as their relative geometry and hence the between-nodes signal coherence models are unknown, (3) the compressed signals of the GEVD-based DANSE have actually been optimized to improve the noise reduction performance (and not the DOA estimation performance). By re-using these compressed signals, the proposed distributed algorithm provides a cooperative integrated noise reduction and DOA estimation scheme, where the noise reduction (i.e., GEVD-based DANSE's final filtering stage) can actually be shortcut to either let the nodes estimate both their node-specific desired signals and their node-specific DOAs simultaneously, or to reduce the per-node computational cost when only the DOA estimation task has to be performed on the denoised signals.
Chapter 4 has been published as:

- **Hassani A.**, Bertrand A., Moonen M., "Cooperative integrated noise reduction and node-specific direction-of-arrival estimation in a fully connected wireless acoustic sensor network", Signal Processing, vol. 107, Feb. 2015, pp. 68-81.

Chapter 5 deals with distributed estimation in a particular heterogeneous multi-task WASN, in which some of the nodes aim at estimating their node-specific DOAs with respect to a single desired speech source, while other nodes

aim at implementing either an MWF or an MVDR beamformer in order to estimate the signal of the desired source as impinges on their local microphones. As a result, unlike the envisaged WSNs in the previous chapters, the nodes of this heterogeneous multi-task WSN rely on different SP techniques to attain their tasks, leading to per-node filter-and-sum compression rules that is now extracted based on different SP techniques. This chapter then proposes a distributed algorithm which allows each node to immediately start cooperating with other nodes in such circumstances, without having any prior knowledge about the SP techniques other nodes use to attain their tasks. Following a bottom-up strategy, the proposed distributed algorithm remarkably allows the nodes to obtain the centralized solution of their corresponding node-specific task, while cooperating with a single-channel per-node signal transmission.
Chapter 5 has been published as:

- **Hassani A.**, Plata-Chaves J., Bertrand A., Moonen M., "Multi-Task Wireless Acoustic Sensor Network for Node-Specific Speech Enhancement and DOA Estimation", in Proc. of the IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM), Rio De Janeiro, Brazil, Jul. 2016, pp. 1-5.

Chapter 6 extends the distributed algorithm of Chapter 5 to a heterogeneous multi-task WSN where multiple sources are present and where nodes can have their freedom to choose the way they treat different sources, i.e., they can define their own desired/interfering sources. To realize such node-specific constraints, the MVDR beamformers used in Chapter 5 are replaced by LCMV beamformers. This replacement, however, poses a great challenge since the LCMV nodes now require to partition their nodes-specific estimates of their desired and interfering sources, whereas other DOA or MWF nodes only estimate their full (joint) subspace of all the sources (excluding possible noise sources). The analysis provided in this chapter will reveal that the compatibility between these heterogeneous nodes is effectively achieved by using a GEVD-based subspace estimation (including the per-node joint, desired and interfering subspaces). Furthermore, in this setup the nodes do not know the tasks of the other nodes, hence a bottom-up strategy is followed. Despite this fact, and even though nodes solve very different signal processing tasks, the proposed distributed algorithm let the nodes cooperate with reduced communication resources and allows each node to attain the centralized solution of its corresponding node-specific task.
Chapter 6 has been published as:

- **Hassani A.**, Plata-Chaves J., Bahari M.H., Moonen M., Bertrand A., "Multi-Task Wireless Sensor Network for Joint Distributed Node-Specific Signal Enhancement, LCMV Beamforming and DOA Estimation", IEEE

Journal of Selected Topics in Signal Processing, vol. 11, no. 3, Apr. 2017, pp. 518 - 533.

Chapter 6 also references Appendix A which proposes a subspace projection-based method to improve the performance of an LCMV beamformer. This method is shown to be highly beneficial, especially in scenarios when only a limited number of relevant samples are available to accurately estimate the subspaces. Appendix A has been published as:

- **Hassani A.**, Bertrand A., Moonen M., "LCMV beamforming with subspace projection for multi-speaker speech enhancement", in Proc. of the IEEE International Conference on Acoustics, Speech and Signal processing (ICASSP), Shanghai, China, Mar. 2016, pp. 91-95.

Chapter 7 aims to evaluate the speech enhancement performance of the GEVD-based DANSE algorithm which is presented in Chapter 3. To achieve this, a fully adaptive and realistic scenario is created using an acoustic sensor network with three collaborating microphone arrays. The output signals of the distributed algorithm are assessed by means of instrumental measures for both speech intelligibility and speech quality.
The material of Chapter 7 has been the topic of a "show-and-tell" demonstration session at the IEEE International Conference on Acoustics, Speech and Signal processing (ICASSP), New Orleans, USA, Mar. 2017.

Chapter 8 concludes this thesis.

Part I

Distributed signal processing algorithms for homogeneous multi-task WSNs

Chapter 2

Distributed algorithms for GEVD-based signal subspace estimation

Distributed GEVD-based signal subspace estimation in a fully-connected wireless sensor network

Amin Hassani, Alexander Bertrand and Marc Moonen

Published in proceedings of the 22nd *European Signal Processing Conference (EUSIPCO)*, Lisbon, Portugal, Sep. 2014, pp. 1292-1296, DOI: 10.1109/EUSIPCO.2015.736261.

Distributed signal subspace estimation based on local generalized eigenvector matrix inversion

Amin Hassani, Alexander Bertrand and Marc Moonen

Published in proceedings of the 23rd *European Signal Processing Conference (EUSIPCO)*, Nice, France, Sep. 2015, pp. 1401-1405, Electronic ISBN: 978-0-9928-6261-9.

2.1 Introduction

Many array-processing algorithms or applications require the estimation of a target signal subspace and their output performance heavily depends upon how accurately this subspace is estimated [4,16]. For instance the direction-of-arrival (DOA) estimation performance of algorithms like MUSIC [114] or ESPRIT [115] strongly depends upon a signal subspace estimation. Moreover, in the field of adaptive beamforming, an imprecise signal subspace estimation often results in a significant performance degradation [116]. We consider the problem of network-wide signal subspace estimation in a fully-connected[1] wireless sensor network (WSN), with multi-sensor nodes and where the noise is possibly spatially correlated. Although the per-node signal subspace can be estimated locally without any signal exchange between nodes, the network-wide signal subspace provides better estimates, since more correlation structure can be exploited (as demonstrated in [118, 119]). Furthermore in some applications such as WSN positioning, the network-wide relative geometry between the nodes has to be captured in the network-wide signal subspace. In a WSN, the straightforward estimation of a network-wide signal subspace would require a centralization of all the sensor signals to compute network-wide covariance matrices. However this centralization of data requires the availability of a sufficiently powerful fusion center and demands a significant per-node communication cost. In this chapter we propose an alternative distributed algorithm to obtain the centralized estimation performance, without such data centralization and hence without explicitly constructing a network-wide sensor signal covariance matrix.

The network-wide signal subspace can be estimated using an eigenvalue decomposition (EVD) of the network-wide sensor signal covariance matrix, where part of the eigenvectors directly corresponds to the underlying signal subspace. However, a generalized EVD- (GEVD-) based counterpart delivers a better estimation performance for scenarios with spatially correlated noise, assuming that the noise covariance is either known a-priori or can be estimated, e.g., based on 'noise-only' signal segments [1, 119]. Furthermore as discussed in [118, 119], the GEVD is immune to arbitrary sensor gains at different nodes (e.g., if the sensor gain is not calibrated between the nodes). Hence we consider a GEVD-based method for the estimation of the network-wide signal subspace. When a GEVD is employed, the actual network-wide signal subspace can be extracted by the inversion of a matrix containing all the network-wide generalized eigenvectors (GEVCs) [119, 120]. Two different attempts to estimate

[1] For the sake of an easy exposition, we only consider the case of a fully-connected WSN, but it is noted that all results in this chapter can be extended to partially-connected WSNs, using similar techniques as in [117].

this network-wide signal subspace in a distributed fashion have been proposed in [119] and [120], as follows:

- First *only* the S local principal GEVCs are estimated using the distributed adaptive covariance-matrix generalized eigenvector estimation (DACGEE) algorithm [1], where S is the a-priori defined dimension of the signal subspace. A technique in [119] has been proposed that allows to transform the estimated principal GEVCs into a set of basis vectors that span the actual signal subspace, i.e., without the need to also compute the other GEVCs. This method is referred to as the *correlation-based* method and will be explained in Section 2.4.2.
- In [120] an alternative distributed algorithm has been proposed in which each node first uses DACGEE to estimate the matrix containing *all* the local GEVCs in each iteration, and then applies a local matrix inversion to estimate part of the network-wide signal subspace. Remarkably, while these inverted matrices are based on the *local* GEVCs of per-node *reduced-dimension* covariance matrices, we show that a concatenation of part of these matrices converges to the signal subspace that can be obtained from the inversion of the matrix containing all *network-wide* GEVCs. This method is referred to as the *inversion-based* method and will be explained in Section 2.4.2.

It will be shown via numerical simulations that the *inversion-based* method delivers a faster convergence speed, compared to the *correlation-based* method. This improvement is achieved at the cost of more complex computations.

The chapter is organized as follows. The data model and the problem statement are presented in Section 2.2. Section 2.3 explains how the network-wide signal subspace can be estimated by means of a centralized GEVD. Section 2.4 presents the correlation-based and the inversion-based methods to compute the network-wide signal subspace in a distributed fashion and without computing the centralized GEVD. Section 2.5 evaluates and compares the proposed distributed algorithms via numerical simulation results. Finally, conclusions are drawn in Section 2.6.

2.2 Data model and problem statement

We consider a fully-connected WSN with K multi-sensor nodes in which each node $k \in \mathcal{K} = \{1, \ldots, K\}$ collects observations of a complex-valued M_k-channel sensor signal $\mathbf{y}_k$. Note that this also allows for a hierarchical WSN where K

master nodes receive sensor observations from M_k slave nodes with a single sensor. The topology of the network is assumed to be fully-connected which means that data broadcast by a node can be received by all other $K-1$ nodes in the network. The sensor signal $\mathbf{y}_k$ consists of a mixture of S target source signals and additive noise, which may be spatially correlated between nodes. Hence $\mathbf{y}_k$ can be modeled as

$$\mathbf{y}_k = \mathbf{A}_k\mathbf{s} + \mathbf{n}_k \tag{2.1}$$

where $\mathbf{s}$ is an S-channel signal containing S target source signals, $\mathbf{A}_k = [\mathbf{a}_{k1} \cdots \mathbf{a}_{kS}]$ is a static (or slowly varying) $M_k \times S$ steering matrix with $\mathbf{a}_{ks}$ $(r = 1, \cdots, S)$ the so-called steering vector from target source s to the sensors of node k, and $\mathbf{n}_k$ is the additive noise which can be spatially correlated between nodes. It should be mentioned that since all signals in (2.1) are assumed to be complex-valued, the model also includes, e.g., convolutive time-domain mixtures, described as instantaneous per-frequency mixtures in the (short-term) Fourier transform domain[2].

By stacking all $\mathbf{y}_k$'s and $\mathbf{n}_k$'s, we obtain the network-wide M-channel ($M = \sum_{k=1}^{K} M_k$) signals $\mathbf{y}$ and $\mathbf{n}$, respectively. Likewise, we define the $M \times S$ matrix $\mathbf{A} = [\mathbf{a}_1 \cdots \mathbf{a}_S]$ as the stacked version of all $\mathbf{A}_k$'s such that

$$\mathbf{y} = \mathbf{A}\mathbf{s} + \mathbf{n}. \tag{2.2}$$

In this chapter we consider the problem of estimating an S-dimensional basis of the network-wide signal subspace, i.e., the *range* or the *column space* of the network-wide steering matrix $\mathbf{A}$. We aim to do this in a distributed fashion, i.e., without explicitly constructing network-wide covariance matrices, as this would require centralization of all the sensor signal observations. Instead the nodes will only exchange S-channel sensor signal observations, which results in a compression factor of M_k/S at node k when compared to the centralized case (assuming $M_k \geq S$). Here we assume that S is either known or estimated a-priori (e.g., as in [4, 121]). It is noted that, if $S = 1$, the problem reduces to an steering vector estimation problem, where we estimate $\mathbf{a}_1$ up to a scaling ambiguity.

[2]Note that in applications such as audio signal processing where the microphone signals are often represented as instantaneous per-frequency mixtures in the short-time Fourier transform (STFT) domain (associated with rather long time-domain acoustic impulse responses), the data model (2.1) is only approximately satisfied. In these cases, shorter window lengths introduce more model imperfections [10].

2.3 Centralized GEVD-based signal subspace estimation

In this section we first explain how the network-wide signal subspace can be estimated by means of a centralized GEVD, i.e., in the case where all the sensor observations are collected in a fusion center. Without loss of generality (w.l.o.g.), we assume that $\mathbf{y}$ is zero-mean which possibly requires a mean subtraction preprocessing step. The network-wide sensor signal correlation matrix is then defined as

$$\mathbf{R}_{yy} = E\{\mathbf{y}\mathbf{y}^H\} \tag{2.3}$$

where $E\{\cdot\}$ denotes the expected value operator, and the superscript H denotes the conjugate transpose operator. The exact sensor signal covariance matrix as defined in (2.3) is often not available in practice, but can be estimated via sample averaging. To this end, we define the $M \times N$ observation matrix $\mathbf{Y}$, where each column corresponds to an observation of $\mathbf{y}$ at a certain time instant, such that $\mathbf{R}_{yy}$ can be approximated as

$$\mathbf{R}_{yy} \approx \frac{1}{N}\mathbf{Y}\mathbf{Y}^H \tag{2.4}$$

and when having an infinitely long observation window we can write $\mathbf{R}_{yy} = \lim_{N\to\infty} \frac{1}{N}\mathbf{Y}\mathbf{Y}^H$.

We also define the network-wide sensor noise covariance matrix $\mathbf{R}_{nn} = E\{\mathbf{n}\mathbf{n}^H\}$ where it is assumed that $\mathbf{R}_{nn}$ is either known a-priori or can be estimated from noise-only segments in the sensor observations (similar to (2.4)). The latter can be performed in applications such as speech enhancement where $\mathbf{R}_{yy}$ and $\mathbf{R}_{nn}$ can be estimated during "speech-and-noise" and "noise-only" segments, respectively, which can be distinguished by means of a voice activity detection (VAD) mechanism [112].

A GEVD of the ordered matrix pair $(\mathbf{R}_{yy}, \mathbf{R}_{nn})$ is defined as [122]

$$\mathbf{R}_{yy}\hat{\mathbf{X}} = \mathbf{R}_{nn}\hat{\mathbf{X}}\hat{\mathbf{\Lambda}} \tag{2.5}$$

where $\hat{\mathbf{X}} = [\hat{\mathbf{x}}_1 \dots \hat{\mathbf{x}}_M]$, with $\hat{\mathbf{x}}_m$ the m-th GEVC, and $\hat{\mathbf{\Lambda}} = diag\{\hat{\lambda}_1 \dots \hat{\lambda}_M\}$, with $\hat{\lambda}_m$ the m-th largest generalized eigenvalue (GEVL), and where the hat $(\hat{\cdot})$ refers to the fact that the centralized estimation is considered. Note that when $\mathbf{R}_{nn}$ is invertible, (2.5) can be written as a non-symmetric EVD such that

$$\mathbf{R}_{nn}^{-1}\mathbf{R}_{yy} = \hat{\mathbf{X}}\hat{\mathbf{\Lambda}}\hat{\mathbf{X}}^{-1}. \tag{2.6}$$

In general, the corresponding joint diagonalization of $\mathbf{R}_{yy}$ and $\mathbf{R}_{nn}$ can be written as $\mathbf{R}_{yy} = \hat{\mathbf{Q}}\hat{\mathbf{\Sigma}}\hat{\mathbf{Q}}^H$ and $\mathbf{R}_{nn} = \hat{\mathbf{Q}}\hat{\mathbf{\Gamma}}\hat{\mathbf{Q}}^H$, where $\hat{\mathbf{\Sigma}}$ and $\hat{\mathbf{\Gamma}}$ are diagonal

matrices. With this and using (2.6), it follows that $\hat{\mathbf{Q}} = \hat{\mathbf{X}}^{-H}$, with $\hat{\mathbf{Q}}$ a full-rank $M \times M$ matrix (not necessarily orthogonal). It can be then verified that $\hat{\mathbf{\Sigma}} = \hat{\mathbf{X}}^H \mathbf{R}_{yy} \hat{\mathbf{X}}$ and $\hat{\mathbf{\Gamma}} = \hat{\mathbf{X}}^H \mathbf{R}_{nn} \hat{\mathbf{X}}$ and that $\hat{\mathbf{\Lambda}} = \hat{\mathbf{\Sigma}}(\hat{\mathbf{\Gamma}})^{-1}$. Since the GEVCs are defined up to a scaling, here we assume without loss of generality (w.l.o.g.) that all $\hat{\mathbf{x}}_m$'s are scaled such that $\hat{\mathbf{X}}^H \mathbf{R}_{nn} \hat{\mathbf{X}} = \mathbf{I}_M$. With this we have $\hat{\mathbf{\Gamma}} = \mathbf{I}_M$ and $\hat{\mathbf{\Sigma}} = \hat{\mathbf{\Lambda}}$, and hence the joint diagonalization becomes

$$\mathbf{R}_{yy} = \hat{\mathbf{Q}} \hat{\mathbf{\Lambda}} \hat{\mathbf{Q}}^H, \ \mathbf{R}_{nn} = \hat{\mathbf{Q}} \hat{\mathbf{Q}}^H. \tag{2.7}$$

Defining $\mathbf{\Pi} = \text{diag}\{P_1, ..., P_S\}$ with P_s the power of target source signal s, equation (2.2) implies that $\mathbf{A\Pi A}^H = \mathbf{R}_{yy} - \mathbf{R}_{nn}$, where further incorporating (2.7) yields

$$\mathbf{A\Pi A}^H = \hat{\mathbf{Q}} \big(\hat{\mathbf{\Lambda}} - \mathbf{I}_M \big) \hat{\mathbf{Q}}^H. \tag{2.8}$$

Note that $\hat{\mathbf{Q}}$ is full rank, and that the left-hand side of (2.8) consists of a positive semi-definite matrix with rank S. This requires that $\hat{\mathbf{\Lambda}} - \mathbf{I}_M$ contains only S non-zero diagonal entries. Therefore, the first S GEVLs are larger than one, and all others are equal to one. The first S columns of $\hat{\mathbf{Q}}$ must then span the same S-dimensional subspace as the columns of $\mathbf{A}$ and hence fully define the network-wide signal subspace.

We define $\hat{\mathbb{X}}$ and $\hat{\mathbb{Q}}$ as an $M \times S$ matrix containing the first S columns of $\hat{\mathbf{X}}$ and $\hat{\mathbf{Q}}$, respectively. Hence $\hat{\mathbb{Q}}$ spans the network-wide signal subspace. We further define the partitioning

$$\hat{\mathbb{X}} \triangleq \begin{bmatrix} \hat{\mathbb{X}}_1 \\ \vdots \\ \hat{\mathbb{X}}_K \end{bmatrix}, \ \hat{\mathbb{Q}} \triangleq \begin{bmatrix} \hat{\mathbb{Q}}_1 \\ \vdots \\ \hat{\mathbb{Q}}_K \end{bmatrix} \tag{2.9}$$

where $(.)_k$ is the $M_k \times S$ submatrix that corresponds to node k. Moreover we define the $S \times S$ diagonal matrix $\hat{\mathbb{\Lambda}} = \text{diag}\{\hat{\lambda}_1 \dots \hat{\lambda}_S\}$.

It is reiterated that the GEVD-based signal subspace estimation allows to directly incorporate the (estimated) noise covariance matrix, which is not the case in an EVD-based approach. In addition, the GEVD is also immune to unknown per-channel scalings (e.g., due to lack of sensor calibration between nodes), which is explained as follows. Applying a random scaling to (some) channels of $\mathbf{y}_k$ at node k results in a similar scaling of the corresponding rows and columns of the network-wide sensor signal correlation matrix $\mathbf{R}_{yy}$. This scaling has an influence on the entire eigenstructure of $\mathbf{R}_{yy}$, i.e., all coefficients of all its eigenvectors are affected. This is an undesirable effect if the eigenvectors are used to estimate the signal subspace or steering vector. Indeed, a simple scaling of the channels in one node should not affect the signal subspace or SV estimate in other nodes. However, it can be shown that the GEVD of

$(\mathbf{R}_{yy}, \mathbf{R}_{nn})$ does not have this effect, i.e., the same scaling will only affect the coefficients in the GEVCs corresponding to the scaled channels at the node k, i.e., the scaling remains localized and does not spread out to other GEVC coefficients. As a result, the signal subspace or SV estimate at other nodes will not be affected.

2.4 Distributed GEVD-based signal subspace estimation

So far we have considered the GEVD-based estimation of $\hat{\mathbb{Q}}$ via data centralization. However in a typical WSN, a node k only observes its own M_k-channel sensor signal $\mathbf{y}_k$ and hence can only estimate an $M_k \times M_k$ submatrix of $\mathbf{R}_{yy}$ and $\mathbf{R}_{nn}$. In this section we explain how the nodes can efficiently cooperate to estimate the columns of the network-wide $\hat{\mathbb{Q}}$, without constructing the network-wide correlation matrices $\mathbf{R}_{yy}$ and $\mathbf{R}_{nn}$. The algorithm derivation starts from a distributed GEVD algorithm, referred to as the distributed adaptive covariance-matrix generalized eigenvector estimation (DACGEE) algorithm [1]. We will then show that the network-wide $\hat{\mathbb{Q}}$ can be inferred from the inversion of compressed GEVC matrices that are computed at the individual nodes in the DACGEE algorithm. This is remarkable, since DACGEE only computes $\hat{\mathbb{X}}$, whereas the inversion of the full matrix $\hat{\mathbf{X}}$ is at first sight required to find $\hat{\mathbb{Q}}$.

2.4.1 The fully-connected DACGEE algorithm

The DACGEE algorithm [1] estimates and updates (in a distributed and block-adaptive fashion) the matrix $\hat{\mathbb{X}}$, where the communication cost of node k is reduced by a factor M_k/S (compared to the centralized case and assuming $M_k \geq S$). In DACGEE, each node k only updates the $M_k \times S$ submatrix $\mathbb{X}_k^i$, which is the estimate of $\hat{\mathbb{X}}_k$ at iteration i. We define the $M \times S$ matrix $\mathbb{X}^i$ as the estimate of $\hat{\mathbb{X}}$, which is constructed by concatenating all submatrices $\mathbb{X}_k^i, \forall k \in \mathcal{K}$, i.e., $\mathbb{X}^i \triangleq [\mathbb{X}_1^{iT} \dots \mathbb{X}_K^{iT}]^T$. Hence the objective of DACGEE is to obtain $\lim_{i\to\infty} \mathbb{X}^i = \hat{\mathbb{X}}$.

Each node k, at iteration i, uses $\mathbb{X}_k^i$ to compress N observations of its M_k-channel sensor signal into observations of the S-channel signal

$$\mathbf{z}_k^i = \mathbb{X}_k^{iH} \mathbf{y}_k \tag{2.10}$$

and then broadcasts these N observations of $\mathbf{z}_k^i$ to all other nodes. Therefore an updating node q observes the following P_q-channel sensor signal ($P_q \triangleq$

$M_q + S(K-1)$) and estimates the corresponding covariance matrix:

$$\widetilde{\mathbf{y}}_q^i = \begin{bmatrix} \mathbf{y}_q \\ \mathbf{z}_{-q}^i \end{bmatrix} \Rightarrow \mathbf{R}_{\tilde{y}_q \tilde{y}_q}^i = E\{\widetilde{\mathbf{y}}_q^i \widetilde{\mathbf{y}}_q^{i\,H}\} \tag{2.11}$$

where $\mathbf{z}_{-q}^i = [\mathbf{z}_1^{i\,T} \dots \mathbf{z}_{(q-1)}^{i\,T} \mathbf{z}_{(q+1)}^{i\,T} \dots \mathbf{z}_K^{i\,T}]^T$. In a similar way, we can define $\mathbf{R}_{\tilde{n}_q \tilde{n}_q}^i$, which can be estimated from $\widetilde{\mathbf{y}}_q^i$ during 'noise-only' segments[3]. The updating node q can then compute the GEVD of the reduced-dimension GEVD of $(\mathbf{R}_{\tilde{y}_q \tilde{y}_q}^i, \mathbf{R}_{\tilde{n}_q \tilde{n}_q}^i)$ as (compare with (2.5))

$$\mathbf{R}_{\tilde{y}_q \tilde{y}_q}^i \widetilde{\mathbf{X}}_q^i = \mathbf{R}_{\tilde{n}_q \tilde{n}_q}^i \widetilde{\mathbf{X}}_q^i \widetilde{\mathbf{\Lambda}}_q^i \;\text{ s.t. }\; \widetilde{\mathbf{X}}_q^{i\,H} \mathbf{R}_{\tilde{n}_q \tilde{n}_q}^i \widetilde{\mathbf{X}}_q^i = \mathbf{I}_{P_q} \tag{2.12}$$

where the $P_q \times P_q$ matrix $\widetilde{\mathbf{X}}_q^i$ and the $P_q \times P_q$ diagonal matrix $\widetilde{\mathbf{\Lambda}}_q^i$ contain all P_q local GEVCs and GEVLs of $(\mathbf{R}_{\tilde{y}_q \tilde{y}_q}^i, \mathbf{R}_{\tilde{n}_q \tilde{n}_q}^i)$, respectively. We now define the $P_q \times S$ matrix $\widetilde{\mathbb{X}}_q^i$ containing the first S columns of $\widetilde{\mathbf{X}}_q^i$. Moreover we define the $S \times S$ diagonal matrix $\widetilde{\mathbb{\Lambda}}_q^i$ as the $S \times S$ submatrix of $\widetilde{\mathbf{\Lambda}}_q^i$ containing its first S diagonal entries. In DACGEE, node q then uses the first M_q rows of $\widetilde{\mathbb{X}}_q^i$ as $\mathbb{X}_q^{i+1}$. In the next iteration, the updating node q is changed, and a new block of N sensor signal observations is used. Note that the latter means that the iterations are spread out over different signal segments in a block-recursive fashion.
Using the above updating procedure, it can be shown that (up to estimation errors in the covariance matrices) [1, 123].

$$\lim_{i \to \infty} \mathbb{X}^i = \hat{\mathbb{X}} \;\text{ and }\; \lim_{i \to \infty} \widetilde{\mathbb{\Lambda}}_k^i = \hat{\mathbb{\Lambda}}, \forall k \in \mathcal{K} \tag{2.13}$$

2.4.2 Extracting the signal subspace

Similar to $\mathbb{X}^i$ we define $\mathbb{Q}^i \triangleq [\mathbb{Q}_1^{i\,T} \dots \mathbb{Q}_K^{i\,T}]^T$ as the estimation of $\hat{\mathbb{Q}}$ at iteration i. Then the objective is to estimate $\mathbb{Q}^i$ in a distributed fashion such that it converges to the network-wide $\hat{\mathbb{Q}}$. We first briefly review the correlation-based method presented in [119].

Correlation-based method

Let $\hat{\mathbf{z}} = \hat{\mathbb{X}}^H \mathbf{y} = \sum_{k \in \mathcal{K}} \hat{\mathbb{X}}_k^H \mathbf{y}_k$. Note that, due to the convergence of the DACGEE algorithm, $\hat{\mathbf{z}}$ will be equal to $\hat{\mathbf{z}} = \sum_{k \in \mathcal{K}} \mathbf{z}_k^i$ for $i \to \infty$ (see (2.10)).

[3] If the network-wide $\mathbf{R}_{nn}$ is known a-priori, one can compute $\mathbf{R}_{\tilde{n}_q \tilde{n}_q}^i$ directly by means of the compression matrices from the other nodes, as in (2.33).

When considering the covariance between $\mathbf{y}$ and $\hat{\mathbf{z}}$, we have that

$$\hat{\mathbf{R}}_{y\hat{z}} = E\{\mathbf{y}\hat{\mathbf{z}}^H\} = E\{\mathbf{y}\mathbf{y}^H\hat{\mathbb{X}}\} = \mathbf{R}_{yy}\hat{\mathbb{X}}\,. \tag{2.14}$$

Using this with (2.7) and the fact that $\hat{\mathbf{Q}} = \hat{\mathbf{X}}^{-H}$, we find

$$\hat{\mathbf{R}}_{y\hat{z}} = \hat{\mathbf{Q}}\hat{\mathbf{\Lambda}}\hat{\mathbf{Q}}^H\hat{\mathbf{Q}}^{-H}\mathbf{E}_S = \hat{\mathbb{Q}}\hat{\mathbb{\Lambda}} \tag{2.15}$$

where $\mathbf{E}_S$ is the column selector matrix which selects the first S columns. Since $\hat{\mathbb{\Lambda}}$ only scales the columns of $\hat{\mathbb{Q}}$, $\hat{\mathbf{R}}_{y\hat{z}}$ defines the network-wide signal subspace. With $\mathbf{z}^i = \sum_{k\in\mathcal{K}} \mathbf{z}_k^i$, each node in iteration i computes its part of $\hat{\mathbf{R}}_{y\hat{z}}$, defined as $\mathbf{R}^i_{y_k z} = E\{\mathbf{y}_k \mathbf{z}^{i\,H}\}$, as follows:

$$\mathbf{R}^i_{y_k z} = E\{\mathbf{y}_k \mathbf{z}^{i\,H}\} \approx \frac{1}{N}\mathbf{Y}_k\mathbf{Z}^{i\,H} \tag{2.16}$$

where $\mathbf{Y}_k$ and $\mathbf{Z}^i$ are $M_k \times N$ and $S \times N$ matrices containing N observations of $\mathbf{y}_k$ and $\mathbf{z}^i$ in their columns, respectively.

The stacked matrix of all $\mathbf{R}^i_{y_k z}$'s, i.e., $\mathbf{R}^i_{yz}$, is shown to converge to $\hat{\mathbf{R}}_{y\hat{z}}$, i.e.,

$$\lim_{i\to\infty} \mathbf{R}^i_{yz} = \hat{\mathbf{R}}_{y\hat{z}}. \tag{2.17}$$

The resulting algorithm is described in Table 2.1.

Inversion-based method

In this section we propose a new method to estimate the network-wide signal subspace $\hat{\mathbb{Q}}$, based on the inversion of the matrix $\widetilde{\mathbf{X}}^i_k$ at each node k. $\widetilde{\mathbf{X}}^i_k$ is computed by the DACGEE algorithm, although DACGEE for its own convergence only needs to compute the matrix $\widetilde{\mathbb{X}}^i_q$ at the updating node q. Therefore, similar to (2.7), we can write

$$\mathbf{R}^i_{\tilde{y}_k\tilde{y}_k} = \widetilde{\mathbf{Q}}^i_k\widetilde{\mathbf{\Lambda}}^i_k\widetilde{\mathbf{Q}}^{i\,H}_k,\ \mathbf{R}^i_{\tilde{n}_k\tilde{n}_k} = \widetilde{\mathbf{Q}}^i_k\widetilde{\mathbf{Q}}^{i\,H}_k \tag{2.21}$$

where $\widetilde{\mathbf{Q}}^i_k = (\widetilde{\mathbf{X}}^i_k)^{-H}$. Node k then extracts $\mathbb{Q}^i_k$ as

$$\mathbb{Q}^i_k = [\mathbf{I}_{M_k}\ \mathbf{0}]\,\widetilde{\mathbf{Q}}^i_k \begin{bmatrix} \mathbf{I}_S \\ \mathbf{0} \end{bmatrix}. \tag{2.22}$$

The stacked matrix $\mathbb{Q}^i$ will be shown below to converge to $\hat{\mathbb{Q}}$, which is remarkable since $\hat{\mathbb{Q}}$ is part of the inverse of the full matrix $\hat{\mathbf{X}}$, whereas the $\mathbb{Q}^i_k$'s are extracted from the inverses of the compressed matrices $\widetilde{\mathbf{X}}^i_k$. The resulting distributed

Table 2.1: Distributed signal subspace estimation based on the local correlation-based method in a fully-connected WSN

1. Set $i \leftarrow 0$, $q \leftarrow 1$, and initialize all $\mathbb{X}_k^0$, $\forall\, k \in \mathcal{K}$, with random entries.
2. Each node $k \in \mathcal{K}\backslash q$ broadcasts N new compressed observations of
$$\mathbf{z}_k^i[j] = \mathbb{X}_k^{i\,H}\mathbf{y}_k[iN+j], \quad j = 1 \dots N \tag{2.18}$$
where [.] denotes a sample index and where N is sufficiently large such that it includes both 'signal+noise' and 'noise-only' samples.
3. At node q:
   (a) Estimate $\mathbf{R}^i_{\tilde{y}_q\tilde{y}_q}$ and $\mathbf{R}^i_{\tilde{n}_q\tilde{n}_q}$ via sample averaging.
   (b) Compute the columns of $\widetilde{\mathbb{X}}_q^{i+1}$ as the first S principal GEVCs of $(\mathbf{R}^i_{\tilde{y}_q\tilde{y}_q}, \mathbf{R}^i_{\tilde{n}_q\tilde{n}_q})$.
   (c) Partition $\widetilde{\mathbb{X}}_q^{i+1}$ as
$$\mathbb{X}_q^{i+1} = \left[\mathbf{I}_{M_k}\ \mathbf{0}\right]\widetilde{\mathbb{X}}_q^{i+1} \tag{2.19}$$
$$\mathbf{G}_{-q} = \left[\mathbf{0}\ \mathbf{I}_{S(K-1)}\right]\widetilde{\mathbb{X}}_q^{i+1}. \tag{2.20}$$
   (d) Broadcast $\mathbf{G}_{-q} \triangleq \left[\mathbf{G}_1^T \dots \mathbf{G}_{q-1}^T\, \mathbf{G}_{q+1}^T \dots \mathbf{G}_K^T\right]^T$ and $\mathbf{z}_q^i[j] = \mathbb{X}_q^{i+1\,H}\mathbf{y}_q[iN+j]$ to all other nodes.
4. Each node $k \in \mathcal{K}\backslash\{q\}$ updates its compressor as $\mathbb{X}_k^{i+1} = \mathbb{X}_k^i\mathbf{G}_k$.
5. Each node $k \in \mathcal{K}$ computes $\mathbf{z}^i = \sum_{k\in\mathcal{K}\backslash\{q\}}\mathbf{z}_k^i + \mathbf{z}_q^i$ locally and updates $\hat{\mathbf{R}}^i_{y_k z}$ as in (2.16).
6. $i \leftarrow i+1$ and $q \leftarrow (q \mod K)+1$ and return to step 2.

algorithm is provided in Table 2.2. It will be shown in Section **??** that the new method significantly outperforms the correlation-based method in terms of convergence speed. This improvement is indeed achieved at the cost of more complex computations, since each node $k \in \mathcal{K}$ at iteration i must compute all the P_k GEVCs in $\widetilde{\mathbf{X}}_k^i$ from the GEVD of the reduced-dimension correlation matrices $(\mathbf{R}^i_{\tilde{y}_k\tilde{y}_k}, \mathbf{R}^i_{\tilde{n}_k\tilde{n}_k})$ as well as a matrix inverse.

Theorem I: *In the algorithm defined in Table 2.2, the network-wide signal subspace* $\mathbb{Q}^i$ *converges to the centralized network-wide signal subspace* $\hat{\mathbb{Q}}$*, i.e.,* $\lim_{i\to\infty}\mathbb{Q}^i = \hat{\mathbb{Q}}$ *or in particular,* $\lim_{i\to\infty}\mathbb{Q}_k^i = \hat{\mathbb{Q}}_k, \forall k \in \mathcal{K}$.

Table 2.2: Distributed signal subspace estimation based on the local inversion-based method in a fully-connected WSN

1. Set $i \leftarrow 0$, $q \leftarrow 1$, and initialize all $\mathbb{X}_k^0$, $\forall\, k \in \mathcal{K}$, with random entries.
2. Each node $k \in \mathcal{K}\backslash q$ broadcasts N new compressed observations of
$$\mathbf{z}_k^i[j] = \mathbb{X}_k^{i\,H}\mathbf{y}_k[iN+j], \quad j = 1 \dots N \tag{2.23}$$
where [.] denotes a sample index and where N is sufficiently large such that it includes both 'signal+noise' and 'noise-only' samples.
3. At node q:
   (a) Estimate $\mathbf{R}^i_{\tilde{y}_q\tilde{y}_q}$ and $\mathbf{R}^i_{\tilde{n}_q\tilde{n}_q}$ via sample averaging.
   (b) Compute local GEVCs $\widetilde{\mathbf{X}}_q^{i+1}$ from the GEVD of $(\mathbf{R}^i_{\tilde{y}_q\tilde{y}_q}, \mathbf{R}^i_{\tilde{n}_q\tilde{n}_q})$ and then $\widetilde{\mathbf{Q}}_q^{i+1} = \left(\widetilde{\mathbf{X}}_q^{i+1}\right)^{-H}$.
   (c) Partition $\widetilde{\mathbf{Q}}_q^{i+1}$ as
$$\mathbb{Q}_q^{i+1} = \left[\mathbf{I}_{M_k}\ \mathbf{0}\right] \widetilde{\mathbf{Q}}_q^{i+1} \begin{bmatrix} \mathbf{I}_R \\ \mathbf{0} \end{bmatrix}. \tag{2.24}$$
   (d) Partition $\widetilde{\mathbf{X}}_q^{i+1}$ as
$$\mathbb{X}_q^{i+1} = \left[\mathbf{I}_{M_k}\ \mathbf{0}\right] \widetilde{\mathbf{X}}_q^{i+1} \begin{bmatrix} \mathbf{I}_S \\ \mathbf{0} \end{bmatrix} \tag{2.25}$$
$$\mathbf{G}_{-q} = \left[\mathbf{0}\ \mathbf{I}_{S(K-1)}\right] \widetilde{\mathbf{X}}_q^{i+1} \begin{bmatrix} \mathbf{I}_S \\ \mathbf{0} \end{bmatrix}. \tag{2.26}$$
   (e) Broadcast $\mathbf{G}_{-q} \triangleq \left[\mathbf{G}_1^T \dots \mathbf{G}_{q-1}^T\, \mathbf{G}_{q+1}^T \dots \mathbf{G}_K^T\right]^T$ and $\mathbf{z}_q^i[j] = \mathbb{X}_q^{i+1\,H}\mathbf{y}_q[iN+j]$ to all other nodes.
4. Each node $k \in \mathcal{K}\backslash\{q\}$ updates its compressor as $\mathbb{X}_k^{i+1} = \mathbb{X}_k^i\mathbf{G}_k$.
5. Each node $k \in \mathcal{K}\backslash\{q\}$ updates its $\mathbb{Q}_k^{i+1}$ similar to steps 3a-3c.
6. $i \leftarrow i+1$ and $q \leftarrow (q \mod K)+1$ and return to step 2.

Proof: Replacing $\mathbf{R}^i_{\tilde{y}_k\tilde{y}_k}$ in (2.12) by (2.21), and considering the fact that $\widetilde{\mathbf{Q}}_k^{i\,H}\widetilde{\mathbf{X}}_k^i = \mathbf{I}_{P_k}$, equation (2.12) can be rewritten as

$$\widetilde{\mathbf{Q}}_k^i\widetilde{\mathbf{\Lambda}}_k^i = \mathbf{R}^i_{\tilde{n}_k\tilde{n}_k}\widetilde{\mathbf{X}}_k^i\widetilde{\mathbf{\Lambda}}_k^i \tag{2.27}$$

where taking the first S columns yields

$$\widetilde{\mathbb{Q}}_k^i = \mathbf{R}^i_{\tilde{n}_k\tilde{n}_k}\widetilde{\mathbb{X}}_k^i. \tag{2.28}$$

We define the $M \times P_k$ compressor matrix $\mathbf{C}_k^i$ as

$$\mathbf{C}_k^i = \left[\begin{array}{c|c|c} \mathbf{0} & \mathbf{B}_{<k}^i & \mathbf{0} \\ \hline \mathbf{I}_{M_k} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{B}_{>k}^i \end{array}\right] \tag{2.29}$$

where $\mathbf{0}$ is an all-zero matrix of proper dimension, and where[4]

$$\mathbf{B}_{<k}^i = \text{Blkdiag}(\mathbb{X}_1^i, \ldots, \mathbb{X}_{(k-1)}^i) \tag{2.30}$$

$$\mathbf{B}_{>k}^i = \text{Blkdiag}(\mathbb{X}_{(k+1)}^i, \ldots, \mathbb{X}_K^i). \tag{2.31}$$

With this, and using (2.10)-(2.11), we can link the reduced P_k-dimensional correlation matrices $(\mathbf{R}_{\tilde{y}_k\tilde{y}_k}, \mathbf{R}_{\tilde{n}_k\tilde{n}_k})$ with the full M-dimensional correlation matrices $(\mathbf{R}_{yy}, \mathbf{R}_{nn})$, i.e.,

$$\mathbf{R}_{\tilde{y}_k\tilde{y}_k}^i = \mathbf{C}_k^{i\,H}\mathbf{R}_{yy}\mathbf{C}_k^i \tag{2.32}$$

$$\mathbf{R}_{\tilde{n}_k\tilde{n}_k}^i = \mathbf{C}_k^{i\,H}\mathbf{R}_{nn}\mathbf{C}_k^i. \tag{2.33}$$

We now assume that $i \to \infty$, i.e., when DACGEE has converged and hence at all nodes $k \in \mathcal{K}$ we have that $\mathbb{X}_k^i = \hat{\mathbb{X}}_k$ and that $\widetilde{\mathbb{\Lambda}}_k^i = \hat{\mathbb{\Lambda}}$ (see (2.13)). Moreover since after the convergence at all nodes $k \in \mathcal{K}$ we have that $\mathbb{X}_k^{i+1} = \mathbb{X}_k^i$, this requires that [123]

$$\widetilde{\mathbb{X}}_k^{i+1} = [\mathbb{X}_k^{i\,T}\,\mathbf{I}_S \ldots \mathbf{I}_S]^T. \tag{2.34}$$

Let $\hat{\mathbf{C}}_k$ be the compressor matrix $\mathbf{C}_k^i$ after the convergence of the DACGEE algorithm, i.e., when in (2.29)-(2.31) $\mathbb{X}_k^i = \hat{\mathbb{X}}_k, \forall k \in \mathcal{K}$. Considering this together with (2.33), we can rewrite (2.28) as

$$\lim_{i\to\infty} \widetilde{\mathbb{Q}}_k^i = \lim_{i\to\infty} \hat{\mathbf{C}}_k^H \mathbf{R}_{nn} \hat{\mathbf{C}}_k \widetilde{\mathbb{X}}_k^i \tag{2.35}$$

Based on (2.34), after convergence we have that $\hat{\mathbb{X}} = \hat{\mathbf{C}}_k \widetilde{\mathbb{X}}_k^i$. Now plugging (2.7) into (2.35), and considering the fact that $\hat{\mathbb{Q}}^H \hat{\mathbb{X}} = \mathbf{I}_S$, it follows that

$$\lim_{i\to\infty} \widetilde{\mathbb{Q}}_k^i = \lim_{i\to\infty} \hat{\mathbf{C}}_k^H \hat{\mathbf{Q}} \hat{\mathbf{Q}}^H \hat{\mathbb{X}} \tag{2.36}$$

$$= \lim_{i\to\infty} \hat{\mathbf{C}}_k^H \hat{\mathbf{Q}} \begin{bmatrix} \mathbf{I}_S \\ \mathbf{0} \end{bmatrix} \tag{2.37}$$

$$= \lim_{i\to\infty} \hat{\mathbf{C}}_k^H \hat{\mathbb{Q}}. \tag{2.38}$$

[4]It is noted that the diagonal blocks are not square here, i.e., in this case Blkdiag(.) is not truly a block-diagonal matrix in the strict sense.

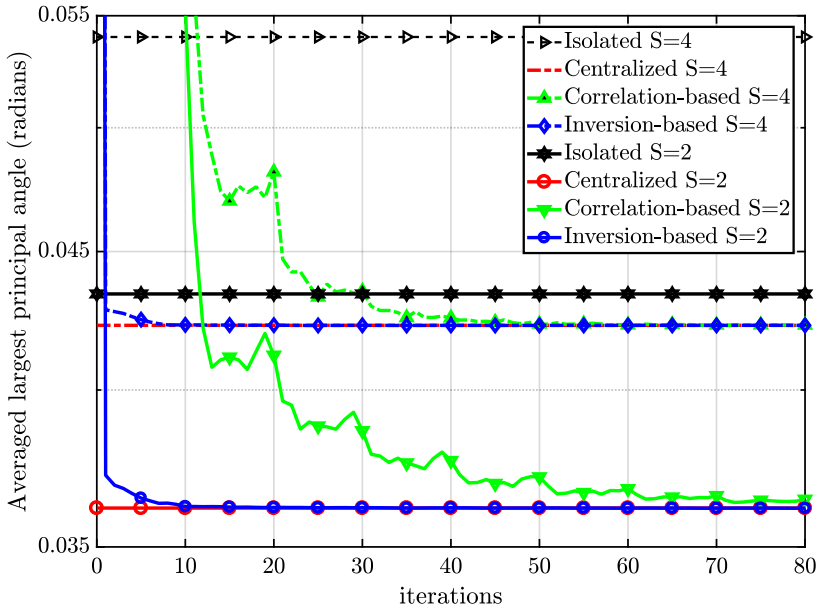


Figure 2.1: Distributed signal subspace estimation for S target sources.

Selecting the first M_k rows of (2.37), we obtain

$$\lim_{i\to\infty} \mathbb{Q}_k^i = \hat{\mathbb{Q}}_k \tag{2.39}$$

which proves the theorem.

2.5 Simulation results

In this section, we compare the performance of the proposed inversion-based method [120] with the correlation-based method [119] via numerical Monte-Carlo (MC) simulations. The simulations compare the two methods with the 'centralized' and the 'isolated' cases. The latter corresponds to the case where each node only observes its own M_k-channel sensor signal and hence does not cooperate with other nodes. In all MC runs, $K = 10$ and $M_k = 15, \forall k \in \mathcal{K}$. Out of 10 localized sources in each MC scenario, S of them are considered as the target sources (with an on-off behavior) and the remaining $(10 - S)$ sources are treated as noise sources (continuously active). The network-wide noise signal $\mathbf{n}$ (see (2.2)) can be described as $\mathbf{n} = \mathbf{J}\mathbf{t} + \mathbf{v}$ where $\mathbf{J}$ is the steering matrix corresponding to the noise sources, $\mathbf{t}$ contains the $(10 - S)$ noise source signals,

and $\mathbf{v}$ models the spatially uncorrelated noise signals. The network-wide steering matrices $\mathbf{A}$ and $\mathbf{J}$ are static matrices with dimensions $150 \times S$ and $150 \times (10-S)$, respectively, in which the entries are drawn from a uniform distribution over the interval $[-0.5; 0.5]$. $\mathbf{s}$ and $\mathbf{t}$ are S-channel and $(10-S)$-channel stochastic source signals from which the observations are independently drawn from a uniform distribution over the interval $[-0.5; 0.5]$. Moreover, $\mathbf{v}$ is a 150-channel stochastic signal from which the observations are independently drawn from a uniform distribution over the interval $[-\sqrt{0.1}/2; \sqrt{0.1}/2]$. In each MC run, a different simulation scenario is created and an average of the largest canonical angle (principal angle) between the true steering matrix $\mathbf{A}_k$ and its corresponding signal subspace estimate over all nodes is considered as a performance measure.

Fig. 2.1 illustrates the results for the cases where $S = 2$ and $S = 4$, averaged over 200 MC runs. This figure clearly shows that: 1) The cooperative estimation (either centralized or distributed) significantly outperforms the one achieved by the isolated estimation in terms of the estimation accuracy; 2) The both estimates obtained with the proposed distributed algorithms converge to the estimate obtained with the centralized estimation; 3) The distributed signal subspace estimation obtained with the inversion-based method presented in [120] converges significantly faster than the correlation-based method presented in [119]; 4) While the value of S remarkably affects the convergence speed of the correlation-based method, it has almost no effect on the convergence speed of the the inversion-based method.

2.6 Conclusion

In this Chapter, we have presented two different distributed algorithms for network-wide signal subspace estimation in a fully-connected WSN. We have applied a GEVD-based method, which allows to better estimate a target signal subspace in spatially correlated noise. The correlation-based method only computes the S local principal GEVCs from which the signal subspace is extracted, while the inversion-based method first computes all local GEVCs at each node, and then inverts this matrix to obtain a basis of the corresponding part of the network-wide signal subspace. It has been demonstrated that the network-wide signal subspace estimate obtained with the inversion-based method converges to the estimate obtained with the centralized network-wide signal subspace, with a faster convergence speed compared to the correlation-based method. This convergence speed improvement is indeed achieved at the cost of more complex computations, since each node in iteration must compute all the local GEVCs as well as a matrix inverse.

Chapter 3

GEVD-based DANSE for robust node-specific signal estimation

GEVD-based low-rank approximation for distributed adaptive node-specific signal estimation in wireless sensor networks

Amin Hassani, Alexander Bertrand and Marc Moonen

Published in *IEEE Transactions on Signal Processing*, vol. 64, no. 10, May 2016, pp. 2557-2572, DOI: 10.1109/TSP.2015.2510973.

Abstract

In this chapter, we address the problem of distributed adaptive estimation of node-specific signals for signal enhancement or noise reduction in wireless sensor networks with multi-sensor nodes. The estimation is performed by a multi-channel Wiener filter (MWF) in which a low-rank approximation based on a generalized eigenvalue decomposition (GEVD) is incorporated. In non-stationary or low-SNR conditions, this GEVD-based MWF has been demonstrated to be more robust than the original MWF. In a centralized realization where a fusion center has access to all the nodes' sensor signal observations, the network-wide sensor signal correlation matrices and the low-rank approximation can be directly estimated and used to compute the network-wide GEVD-based MWF. However, in this chapter we aim to avoid centralizing the sensor signal observations, in which case the network-wide sensor signal correlation matrices cannot be estimated. To this end, we start from the so-called distributed adaptive node-specific signal estimation (DANSE) algorithm, and include GEVD-based low-rank approximations in the per-node local computations. Remarkably, the new algorithm is able to significantly compress the signal observations transmitted between the nodes, while still converging to the network-wide GEVD-based MWF as if each node would have access to all sensor signal observations, even though the low-rank approximations are applied locally at each node. We provide a theoretical convergence analysis, which shows that the algorithm converges to the network-wide GEVD-based MWF under conditions that are less strict than in the original DANSE algorithm. The convergence and performance of the algorithm are further investigated via numerical simulations.

3.1 Introduction

Spatial filtering or beamforming algorithms make use of a sensor array to exploit spatial characteristics of the sensing environment. However, traditional sensor arrays usually have a limited number of closely-spaced sensors, which then results in a rather local sampling of the spatial field [3, 4]. To collect more diverse information on the spatial field, wireless sensor networks (WSNs) have been introduced [15, 17], which consist of a multitude of sensor nodes which are distributed over the sensing environment. Each node of the WSN is equipped with a small sensor array, a digital signal processor (DSP) and with a wireless transceiver to communicate with other nodes.

To process the sensor signal observations of a WSN, one possibility is to gather them in a fusion center, which allows to compute an optimal estimate of a parameter or signal of interest. In the sequel, we refer to this as a centralized approach where nodes transmit their uncompressed sensor signal observations to either the fusion center or to all other nodes. However, this centralized approach requires a large communication bandwidth and computational power, which is often not practical in applications operating at high sampling rate and/or with many nodes. Moreover, in many WSN applications, the assumption about the availability of such a powerful fusion center cannot be made. Alternatively, a WSN further allows us to distribute the whole processing task between all nodes. This means that nodes pre-process their raw sensor signal observations and only share part of their sensor data with other nodes. The nodes then cooperate in a distributed fashion to estimate the parameter or signal of interest. In the sequel, we refer to this as a distributed approach, which is the main focus of this chapter. In general, since distributed approaches are more scalable in terms of their required communication bandwidth and computational power, they are more preferable.

A large class of estimation problems in WSNs deals with the estimation of a common network-wide parameter or signal of interest [61, 64, 124–127]. Recently, also other distributed estimation problems have been investigated in which each node is interested in estimating a different node-specific parameter or signal [10,33,34,67–69,71,80,83,118,128–130]. For example, in some applications the parameters or signals of interest must be estimated as they are observed at the local sensors of each node. This is important in, e.g., localization [118], speech enhancement in binaural hearing aids [33, 34], per-sensor signal enhancement as a pre-processing [118, 131], or blind beamforming [69, 132–134].

In this chapter, we focus on distributed node-specific *signal* estimation rather than parameter estimation. This means that each node aims to estimate samples of a desired signal, of which only noisy observations are available.

Distributed node-specific signal estimation was first introduced in the context of binaural hearing aids speech enhancement with the aim of preserving the spatial characteristics of the speech signal at the two ears, where the binaural system can indeed be considered as a 2-node WSN [34]. This approach was referred to as the distributed multichannel Wiener filter (DB-MWF) as it is a distributed realization of the (centralized) multichannel Wiener filter (MWF) [112]. The distributed adaptive node-specific signal estimation (DANSE) algorithm was then designed in [67] to estimate a node-specific desired signal at each node of a fully-connected WSN in a distributed fashion, which generalizes DB-MWF to any number of nodes and to multiple desired signal sources. Other versions of DANSE have been developed, e.g., in [71] for other topologies, in [68] for asynchronous node-updating, in [69] for node-specific linear constraints, and in [10], [129] for more robust realizations. The DANSE algorithm provides a distributed realization of the network-wide MWF and its design relies on the assumption that the node-specific desired signals share a common (unknown) latent signal subspace [67]. By exploiting this so-called common interest of the nodes, DANSE significantly compresses the sensor signal observations that are transmitted between the nodes, while still converging to the network-wide MWF solutions as if each node would have access to all sensor signal observations in the WSN.

The MWF is implicitly based on a low-rank approximation of a signal correlation matrix with a so-called column decomposition [103]. However, in low SNR conditions, and for highly non-stationary noise in particular, this signal correlation matrix approximation may become indefinite, possibly leading to an unstable filter [103]. Alternatively, either an eigenvalue decomposition (EVD)-based or a generalized EVD (GEVD)-based low-rank approximation can be applied to improve the estimation performance in such cases. MWF with GEVD-based low-rank approximation has been shown to deliver the best performance, as it effectively selects the 'mode' corresponding to the highest SNR [103]. The resulting spatial filter is referred to as the GEVD-based MWF.

The objective of this chapter is to develop a distributed realization of this network-wide GEVD-based MWF for node-specific signal estimation in a fully-connected WSN[1]. It turns out that a surprisingly minor modification in the DANSE algorithm [67,68] achieves this goal, i.e., each node merely has to locally apply a GEVD-based low-rank approximation using local signal correlation matrices. However, the fact that the inclusion of these GEVD-based low-rank approximations in the per-node local computations results in the network-wide

[1]This is mainly for the sake of an easy exposition, since all results can be extended to, e.g., tree topologies using similar strategies as in [71]. In [71] it has been proved that a DANSE-based approach will not converge if the network contains cycles, due to feedback components.

GEVD-based MWF is remarkable and not intuitive. Furthermore, this minor modification has a large impact on the convergence analysis of the algorithm, which can no longer rely on the strategy used to prove convergence in the original DANSE algorithm [67, 68].

Similar to the DANSE algorithm, the GEVD-based DANSE algorithm compresses the sensor signal observations at each node into a smaller number of signal observations which are then broadcast to the other nodes. Under some technical conditions, we show that even though the GEVD-based DANSE algorithm is not able to compute the network-wide signal correlation matrices (and their GEVD) from these compressed sensor signal observations, the algorithm does converge to the network-wide GEVD-based MWF as if each node would have access to all (uncompressed) sensor signal observations. Furthermore, the conditions under which the algorithm converges to the network-wide solution are less strict than those of the original DANSE algorithm.

The chapter is organized as follows. The data model and the problem statement are presented in Section 3.2. Network-wide GEVD-based MWF is explained in Section 3.3. The GEVD-based DANSE algorithm and its convergence analysis are developed in Section 3.4. The convergence is illustrated with numerical simulations in Section 3.5. Finally conclusions are drawn in Section 3.6.

3.2 Data model

We consider K multi-sensor nodes in a fully-connected WSN where the data broadcast by a node can be received by all other $(K-1)$ nodes in the network through an ideal wireless link. Each node $k \in \mathcal{K} = \{1, \ldots, K\}$ is assumed to collect observations of a complex-valued M_k-channel sensor signal $\mathbf{y}_k$. Note that this also allows for a hierarchical WSN where K master nodes collect sensor signal observations from M_k slave nodes each with a single sensor. $\mathbf{y}_k$ is modeled as

$$\mathbf{y}_k = \mathbf{s}_k + \mathbf{n}_k = \mathbf{A}_k \check{\mathbf{s}} + \mathbf{n}_k \tag{3.1}$$

where $\check{\mathbf{s}}$ is a latent S-channel signal defining S mutually uncorrelated latent source signals of interest to the WSN, $\mathbf{A}_k$ is an unknown $M_k \times S$ complex-valued steering matrix, and $\mathbf{n}_k$ is additive noise which is possibly correlated over the different nodes. The sensor signal $\mathbf{y}_k$ is assumed to satisfy short-term stationarity and ergodicity conditions. By stacking all $\mathbf{y}_k$, $\mathbf{n}_k$ and $\mathbf{s}_k$, we obtain the network-wide M-channel signals $\mathbf{y}$, $\mathbf{s}$ and $\mathbf{n}$, respectively, where $M = \sum_{k=1}^{K} M_k$ and

$$\mathbf{y} = \mathbf{s} + \mathbf{n} = \mathbf{A}\check{\mathbf{s}} + \mathbf{n} \tag{3.2}$$

where $\mathbf{A}$ is an $M \times S$ matrix which is the stacked version of all $\mathbf{A}_k$ steering matrices.

The goal for each node $k \in \mathcal{K}$ is to estimate the L-channel signal $\mathbf{d}_k$ which is defined as L pre-selected channels of $\mathbf{s}_k$, i.e., node k aims to denoise L channels of $\mathbf{y}_k$, where L can be any number between 1 and M_k. This means that the estimation procedure will preserve the node-specific spatial information in $\mathbf{s}_k$ while reducing the noise $\mathbf{n}_k$. Without loss of generality (w.l.o.g.), we assume that $\mathbf{d}_k$ corresponds to the first L channels of $\mathbf{s}_k$, i.e.,

$$\mathbf{d}_k = \mathbf{E}_{d_k}^T \mathbf{s}_k \quad \forall k \in \mathcal{K} \tag{3.3}$$

where $\mathbf{E}_{d_k}$ is an $M_k \times L$ matrix which selects the first L channels of $\mathbf{s}_k$, i.e., $\mathbf{E}_{d_k} = [\mathbf{I}_L \ \mathbf{0}]^T$, where $\mathbf{I}_L$ is the $L \times L$ identity matrix and $\mathbf{0}$ is an all-zero matrix with matching dimensions. Note that since š and $\mathbf{A}_k$ are both assumed to be unknown, nodes do not know how their node-specific desired signals $\mathbf{d}_k$ relate to each other, even though they are in the same latent signal subspace. It should be mentioned that since all signals in (3.1)-(3.3) are assumed to be complex-valued, the model also includes, e.g., convolutive time-domain mixtures, described as instantaneous per-frequency mixtures in the (short-term) Fourier transform domain.

3.3 Network-wide GEVD-based MWF

3.3.1 Network-wide MWF

We first consider the centralized estimation problem where we assume that all nodes transmit all their sensor signal observations of $\mathbf{y}_k$ to all other nodes. Therefore the objective for node k is to estimate its node-specific desired signal $\mathbf{d}_k$, from all the sensor signals in $\mathbf{y}$. Node k uses an $M \times L$ linear estimator $\hat{\mathbf{W}}_k$, which can be viewed as a network-wide spatial filter, to estimate $\mathbf{d}_k$ as $\hat{\mathbf{d}}_k = \hat{\mathbf{W}}_k^H \mathbf{y}$, where superscript H denotes the conjugate transpose operator and where the hat ($\hat{\cdot}$) refers to the centralized approach. The MWF [112] computes $\hat{\mathbf{W}}_k$ based on the linear minimum mean square error (LMMSE) criterion, i.e.,

$$\hat{\mathbf{W}}_k^{\text{LMMSE}} = \underset{\mathbf{W}_k}{\operatorname{argmin}} \, E\left\{ \left\| \mathbf{d}_k - \mathbf{W}_k^H \mathbf{y} \right\|^2 \right\} \tag{3.4}$$

where $E\{.\}$ is the expected value operator. Assuming $\mathbf{R}_{yy} = E\{\mathbf{y}\mathbf{y}^H\}$ has full rank, the unique solution of (3.4) is [112]:

$$\hat{\mathbf{W}}_k^{\text{LMMSE}} = \mathbf{R}_{yy}^{-1} \mathbf{R}_{yd_k} \tag{3.5}$$

with $\mathbf{R}_{yd_k} = E\{\mathbf{y}\mathbf{d}_k^H\}$. Note that (3.5) can be rewritten as

$$\hat{\mathbf{W}}_k^{\text{LMMSE}} = \mathbf{R}_{yy}^{-1}\mathbf{R}_{ss}\hat{\mathbf{E}}_{d_k} \tag{3.6}$$

where $\mathbf{R}_{ss} = E\{\mathbf{s}\mathbf{s}^H\}$, and where $\hat{\mathbf{E}}_{d_k} = [\mathbf{0}\ \mathbf{I}_L\ \mathbf{0}]^T$ is an $M \times L$ matrix that selects the L columns of $\mathbf{R}_{ss}$ corresponding to the channels of $\mathbf{s}$ that are included in $\mathbf{d}_k$.

The sensor signal correlation matrix $\mathbf{R}_{yy}$ is often not available in practice, but can be estimated via sample averaging. Hence, we define the $M \times N$ observation matrix $\mathbf{Y}$, where each column corresponds to an observation of $\mathbf{y}$ at a particular time instant, such that $\mathbf{R}_{yy}$ can be estimated as (overline indicates the estimation)

$$\bar{\mathbf{R}}_{yy} \approx \frac{1}{N}\mathbf{Y}\mathbf{Y}^H \tag{3.7}$$

and when having an infinitely long observation window we can write $\mathbf{R}_{yy} = \lim_{N\to\infty} \frac{1}{N}\mathbf{Y}\mathbf{Y}^H$. We also define the network-wide noise correlation matrix $\mathbf{R}_{nn} = E\{\mathbf{n}\mathbf{n}^H\}$, where it is assumed that $\mathbf{R}_{nn}$ is either known a-priori or can be estimated from noise-only segments in the sensor signal observations (similar to (3.7)). The latter can be performed when the desired signal source has an on-off behavior, such as in speech enhancement where $\mathbf{R}_{yy}$ and $\mathbf{R}_{nn}$ can be estimated during 'speech-and-noise' and 'noise-only' segments, respectively, using a voice activity detection [112], [34].

Assuming $\mathbf{s}$ and $\mathbf{n}$ are uncorrelated, the signal correlation matrix $\mathbf{R}_{ss}$ can then be estimated as $\bar{\mathbf{R}}_{ss} = \bar{\mathbf{R}}_{yy} - \bar{\mathbf{R}}_{nn}$. This $\bar{\mathbf{R}}_{ss}$ can then be used together with $\bar{\mathbf{R}}_{yy}$ to compute $\hat{\mathbf{W}}_k^{\text{LMMSE}}$ as required in (3.6).

3.3.2 Network-wide GEVD-based MWF

In theory $\mathbf{R}_{ss}$ is a rank-S matrix, which can be verified by considering

$$\mathbf{R}_{ss} = E\{\mathbf{s}\mathbf{s}^H\} = \mathbf{A}\mathbf{\Phi}\mathbf{A}^H \tag{3.8}$$

where $\mathbf{\Phi} = \text{diag}\{\phi_1, \ldots, \phi_S\}$ is an $S \times S$ diagonal matrix, where $\phi_s = E\{|\check{s}_s|^2\}$, where $\check{s}_s$ denotes the s-th channel of $\check{\mathbf{s}}$. In practice, however, the estimated $\bar{\mathbf{R}}_{ss}$ generally has a rank greater than S, and it may even not be positive semi-definite due to the subtraction $\bar{\mathbf{R}}_{yy} - \bar{\mathbf{R}}_{nn}$. In this case, it has been demonstrated in [103] that incorporating a low rank approximation based on either the eigenvalue decomposition (EVD) of $\bar{\mathbf{R}}_{ss}$ or the generalized eigenvalue decomposition (GEVD) of $\bar{\mathbf{R}}_{yy}$ and $\bar{\mathbf{R}}_{nn}$ enhances the estimation performance of the MWF, especially in low-SNR conditions. The GEVD-based rank-R approximation has been shown to deliver the best performance, as it effectively

selects the R 'modes' corresponding to the highest SNR [103]. In the rest of this section, the network-wide GEVD-based MWF is explained in detail.

In order to perform a GEVD of the ordered matrix pair $(\bar{\mathbf{R}}_{yy}, \bar{\mathbf{R}}_{nn})$, each generalized eigenvector (GEVC) and its corresponding generalized eigenvalue (GEVL), $\hat{\mathbf{x}}_m$ and $\hat{\lambda}_m$ $(m = 1 \dots M)$, respectively, must be computed such that $\bar{\mathbf{R}}_{yy}\hat{\mathbf{x}}_m = \hat{\lambda}_m \bar{\mathbf{R}}_{nn}\hat{\mathbf{x}}_m$ [122], or equivalently

$$\bar{\mathbf{R}}_{yy}\hat{\mathbf{X}} = \bar{\mathbf{R}}_{nn}\hat{\mathbf{X}}\hat{\mathbf{L}} \tag{3.9}$$

where $\hat{\mathbf{X}} = [\hat{\mathbf{x}}_1 ... \hat{\mathbf{x}}_M]$ and $\hat{\mathbf{L}} = \text{diag}\{\hat{\lambda}_1 \dots \hat{\lambda}_M\}$. It can be shown that the GEVD in (3.9) extracts the directions with maximal SNR, similar to how principal component analysis extracts the directions with maximal variance [1,122]. Note that when $\bar{\mathbf{R}}_{nn}$ is invertible, (3.9) can be written as a non-symmetric EVD as

$$\bar{\mathbf{R}}_{nn}^{-1}\bar{\mathbf{R}}_{yy} = \hat{\mathbf{X}}\hat{\mathbf{L}}\hat{\mathbf{X}}^{-1}. \tag{3.10}$$

In the sequel, we assume w.l.o.g. that the GEVLs in $\hat{\mathbf{L}}$ are sorted in descending order, i.e., with $\hat{\lambda}_1$ the largest GEVL. This GEVD is equivalent to a joint diagonalization of $\bar{\mathbf{R}}_{yy}$ and $\bar{\mathbf{R}}_{nn}$ which can be generally written as

$$\bar{\mathbf{R}}_{yy} = \hat{\mathbf{Q}}\hat{\mathbf{\Sigma}}\hat{\mathbf{Q}}^H, \; \bar{\mathbf{R}}_{nn} = \hat{\mathbf{Q}}\hat{\mathbf{\Gamma}}\hat{\mathbf{Q}}^H \tag{3.11}$$

where $\hat{\mathbf{Q}}$ is a full-rank $M \times M$ matrix (not necessarily orthogonal), and where $\hat{\mathbf{\Sigma}}$ and $\hat{\mathbf{\Gamma}}$ are diagonal matrices. Using (3.10) it can then be verified that $\hat{\mathbf{Q}} = \hat{\mathbf{X}}^{-H}$, and that $\hat{\mathbf{L}} = \hat{\mathbf{\Sigma}}\hat{\mathbf{\Gamma}}^{-1}$ and that $\hat{\mathbf{\Sigma}} = \hat{\mathbf{X}}^H\bar{\mathbf{R}}_{yy}\hat{\mathbf{X}}$ and $\hat{\mathbf{\Gamma}} = \hat{\mathbf{X}}^H\bar{\mathbf{R}}_{nn}\hat{\mathbf{X}}$. Since the GEVCs are defined up to a scaling, we assume w.l.o.g. that all $\hat{\mathbf{x}}_m$'s are scaled such that

$$\hat{\mathbf{X}}^H\bar{\mathbf{R}}_{nn}\hat{\mathbf{X}} = \mathbf{I}_M \tag{3.12}$$

where $\mathbf{I}_M$ denotes the $M \times M$ identity matrix. It follows then that $\hat{\mathbf{\Gamma}} = \mathbf{I}_M$ and $\hat{\mathbf{\Sigma}} = \hat{\mathbf{L}}$, i.e., (3.11) becomes

$$\bar{\mathbf{R}}_{yy} = \hat{\mathbf{Q}}\hat{\mathbf{L}}\hat{\mathbf{Q}}^H, \; \bar{\mathbf{R}}_{nn} = \hat{\mathbf{Q}}\hat{\mathbf{Q}}^H. \tag{3.13}$$

Reconsidering $\mathbf{y} = \mathbf{s} + \mathbf{n}$ and (3.13), it follows that

$$\bar{\mathbf{R}}_{ss} = \bar{\mathbf{R}}_{yy} - \bar{\mathbf{R}}_{nn} = \hat{\mathbf{Q}}\hat{\mathbf{L}}\hat{\mathbf{Q}}^H - \hat{\mathbf{Q}}\hat{\mathbf{Q}}^H \tag{3.14}$$

$$= \hat{\mathbf{Q}}\big(\hat{\mathbf{L}} - \mathbf{I}_M\big)\hat{\mathbf{Q}}^H. \tag{3.15}$$

The rank-R approximation of $\bar{\mathbf{R}}_{ss}$ becomes $\hat{\mathbf{Q}}\hat{\mathbf{\Delta}}\hat{\mathbf{Q}}^H$ with $\hat{\mathbf{\Delta}}$ denoting the diagonal matrix $(\hat{\mathbf{L}} - \mathbf{I}_M)$ with the $M - R$ smallest diagonal entries set to zero, i.e.,

$$\hat{\mathbf{\Delta}} = \text{diag}\,\{(\hat{\lambda}_1 - 1), \dots, (\hat{\lambda}_R - 1), \underbrace{0, \dots, 0}_{M-R}\}. \tag{3.16}$$

By replacing $\bar{\mathbf{R}}_{ss}$ with its rank-R approximation in (3.6), and from (3.13) the network-wide GEVD-based MWF is obtained as

$$\hat{\mathbf{W}}_k = \bar{\mathbf{R}}_{yy}^{-1}\hat{\mathbf{Q}}\hat{\mathbf{\Delta}}\hat{\mathbf{Q}}^H\hat{\mathbf{E}}_{d_k} = (\hat{\mathbf{Q}}\hat{\mathbf{L}}\hat{\mathbf{Q}}^H)^{-1}\hat{\mathbf{Q}}\hat{\mathbf{\Delta}}\hat{\mathbf{Q}}^H\hat{\mathbf{E}}_{d_k} \tag{3.17}$$

$$= \hat{\mathbf{Q}}^{-H}\hat{\mathbf{L}}^{-1}\hat{\mathbf{Q}}^{-1}\hat{\mathbf{Q}}\hat{\mathbf{\Delta}}\hat{\mathbf{Q}}^H\hat{\mathbf{E}}_{d_k} = \hat{\mathbf{Q}}^{-H}\hat{\mathbf{L}}^{-1}\hat{\mathbf{\Delta}}\hat{\mathbf{Q}}^H\hat{\mathbf{E}}_{d_k}$$

where $\hat{\mathbf{L}}^{-1}\hat{\mathbf{\Delta}} = diag\{1-\frac{1}{\hat{\lambda}_1},\dots,1-\frac{1}{\hat{\lambda}_R},\underbrace{0,\dots,0}_{M-R}\}$. The solution (3.17) will be referred to as the network-wide GEVD-based MWF.

Section 3.4 will introduce the GEVD-based DANSE algorithm, which obtains the signal estimates $\hat{\mathbf{d}}_k = \hat{\mathbf{W}}_k^H\mathbf{y}$, i.e., the output signals of the network-wide GEVD-based MWF in a distributed fashion, where all nodes participate in the overall processing and only broadcast compressed sensor signal observations.

When S is known, an obvious choice of R is $R = S$, which is motivated by (3.8). In this case the network-wide GEVD-based MWF (3.17) is (asymptotically) equivalent to the LMMSE solution (3.4). However, cases where $R \neq S$ may occur, e.g., when S is either not known or wrongly estimated. In such cases, if $R \geq S$, still the solution (3.17) is (asymptotically) equivalent to the LMMSE solution (3.4)[2]. On the other hand if $R < S$, the rank-R approximation effectively redefines (imposes) a common latent signal subspace of dimension R for the underlying data model, while the actual data model (3.1) indeed defines a common latent signal subspace of dimension S (see (3.8)). Nevertheless, it can be shown that (3.17) still provides a spatial filter which extracts the *R-dimensional* signal subspace with highest SNR [103]. For the sake of an easier exposition, in the sequel we assume that S is known and hence we set $R = S$ and replace S by R everywhere. This is w.l.o.g. and only for notational convenience in the sequel. It is noted that the convergence proof in Section 3.4.6 is independent of the parameter S.

3.4 GEVD-based DANSE

So far we have assumed that at each node $k \in \mathcal{K}$, the estimation of $\mathbf{d}_k$ is performed based on the entire M-dimensional sensor signal vector $\mathbf{y}$. The objective now is to design an algorithm that computes the network-wide GEVD-based MWF in a distributed fashion in a fully-connected WSN, i.e., obtaining $\hat{\mathbf{d}}_k = \hat{\mathbf{W}}_k^H\mathbf{y}$, where $\hat{\mathbf{W}}_k^H$ is defined in (3.17), at each node $k \in \mathcal{K}$, without broadcasting observations of the entire sensor signal vector $\mathbf{y}$. The computational

[2]Note that as will be discussed in Section 3.4, the distributed implementation of this case requires more communication bandwidth, comparing to the case where $R = S$.

load is then shared between the different nodes, and each node $k \in \mathcal{K}$ only broadcasts observations of a R-channel compressed signal to the other nodes, rather than observations of its full M_k-dimensional signal vector $\mathbf{y}_k$ (assuming $R < M_k, \forall k \in \mathcal{K}$)[3]. It will be shown in the sequel that the GEVD-based DANSE algorithm nevertheless converges to the network-wide GEVD-based MWF (3.17). The proposed algorithm is based on a modification of the original DANSE algorithm in [67]. The algorithm derivation below includes all the necessary ingredients of DANSE to describe the GEVD-based DANSE algorithm. However, for more details and intuition on the DANSE algorithm, we refer to [67].

In the sequel, we will often use the letters k and q for node indices. A notation with the node index $(.)_q$ is used to refer to the *updating* node q at iteration i of the (GEVD-based) DANSE algorithm, whereas a notation with the node index $(.)_k$ is used to refer to general statements that apply to all nodes $k \in \mathcal{K}$ concurrently.

3.4.1 Simplification $L = R$

In general, the parameter L, which describes the number of desired channels in (3.3), can be chosen independently from R. However, for the sake of an easier exposition, but w.l.o.g., we will assume that $L = R$ in the sequel and replace L by R, which allows for a more elegant description of the GEVD-based DANSE algorithm. Nevertheless, we explain below what measures should be taken in the case where $L \neq R$.

If $L < R$, node k can effectively increase L to R by adding $R - L$ auxiliary channels in (3.3). As a result, $\mathbf{d}_k$ will fully capture the latent R-dimensional signal subspace defined by the rank-R approximation defined in (3.15)-(3.16). In cases where $L > R$, only R channels of $\mathbf{d}_k$ will be incorporated in the GEVD-based DANSE algorithm, as the remaining $L - R$ channels will not have an influence on the computations in the algorithm. It will be shown in Remark 6 in Section 3.4.6 that the network-wide estimates of the remaining $L - R$ channels of $\mathbf{d}_k$ can be computed from the estimates of the R channels that were included in the algorithm.

[3]For the sake of an easier exposition, we will assume from now on that $R < M_k, \forall k \in \mathcal{K}$. If there exists a node k for which $R \geq M_k$, node k should broadcast its uncompressed sensor signal observations of $\mathbf{y}_k$ to other nodes as no compression can be obtained by the GEVD-based DANSE algorithm in this case, which will become clear later.

3.4.2 Algorithm assumptions

The GEVD-based DANSE algorithm basically assumes that 1) $\mathbf{R}_{yy}$ is full rank 2) at each node k the steering matrix $\mathbf{A}_k$ is either static or slowly varying (but unknown) 3) the signals are assumed to satisfy short-term stationarity and ergodicity conditions 4) 'noise-only' segments are available to compute the local noise statistics 5) the network has ideal communication links 6) the network is fully-connected (although the results can be extended to tree topologies, similar to the original DANSE algorithm in [71]) 7) $R < M_k$, $\forall k \in \mathcal{K}$ and 8) the sampling rates between the nodes are synchronized. Essentially, these are the same assumptions and conditions as the original DANSE algorithm in [67,68,71]. However, a major difference is the fact that here, there is no explicit assumption that the desired signals of the different nodes span a low-dimensional latent signal subspace. This is because the low-rank structure is imposed by the algorithm itself, by means of the GEVD.

3.4.3 Algorithm description

In GEVD-based DANSE, each node $k \in \mathcal{K}$ first compresses its M_k-channel signal $\mathbf{y}_k$ into an R-channel signal $\mathbf{z}_k^i = \mathbf{F}_k^{iH}\mathbf{y}_k$ with an $M_k \times R$ compression matrix $\mathbf{F}_k^i$, which changes over time and which will be defined later (see (3.25)), and then broadcasts observations of $\mathbf{z}_k^i$ to all other nodes (i is the iteration index). Consequently and compared to the network-wide GEVD-based MWF, the algorithm reduces the required per-node communication bandwidth by a factor of $\max\{(M_k/R), 1\}$.

Considering the stacked signal $\mathbf{z}^i = [\mathbf{z}_1^{iT} \dots \mathbf{z}_K^{iT}]^T$, $\mathbf{z}_{-k}^i$ denotes the vector $\mathbf{z}^i$ with $\mathbf{z}_k^i$ omitted. Each node k in the GEVD-based DANSE algorithm has access to a P_k-channel signal $\widetilde{\mathbf{y}}_k^i$ which is defined as $\widetilde{\mathbf{y}}_k^i = [\mathbf{y}_k^T \ \mathbf{z}_{-k}^{iT}]^T$, with $P_k = M_k + R(K-1)$. We use a similar notation for the signal and the noise component of $\widetilde{\mathbf{y}}_k^i$, i.e., $\widetilde{\mathbf{s}}_k^i$ and $\widetilde{\mathbf{n}}_k^i$.

In the DANSE algorithm [67], the updating node q at iteration i solves the following local LMMSE problem:

$$(\widetilde{\mathbf{W}}_q^{i+1})^{\text{LMMSE}} = \underset{\mathbf{W}_q}{\text{argmin}}\, E\left\{\left\|\mathbf{d}_q - \mathbf{W}_q^H \widetilde{\mathbf{y}}_q^i\right\|^2\right\} \tag{3.18}$$

where the solution is

$$(\widetilde{\mathbf{W}}_q^{i+1})^{\text{LMMSE}} = (\bar{\mathbf{R}}_{\tilde{y}_q\tilde{y}_q}^i)^{-1}\bar{\mathbf{R}}_{\tilde{s}_q\tilde{s}_q}^i\widetilde{\mathbf{E}}_{d_q} \tag{3.19}$$

(compare with (3.4)-(3.6)) where $\bar{\mathbf{R}}^i_{\tilde{y}_q\tilde{y}_q}$, $\bar{\mathbf{R}}^i_{\tilde{n}_q\tilde{n}_q}$ and $\bar{\mathbf{R}}^i_{\tilde{s}_q\tilde{s}_q} = \bar{\mathbf{R}}^i_{\tilde{y}_q\tilde{y}_q} - \bar{\mathbf{R}}^i_{\tilde{n}_q\tilde{n}_q}$ are the P_q-dimensional correlation matrix corresponding respectively to $\widetilde{\mathbf{y}}^i_q$, $\widetilde{\mathbf{n}}^i_q$ and $\widetilde{\mathbf{s}}^i_q$, and where $\widetilde{\mathbf{E}}_{d_q}$ is a $P_q \times R$ matrix which selects the first R columns of $\bar{\mathbf{R}}^i_{\tilde{s}_q\tilde{s}_q}$.

Similar to (3.9)-(3.15), we now perform a local GEVD at node q on the matrix pair $\bar{\mathbf{R}}^i_{\tilde{y}_q\tilde{y}_q}$ and $\bar{\mathbf{R}}^i_{\tilde{n}_q\tilde{n}_q}$, i.e.,

$$(\bar{\mathbf{R}}^i_{\tilde{n}_q\tilde{n}_q})^{-1}\bar{\mathbf{R}}^i_{\tilde{y}_q\tilde{y}_q} = \widetilde{\mathbf{X}}^i_q\widetilde{\mathbf{L}}^i_q(\widetilde{\mathbf{X}}^i_q)^{-1} \tag{3.20}$$

$$\text{s.t. } \widetilde{\mathbf{X}}^{i\,H}_q\bar{\mathbf{R}}^i_{\tilde{n}_q\tilde{n}_q}\widetilde{\mathbf{X}}^i_q = \mathbf{I}_{P_q} \tag{3.21}$$

and based on the joint diagonalization representation we have (similar to (3.13))

$$\bar{\mathbf{R}}^i_{\tilde{y}_q\tilde{y}_q} = \widetilde{\mathbf{Q}}^{-i}_q\widetilde{\mathbf{L}}^i_q\widetilde{\mathbf{Q}}^{i\,H}_q\,,\ \bar{\mathbf{R}}^i_{\tilde{n}_q\tilde{n}_q} = \widetilde{\mathbf{Q}}^{-i}_q\widetilde{\mathbf{Q}}^{i\,H}_q \tag{3.22}$$

$$\bar{\mathbf{R}}^i_{\tilde{s}_q\tilde{s}_q} = \widetilde{\mathbf{Q}}^{-i}_q\big(\widetilde{\mathbf{L}}^i_q - \mathbf{I}_{P_q}\big)\widetilde{\mathbf{Q}}^{i\,H}_q \tag{3.23}$$

where $\widetilde{\mathbf{Q}}^{-i}_q = (\widetilde{\mathbf{X}}^i_q)^{-H}$. When replacing $\bar{\mathbf{R}}^i_{\tilde{s}_q\tilde{s}_q}$ by its GEVD-based rank-R approximation, solution (3.19) becomes (compare with (3.17))

$$\widetilde{\mathbf{W}}^{i+1}_q = (\bar{\mathbf{R}}^i_{\tilde{y}_q\tilde{y}_q})^{-1}\widetilde{\mathbf{Q}}^{-i}_q\widetilde{\mathbf{\Delta}}^i_q\widetilde{\mathbf{Q}}^{i\,H}_q\widetilde{\mathbf{E}}_{d_q} \tag{3.24}$$

where $\widetilde{\mathbf{\Delta}}^i_q$ is the $P_q \times P_q$ diagonal matrix $(\widetilde{\mathbf{L}}^i_q - \mathbf{I}_{P_q})$ with the $P_q - R$ smallest diagonal entries set to zero. The compression matrix $\mathbf{F}^i_q$ at node q is then chosen as

$$\mathbf{F}^{i+1}_q = \big[\mathbf{I}_{M_q}\ \mathbf{0}\big]\ \widetilde{\mathbf{W}}^{i+1}_q \tag{3.25}$$

, i.e., it extracts the part of $\widetilde{\mathbf{W}}^{i+1}_q$ that is applied to the local sensor signals only (excluding the $\mathbf{z}$-signals from other nodes). This compression rule allows for a common parameterization of all the network-wide estimators corresponding to each node, as explained later (see (3.33)). Finally node q estimates its node-specific R-channel desired signal as $\widetilde{\mathbf{d}}^i_q = (\widetilde{\mathbf{W}}^{i+1}_q)^H\widetilde{\mathbf{y}}^i_q$. The resulting GEVD-based DANSE algorithm is described in Table 3.1. It is noted that this algorithm is exactly the same as the original DANSE algorithm, except for the fact that (3.28) now replaces (3.19) in the original formulation of DANSE. Just as DANSE converges to the network-wide MWF (3.6), we will show in Subsection 3.4.6 that this GEVD-based DANSE algorithm indeed converges to the network-wide GEVD-based MWF (3.17). Despite this remarkably elegant similarity, the convergence of GEVD-based DANSE to (3.17) is far from trivial, and can not rely on the convergence results in [67] for the original DANSE algorithm.

Remark 1. In order to ensure that the updates of the GEVD-based DANSE algorithm in Table 3.1 are well-defined in each iteration, for the time being, we make these assumptions:

- $\forall i \in \mathbb{N}$, the matrices $\bar{\mathbf{R}}^i_{\tilde{y}_q\tilde{y}_q}$ and $\bar{\mathbf{R}}^i_{\tilde{n}_q\tilde{n}_q}$ are full rank.
- $\forall i \in \mathbb{N}$, the $(R+1)$ largest GEVLs of $(\bar{\mathbf{R}}^i_{\tilde{y}_q\tilde{y}_q}, \bar{\mathbf{R}}^i_{\tilde{n}_q\tilde{n}_q})$ are distinct (with algebraic multiplicity 1), i.e.,

$$\exists \epsilon > 0, \forall i \in \mathbb{N}, \forall n \in \{1, \dots, R\} : |\tilde{\lambda}^i_n - \tilde{\lambda}^i_{n+1}| > \epsilon \tag{3.26}$$

  where $\tilde{\lambda}^i_n$ is the n-th GEVL of $(\bar{\mathbf{R}}^i_{\tilde{y}_q\tilde{y}_q}, \bar{\mathbf{R}}^i_{\tilde{n}_q\tilde{n}_q})$ (see 3.20).

This guarantees that $\widetilde{\mathbf{W}}^i_q$ is well defined in each iteration. It is noted that the assumptions are merely made for the sake of an easy exposition, and although they are mostly satisfied in practice, we briefly describe in Appendix B.2 how the algorithm can be modified in the rare cases where the assumptions are violated.

3.4.4 Parameterization of the solution space

Before proving convergence and optimality of the GEVD-based DANSE algorithm, we provide some intuition on the parameterization of its solution space.

We first define a partitioning of the network-wide GEVD-based MWF in (3.17) as $\hat{\mathbf{W}}_k = [\hat{\mathbf{W}}_{k1} \dots \hat{\mathbf{W}}_{kK}]^T$ with $\hat{\mathbf{W}}_{kn}$ denoting[4] the $M_n \times R$ submatrix of $\hat{\mathbf{W}}_k$ that is applied to the sensor signal observations $\mathbf{y}_n$ of node n. With this we can extract the $M_k \times R$ submatrix $\mathbf{W}^{i+1}_{kk}$ at each iteration of the GEVD-based DANSE algorithm as

$$\mathbf{W}^{i+1}_{kk} = [\mathbf{I}_{M_k} \ \mathbf{0}] \, \widetilde{\mathbf{W}}^{i+1}_k \tag{3.31}$$

since this is the part of $\widetilde{\mathbf{W}}^{i+1}_k$ that is applied to the local sensor signals $\mathbf{y}_k$. We also define an $R(K-1) \times R$ matrix $\mathbf{G}_{k-k}$ as

$$\mathbf{G}^{i+1}_{k-k} = \begin{bmatrix} \mathbf{0} & \mathbf{I}_{R(K-1)} \end{bmatrix} \widetilde{\mathbf{W}}^{i+1}_k \tag{3.32}$$

$$\mathbf{G}^{i+1}_{k-k} \triangleq [(\mathbf{G}^{i+1}_{k1})^T \dots (\mathbf{G}^{i+1}_{k(k-1)})^T (\mathbf{G}^{i+1}_{k(k+1)})^T \dots (\mathbf{G}^{i+1}_{kK})^T]^T$$

where submatrix $\mathbf{G}_{kn}$ is an $R \times R$ transformation matrix which node k applies to the broadcast signal $\mathbf{z}^i_n$ received from node $n \in \mathcal{K} \backslash k$. Considering these

[4]Note that in few instances where a notation with a node index $(.)_{kn}$ is used, we merely refer to any two nodes $k, n \in \mathcal{K}$.

Table 3.1: GEVD-based DANSE algorithm

1. Set $i \leftarrow 0$, $q \leftarrow 1$, and initialize all $\mathbf{F}_k^0$ and $\widetilde{\mathbf{W}}_k^0$, $\forall\, k \in \mathcal{K}$, with random entries.

2. Each node $k \in \mathcal{K}$ broadcasts N new observations of its R-channel compressed signal $\mathbf{z}_k^i$:
$$\mathbf{z}_k^i[iN+j] = \mathbf{F}_k^{i\,H}\mathbf{y}_k^i[iN+j], \qquad j = 1 \dots N \tag{3.27}$$
where the notation $[.]$ denotes a sample index.

3. At node q:
   - Compute $\bar{\mathbf{R}}^i_{\tilde{y}_q\tilde{y}_q}$ and $\bar{\mathbf{R}}^i_{\tilde{n}_q\tilde{n}_q}$ via sample averaging using the samples at times $iN+1$ up to $(i+1)N$.
   - Compute $\widetilde{\mathbf{Q}}_q^i$ and $\widetilde{\mathbf{\Delta}}_q^i$ from the GEVD of $(\bar{\mathbf{R}}^i_{\tilde{y}_q\tilde{y}_q}, \bar{\mathbf{R}}^i_{\tilde{n}_q\tilde{n}_q})$ similar to (3.20)-(3.23).
   - Compute the local MWF with rank-R approximation of $\bar{\mathbf{R}}^i_{\tilde{s}_q\tilde{s}_q}$ as in (3.24):
$$\widetilde{\mathbf{W}}_q^{i+1} = (\bar{\mathbf{R}}^i_{\tilde{y}_q\tilde{y}_q})^{-1}\widetilde{\mathbf{Q}}_q^i\widetilde{\mathbf{\Delta}}_q^i\widetilde{\mathbf{Q}}_q^{i\,H}\widetilde{\mathbf{E}}_{d_q} \tag{3.28}$$
   - Update the compression matrix as in (3.25)
$$\mathbf{F}_q^{i+1} = \begin{bmatrix}\mathbf{I}_{M_q} & \mathbf{0}\end{bmatrix}\widetilde{\mathbf{W}}_q^{i+1} \tag{3.29}$$

4. Other nodes $k \in \mathcal{K}\backslash q$ update their parameters as $\widetilde{\mathbf{W}}_k^{i+1} = \widetilde{\mathbf{W}}_k^i$ and $\mathbf{F}_k^{i+1} = \mathbf{F}_k^i$.

5. Each node $k \in \mathcal{K}$ estimates the next N samples of its M_k-channel signal $\mathbf{d}_k$, as
$$\widetilde{\mathbf{d}}_k^i[iN+j] = (\widetilde{\mathbf{W}}_k^{i+1})^H\widetilde{\mathbf{y}}_k^i[iN+j] \tag{3.30}$$

6. $i \leftarrow i+1$ and $q \leftarrow (q \mod K)+1$ and return to step 2.

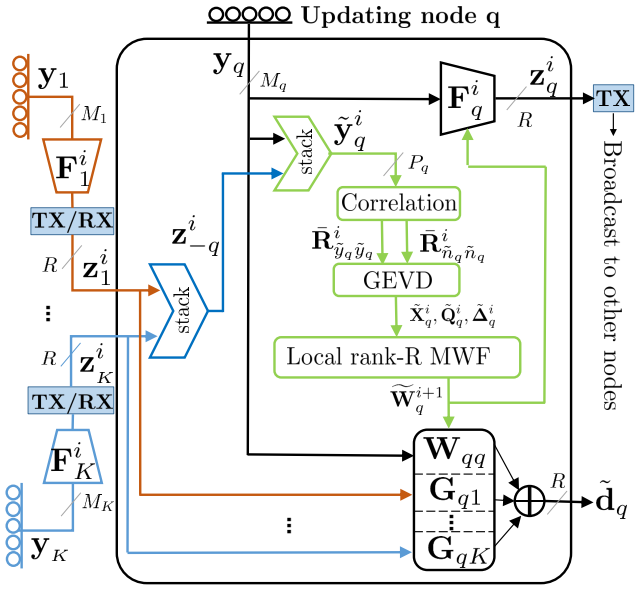


Figure 3.1: GEVD-based DANSE block scheme for an updating node q in a fully-connected WSN.

notations, a block scheme of the GEVD-based DANSE algorithm for an updating node q is depicted in Fig. 3.1. When using the compression rules as defined in (3.25), the network-wide filter at each node $k \in \mathcal{K}$ can then be written as

$$\mathbf{W}_k^i = \begin{bmatrix} \mathbf{F}_1^i \mathbf{G}_{k1}^i \\ \vdots \\ \mathbf{W}_{kk}^i \\ \vdots \\ \mathbf{F}_K^i \mathbf{G}_{kK}^i \end{bmatrix}, \forall k \in \mathcal{K}. \tag{3.33}$$

It can be verified that the optimal estimators $\hat{\mathbf{W}}_k$ are in the solution space defined by the parameterization (3.33). This follows from the fact that the columns of all $\hat{\mathbf{W}}_k$'s span the same R-dimensional column space, and hence are the same up to an $R \times R$ transformation matrix (see (3.17), (3.34)). This $R \times R$ transformation is represented by the $\mathbf{G}_{kn}$ matrix in (3.33).

If $\mathbf{W}_k^i$ is parameterized as (3.33), we have that $\widetilde{\mathbf{d}}_k^i = \widetilde{\mathbf{W}}_k^{iH} \tilde{\mathbf{y}}_k^i = \mathbf{W}_k^{iH} \mathbf{y}$. The goal of the GEVD-based DANSE algorithm is to iteratively update the parameters of (3.33) until $\mathbf{W}_k^i = \hat{\mathbf{W}}_k$, $\forall k \in \mathcal{K}$, or equivalently $\widetilde{\mathbf{d}}_k^i = \hat{\mathbf{d}}_k$, $\forall k \in \mathcal{K}$.

In the rest of this section, we show the actual relationship first between the network-wide GEVC matrix $\hat{\mathbf{X}}$ and the network-wide GEVD-based MWF (3.17) and then between the local GEVC matrix $\widetilde{\mathbf{X}}_k^i$ and the local GEVD-based MWF (3.24) at node $k \in \mathcal{K}$ in GEVD-based DANSE.

We define $\hat{\mathbb{X}} = [\hat{\mathbf{x}}_1 \dots \hat{\mathbf{x}}_R]$ as an $M \times R$ matrix where the columns are the principal GEVCs corresponding to the R largest GEVLs of $(\bar{\mathbf{R}}_{yy}, \bar{\mathbf{R}}_{nn})$ in (3.9), i.e., the first R columns of the network-wide matrix $\hat{\mathbf{X}} = \hat{\mathbf{Q}}^{-H}$. With this we can then rewrite (3.17) as

$$\hat{\mathbf{W}}_k = \left[\hat{\mathbb{X}} | \mathbf{0}_{M\times(M-R)}\right] (\mathbf{I}_M - \hat{\mathbf{L}}^{-1}) \hat{\mathbf{Q}}^H \hat{\mathbf{E}}_{d_k} = \hat{\mathbb{X}} \hat{\mathbf{\Psi}}_k \tag{3.34}$$

where $\hat{\mathbf{\Psi}}_k$ is a node-specific $R \times R$ transformation matrix defined as

$$\hat{\mathbf{\Psi}}_k = \left[\mathbf{I}_R | \mathbf{0}_{R\times(M-R)}\right] (\mathbf{I}_M - \hat{\mathbf{L}}^{-1}) \hat{\mathbf{Q}}^H \hat{\mathbf{E}}_{d_k}. \tag{3.35}$$

Since $\hat{\mathbb{X}}$ is independent of the node index k, (3.34) shows that all $\hat{\mathbf{W}}_k, \forall k \in \mathcal{K}$ share the same R-dimensional column space. For future purposes, we here further introduce the partitioning

$$\hat{\mathbb{X}} = \begin{bmatrix} \hat{\mathbb{X}}_1 \\ \vdots \\ \hat{\mathbb{X}}_K \end{bmatrix} \tag{3.36}$$

with $\hat{\mathbb{X}}_k$ denoting the $M_k \times R$ submatrix of $\hat{\mathbb{X}}$ corresponding to node k.

Similar to (3.17) and (3.34)-(3.35) the local GEVD-based MWF that are computed at node $k \in \mathcal{K}$ can be written as

$$\begin{aligned}
\widetilde{\mathbf{W}}_k^{i+1} &= (\widetilde{\mathbf{Q}}_k^i \widetilde{\mathbf{L}}_k^i \widetilde{\mathbf{Q}}_k^{iH})^{-1} \widetilde{\mathbf{Q}}_k^i \widetilde{\mathbf{\Delta}}_k^i \widetilde{\mathbf{Q}}_k^{iH} \widetilde{\mathbf{E}}_{d_k} \\
&= (\widetilde{\mathbf{Q}}_k^i)^{-H} (\widetilde{\mathbf{L}}_k^i)^{-1} \widetilde{\mathbf{\Delta}}_k^i \widetilde{\mathbf{Q}}_k^{iH} \widetilde{\mathbf{E}}_{d_k} \\
&= \left[\widetilde{\mathbb{X}}_k^i | \mathbf{0}_{P_k\times(P_k-R)}\right] (\widetilde{\mathbf{I}}_{P_k} - (\widetilde{\mathbf{L}}_k^i)^{-1}) \widetilde{\mathbf{Q}}_k^{iH} \widetilde{\mathbf{E}}_{d_k} \\
&= \widetilde{\mathbb{X}}_k^i \widetilde{\mathbf{\Psi}}_k^i
\end{aligned} \tag{3.37}$$

where $\widetilde{\mathbb{X}}_k^i$ is a $P_k \times R$ matrix containing the R principal GEVCs of $(\bar{\mathbf{R}}_{\tilde{y}_k\tilde{y}_k}^i, \bar{\mathbf{R}}_{\tilde{n}_k\tilde{n}_k}^i)$, i.e., the first R columns of $\widetilde{\mathbf{X}}_k^i$ in (3.20) and $\widetilde{\mathbf{\Psi}}_k^i$ is the $R \times R$ transformation matrix defined as

$$\widetilde{\mathbf{\Psi}}_k^i = \left[\mathbf{I}_R | \mathbf{0}_{R\times(P_k-R)}\right] (\widetilde{\mathbf{I}}_{P_k} - (\widetilde{\mathbf{L}}_k^i)^{-1}) \widetilde{\mathbf{Q}}_k^{iH} \widetilde{\mathbf{E}}_{d_k}. \tag{3.38}$$

Note that (3.37) states that $\widetilde{\mathbf{W}}_k^i$ is a transformed version of the principal GEVCs corresponding to the R largest GEVLs of $(\bar{\mathbf{R}}^i_{\tilde{y}_k\tilde{y}_k}, \bar{\mathbf{R}}^i_{\tilde{n}_k\tilde{n}_k})$.

Moreover, based on (3.37), the compression matrix $\mathbf{F}_k^i$ (see (3.25)) at each node $k \in \mathcal{K}$ can be written as

$$\mathbf{F}_k^i = [\mathbf{I}_{M_k} \ \mathbf{0}] \ \widetilde{\mathbb{X}}_k^i \widetilde{\mathbf{\Psi}}_k^i = \mathbb{X}_k^i \widetilde{\mathbf{\Psi}}_k^i \tag{3.39}$$

where $\mathbb{X}_k^i$ is defined as the first M_k rows of $\widetilde{\mathbb{X}}_k^i$.

3.4.5 Communication cost and computational complexity

In each iteration i of the GEVD-based DANSE algorithm, each node k transmits $R.N$ and receives $(K-1).R.N$ scalars, where N is the number of collected samples between two iterations. Assuming the same communication cost for the transmitter and the receiver modules at each node, then the total communication cost at node k at each iteration will be $K.R.N$, leading to a network communication cost of $K^2.R.N$. In a centralized realization however, the communication cost of each individual node and the communication cost of the whole network would be $K.M_k.N$ and $\sum_{k=1}^{K} K.M_k.N$, respectively. If $R < M_k$, then obviously the GEVD-based DANSE algorithm reduces the communication cost for node k with a factor M_k/R compared to the centralized realization (for each block of N samples). This dimensionality reduction then also results in a reduced per-node computational complexity compared to that of the centralized realization. Hence per update, the network-wide GEVD-based MWF has a complexity of $O(M^3)$, whereas the GEVD-based DANSE algorithm has a complexity of $O((M_q + R.(K-1))^3)$ at the updating node q. When compared to the centralized realization, this reduction comes at the cost of having a slower adaptation speed or tracking performance due to the iterative nature of the GEVD-based DANSE algorithm.

3.4.6 Convergence analysis

First note that if we set $\mathbf{W}_{kk}^i = \hat{\mathbb{X}}_k \hat{\mathbf{\Psi}}_k$, $\mathbf{F}_n^i = \hat{\mathbb{X}}_n \hat{\mathbf{\Psi}}_n$ and $\mathbf{G}_{kn}^i = \hat{\mathbf{\Psi}}_n^{-1} \hat{\mathbf{\Psi}}_k$, $\forall k, n \in \mathcal{K}$ in (3.33), then we obtain (3.34). This shows that the solution space of the GEVD-based DANSE algorithm, as described by the parameterization (3.33), indeed allows to obtain the network-wide GEVD-based MWF (3.17) as a special case. In this section we show that under some technical conditions, the proposed GEVD-based DANSE algorithm converges to the network-wide GEVD-based MWF (3.17), i.e., $\lim_{i\to\infty} \mathbf{W}_k^i = \hat{\mathbf{W}}_k$.

Remark 2. It is noted that, similar to [67, 68, 71], the convergence analysis assumes that the correlation matrices can be perfectly estimated using infinite observation windows, i.e., we assume in the sequel that $\bar{\mathbf{R}}_{(.)} = \mathbf{R}_{(.)}$ to make the theoretical analysis mathematically tractable. In practice, finite observation windows are used to estimate correlations, and therefore the convergence analysis should be viewed as an asymptotic analysis for which the approximation accuracy improves when larger observation windows N are used.

Theorem I. *Assume that* $\mathbf{R}_{yy}$ *is full rank. The GEVD-based DANSE algorithm converges for any initialization of its parameters to the network-wide GEVD-based MWF (3.17), i.e., when* $i \to \infty$ *and* $\forall k \in \mathcal{K}$, $\widetilde{\mathbf{d}}_k^i = \hat{\mathbf{d}}_k$ *and* $\mathbf{W}_k^i = \hat{\mathbf{W}}_k$, *where* $\mathbf{W}_k^i$ *is parameterized in (3.31)-(3.33).*

Proof outline: In order to prove the theorem we first postulate Lemma I, which describes an invariance-property of the GEVD under a joint congruence transformation of the defining matrix pair. Then, be it only for the sake of an easier exposition, we introduce a new 'virtual' algorithm (VA) for which the convergence analysis is more tractable. We prove convergence of this VA to the GEVCs of $\mathbf{R}_{yy}$ and $\mathbf{R}_{nn}$. Finally we show that convergence of the GEVD-based DANSE algorithm to the network-wide GEVD-based MWF follows jointly from convergence of the VA and the invariance-property of the GEVD proven in Lemma I.

Proof of Theorem I: We first consider the following Lemma:

Lemma I: *Let* $\mathbf{V}$ *denote the matrix containing all the GEVCs of the matrix pair* $(\mathbf{A}, \mathbf{B}) \in \mathbb{C}^{m \times m}$ *and let* $\mathbf{\Pi}$ *denote the diagonal matrix containing the corresponding GEVLs, with the assumption that all the GEVLs in* $\mathbf{\Pi}$ *are distinct and the columns of* $\mathbf{V}$ *are scaled such that* $\mathbf{V}^H\mathbf{B}\mathbf{V} = \mathbf{I}_m$. *Consider an invertible matrix* $\mathbf{J} \in \mathbb{C}^{m \times m}$ *such that* $\overline{\mathbf{A}} = \mathbf{J}\mathbf{A}\mathbf{J}^H$ *and* $\overline{\mathbf{B}} = \mathbf{J}\mathbf{B}\mathbf{J}^H$. *If* $\overline{\mathbf{V}}$ *and* $\overline{\mathbf{\Pi}}$ *denotes the matrix containing all the GEVCs and GEVLs of the transformed matrices* $(\overline{\mathbf{A}}, \overline{\mathbf{B}})$, *where the columns of* $\overline{\mathbf{V}}$ *is scaled such that* $\overline{\mathbf{V}}^H\overline{\mathbf{B}}\,\overline{\mathbf{V}} = \mathbf{I}_m$, *then* $\overline{\mathbf{V}} = \mathbf{J}^{-H}\mathbf{V}$ *and* $\overline{\mathbf{\Pi}} = \mathbf{\Pi}$.

Proof of Lemma I: From the definition of the GEVD for the matrix pair $(\mathbf{A}, \mathbf{B}) \in \mathbb{C}^{m \times m}$, it follows that [122]

$$\mathbf{A}\mathbf{V} = \mathbf{B}\mathbf{V}\mathbf{\Pi}, \ s.t. \ \mathbf{V}^H\mathbf{B}\mathbf{V} = \mathbf{I}_m. \tag{3.40}$$

Similarly, for the transformed matrices $(\overline{\mathbf{A}}, \overline{\mathbf{B}})$ we can write

$$\overline{\mathbf{A}}\,\overline{\mathbf{V}} = \overline{\mathbf{B}}\,\overline{\mathbf{V}}\,\overline{\mathbf{\Pi}}, \ \text{s.t.} \ \overline{\mathbf{V}}^H\overline{\mathbf{B}}\,\overline{\mathbf{V}} = \mathbf{I}_m \tag{3.41}$$

$$\Rightarrow \mathbf{J}\mathbf{A}\mathbf{J}^H\overline{\mathbf{V}} = \mathbf{J}\mathbf{B}\mathbf{J}^H\overline{\mathbf{V}}\,\overline{\mathbf{\Pi}}, \ \text{s.t.} \ \overline{\mathbf{V}}^H\mathbf{J}\mathbf{B}\mathbf{J}^H\,\overline{\mathbf{V}} = \mathbf{I}_m. \tag{3.42}$$

Left-multiplication of both sides of (3.42) with $\mathbf{J}^{-1}$ gives

$$\mathbf{A}\Big(\mathbf{J}^H\overline{\mathbf{V}}\Big) = \mathbf{B}\Big(\mathbf{J}^H\overline{\mathbf{V}}\Big)\overline{\mathbf{\Pi}}, \text{ s.t. } \Big(\overline{\mathbf{V}}^H\mathbf{J}\Big)\mathbf{B}\Big(\mathbf{J}^H\,\overline{\mathbf{V}}\Big) = \mathbf{I}_m. \tag{3.43}$$

Since all the GEVLs of $(\mathbf{A},\mathbf{B})$ are distinct and due to the normalization constraints, the GEVC matrices $\mathbf{V}$ and $\overline{\mathbf{V}}$ are unique. Therefore when comparing (3.43) with (3.40) it follows that $\overline{\mathbf{\Pi}} = \mathbf{\Pi}$ and that

$$\mathbf{V} = \mathbf{J}^H\overline{\mathbf{V}} \tag{3.44}$$

or alternatively, $\overline{\mathbf{V}} = \mathbf{J}^{-H}\mathbf{V}$, which proves the Lemma.

□

Remark 3. Lemma I assumes that all the GEVLs of $(\mathbf{A},\mathbf{B})$ are distinct and hence the GEVCs are unique. However in cases where this assumption is violated, e.g., when there exists one or more GEVLs with a multiplicity greater than one, we can only define a generalized eigenspace for each degenerate GEVL, i.e., the GEVCs can only be defined up to a transformation. Generally in this case we can write (3.44) as $\mathbf{V} = \mathbf{J}^H\overline{\mathbf{V}}\mathbf{P}$, where $\mathbf{P} = \text{diag}\{\mathbf{I},\mathbf{P}_1,\ldots,\mathbf{P}_q,\mathbf{I}\}$, with $\mathbf{P}_v$ an unknown $\rho\times\rho$ transformation matrix corresponding to the v-th degenerate GEVL with multiplicity ρ.

Now instead of directly considering the convergence of the GEVD-based DANSE algorithm to the network-wide GEVD-based MWF, for the sake of an easier explanation we first define and analyze the simpler VA as described in Table 3.2. In order to make a distinction, in the case of the VA we introduce the new notations $\underline{\mathbf{R}}^i_{\tilde{y}_q\tilde{y}_q}$, $\underline{\mathbf{R}}^i_{\tilde{n}_q\tilde{n}_q}$, $\underline{\widetilde{\mathbf{y}}}^i_q$, $\underline{\widetilde{\mathbf{X}}}^i_q$, $\underline{\widetilde{\mathbf{L}}}^i_q$, $\underline{\mathbb{X}}^i_q$, $\underline{\mathbf{F}}^i_k$ and $\underline{\mathbf{z}}^i_k$, replacing $\mathbf{R}^i_{\tilde{y}_q\tilde{y}_q}$, $\mathbf{R}^i_{\tilde{n}_q\tilde{n}_q}$, $\widetilde{\mathbf{y}}^i_q$, $\widetilde{\mathbf{X}}^i_q$, $\widetilde{\mathbf{L}}^i_q$, $\mathbb{X}^i_q$, $\mathbf{F}^i_k$ and $\mathbf{z}^i_k$, respectively, in the GEVD-based DANSE algorithm. Basically, the VA estimates[5] $\hat{\mathbb{X}}$ in (3.34), instead of estimating the network-wide GEVD-based MWF (3.17). It is noted that the definition of this VA is merely to facilitate the convergence proof of the GEVD-based DANSE algorithm.

Comparing Table 3.1 and Table 3.2 reveals that there exists some similarities and differences between the GEVD-based DANSE algorithm and the VA. For instance, in the VA the compressed signals are

$$\underline{\mathbf{z}}^i_k = \underline{\mathbf{F}}^{i\,H}_k\mathbf{y}_k = \underline{\mathbb{X}}^{i\,H}_k\mathbf{y}_k, \forall k\in\mathcal{K} \tag{3.49}$$

whereas in GEVD-based DANSE we have (based on (3.27) and (3.39))

$$\mathbf{z}^i_k = \mathbf{F}^{i\,H}_k\mathbf{y}_k = \widetilde{\mathbf{\Psi}}^{i\,H}_k\mathbb{X}^{i\,H}_k\mathbf{y}_k, \forall k\in\mathcal{K}. \tag{3.50}$$

[5]It is noted that the VA is actually equivalent to the distributed generalized eigenvector estimation (DACGEE) algorithm, which was introduced in [1], but without convergence proof. Here, we provide a theoretical convergence proof of this algorithm (see Theorem II).

Table 3.2: The VA to estimate the R principal network-wide GEVCs [1]

1. Set $i \leftarrow 0$, $q \leftarrow 1$, and initialize all $\underline{\mathbf{F}}_k^0$ and $\underline{\mathbb{X}}_k^0$, $\forall\, k \in \mathcal{K}$, with random entries.
2. Each node $k \in \mathcal{K}$ broadcasts N new R-channel compressed observations
$$\underline{\mathbf{z}}_k^i[iN+j] = \underline{\mathbf{F}}_k^{i\,H}\mathbf{y}_k[iN+j], \quad j = 1 \dots N. \tag{3.45}$$
3. At node q:
   - Compute $\bar{\underline{\mathbf{R}}}^i_{\tilde{y}_q\tilde{y}_q}$ and $\bar{\underline{\mathbf{R}}}^i_{\tilde{n}_q\tilde{n}_q}$ using the samples at times $iN+1$ up to $(i+1)N$.
   - Compute the columns of $\widetilde{\underline{\mathbb{X}}}_q^{i+1}$ as the R principal GEVCs of $(\bar{\underline{\mathbf{R}}}^i_{\tilde{y}_q\tilde{y}_q}, \bar{\underline{\mathbf{R}}}^i_{\tilde{n}_q\tilde{n}_q})$, normalized such that $\left(\widetilde{\underline{\mathbb{X}}}_q^{i+1}\right)^H \bar{\underline{\mathbf{R}}}^i_{\tilde{n}_q\tilde{n}_q}\widetilde{\underline{\mathbb{X}}}_q^{i+1} = \mathbf{I}_R$.
   - Partition $\widetilde{\underline{\mathbb{X}}}_q^{i+1}$ as
   $$\underline{\mathbb{X}}_q^{i+1} = \left[\mathbf{I}_{M_k}\ \mathbf{O}\right]\widetilde{\underline{\mathbb{X}}}_q^{i+1} \tag{3.46}$$
   $$\underline{\mathbf{G}}_{-q} = \left[\mathbf{O}\ \mathbf{I}_{R(K-1)}\right]\widetilde{\underline{\mathbb{X}}}_q^{i+1} \tag{3.47}$$
   and update the compression matrix as $\underline{\mathbf{F}}_q^{i+1} = \underline{\mathbb{X}}_q^{i+1}$ and broadcast $\underline{\mathbf{G}}_{-q} = \left[\underline{\mathbf{G}}_1^T \cdots \underline{\mathbf{G}}_{(q-1)}^T\ \underline{\mathbf{G}}_{(q+1)}^T \cdots \underline{\mathbf{G}}_K^T\right]^T$ to all the other nodes.
4. Each node $k \in \mathcal{K}\backslash\{q\}$ updates
$$\underline{\mathbb{X}}_k^{i+1} = \underline{\mathbb{X}}_k^i\underline{\mathbf{G}}_k. \tag{3.48}$$
5. $i \leftarrow i+1$ and $q \leftarrow (q \mod K)+1$ and return to step 2.

Hence the collected observations at the updating node q are different in both cases, i.e., $\widetilde{\mathbf{y}}_q^i = [\mathbf{y}_q^T\ \mathbf{z}_{-q}^{i\,T}]^T$ in the GEVD-based DANSE algorithm while $\widetilde{\underline{\mathbf{y}}}_q^i = [\mathbf{y}_q^T\ \underline{\mathbf{z}}_{-q}^{i\,T}]^T$ in the VA, leading to different correlation matrices $(\mathbf{R}^i_{\tilde{y}_q\tilde{y}_q}, \mathbf{R}^i_{\tilde{n}_q\tilde{n}_q})$ and $(\underline{\mathbf{R}}^i_{\tilde{y}_q\tilde{y}_q}, \underline{\mathbf{R}}^i_{\tilde{n}_q\tilde{n}_q})$, respectively. However note that in both algorithms at the updating node q the first M_q rows of $\widetilde{\mathbf{y}}_q^i$ and $\widetilde{\underline{\mathbf{y}}}_q^i$ are the same, i.e., they both contain node q's own sensor signals $\mathbf{y}_q$, whereas the other part is the same up to an $R \times R$ transformation, if $\mathbb{X}_k^i$ would be the same as $\underline{\mathbb{X}}_k^i$ in iteration i (we will exploit this fact later). Furthermore there is another major difference in the two algorithms that originates from the additional broadcasting of $\underline{\mathbf{G}}_{-q}$ in the VA (Step 3 of Table 3.2), which is required to perform the updates that take place at the other nodes $k \in \mathcal{K}\backslash q$ (Step 4 of Table 3.2), whereas this is not required in the GEVD-based DANSE algorithm. Despite all the aforementioned differences, we will later link both algorithms to each other, such that the convergence

properties of the VA can be transferred to the GEVD-based DANSE algorithm.

Let $\underline{\mathbb{X}}^i$ be the concatenation of the estimation variables $\underline{\mathbb{X}}_k^i, \forall k \in \mathcal{K}$ in the VA, i.e.,

$$\underline{\mathbb{X}}^i \triangleq \begin{bmatrix} \underline{\mathbb{X}}_1^i \\ \vdots \\ \underline{\mathbb{X}}_K^i \end{bmatrix}. \tag{3.51}$$

Moreover we introduce the notations $\underline{\widetilde{\mathbb{L}}}_q^i$, $\widetilde{\mathbb{L}}_q^i$, $\hat{\mathbb{L}}$ as the $R \times R$ submatrix that contains the R largest GEVLs of $\underline{\widetilde{\mathbf{L}}}_q^i$, $\widetilde{\mathbf{L}}_q^i$ and $\hat{\mathbf{L}}$, respectively. We then state the following convergence theorem for the VA

Theorem II. *In the VA algorithm (Table 3.2), the concatenated matrix $\underline{\mathbb{X}}^i$ converges to the matrix $\hat{\mathbb{X}}$ containing the R principal network-wide GEVCs of $(\mathbf{R}_{yy}, \mathbf{R}_{nn})$, i.e., $\lim_{i\to\infty} \underline{\mathbb{X}}^i = \hat{\mathbb{X}}$. Moreover $\lim_{i\to\infty} \underline{\widetilde{\mathbb{L}}}_k^i = \hat{\mathbb{L}}$, $\forall k \in \mathcal{K}$.*

Proof: See Appendix B.1.

Theorem II states that the stacked matrix $\underline{\mathbb{X}}^i$ defined in (3.51) converges to the network-wide GEVCs in $\hat{\mathbb{X}}$ when using the VA, i.e., $\lim_{i\to\infty} \underline{\mathbb{X}}^i = \hat{\mathbb{X}}$.

In order to clarify the link between the VA and the GEVD-based DANSE algorithm, we first study how a column transformation of the compression matrix $\underline{\mathbf{F}}_k^i$ (or equivalently of the $\underline{\mathbf{z}}_k^i$ signals) at all nodes $k \in \mathcal{K}\backslash q$ with an $R \times R$ node-specific matrix $\mathbf{M}_k^i$ affects the dynamics of the VA, i.e., when the compression matrix $\underline{\mathbf{F}}_k^i = \underline{\mathbb{X}}_k^i$ at iteration i is replaced by

$$\underline{\mathbf{F}}_k^{i,new} = \underline{\mathbb{X}}_k^i \mathbf{M}_k^i, \ \forall\, k \in \mathcal{K}\backslash q. \tag{3.52}$$

As a result, the new correlation matrices, namely $\underline{\mathbf{R}}_{\tilde{y}_q\tilde{y}_q}^{i,new}$ and $\underline{\mathbf{R}}_{\tilde{y}_q\tilde{y}_q}^{i,new}$, at the updating node q can be related to the VA case in the form of a joint matrix congruence relation:

$$\underline{\mathbf{R}}_{\tilde{y}_q\tilde{y}_q}^{i,new} = \underline{\widetilde{\mathbf{J}}}_q^i \, \underline{\mathbf{R}}_{\tilde{y}_q\tilde{y}_q}^i \, \underline{\widetilde{\mathbf{J}}}_q^{i\,H} \tag{3.53}$$

$$\underline{\mathbf{R}}_{\tilde{n}_q\tilde{n}_q}^{i,new} = \underline{\widetilde{\mathbf{J}}}_q^i \, \underline{\mathbf{R}}_{\tilde{n}_q\tilde{n}_q}^i \, \underline{\widetilde{\mathbf{J}}}_q^{i\,H} \tag{3.54}$$

where

$$\underline{\widetilde{\mathbf{J}}}_q^i = \left[\begin{array}{c|c|c} \mathbf{I}_{M_q} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{M}_{<q}^i & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{M}_{>q}^i \end{array}\right] \tag{3.55}$$

with[6]

$$\mathbf{M}^i_{<q} \triangleq \text{Blkdiag}(\mathbf{M}_1^{i\,H}, \ldots, \mathbf{M}_{(q-1)}^{i\,H}) \tag{3.56}$$

$$\mathbf{M}^i_{>q} \triangleq \text{Blkdiag}(\mathbf{M}_{(q+1)}^{i\,H}, \ldots, \mathbf{M}_K^{i\,H}) \tag{3.57}$$

where Blkdiag(.) is an operator that generates a block diagonal matrix from the matrices in its argument. Based on the result of Lemma I in (3.44), and under a similar assumption as in (3.26), we conclude that at the updating node q (where the GEVD is performed in Step 3 of Table 3.2)

$$\widetilde{\underline{\mathbb{X}}}_q^{i+1,new} = \big(\widetilde{\underline{\mathbf{J}}}_q^i\big)^{-H} \widetilde{\underline{\mathbb{X}}}_q^{i+1} \;\Rightarrow\; \underline{\mathbb{X}}_q^{i+1,new} = \underline{\mathbb{X}}_q^{i+1} \tag{3.58}$$

i.e., the first M_q rows of $\widetilde{\underline{\mathbb{X}}}_q^{i+1,new}$ and $\widetilde{\underline{\mathbb{X}}}_q^{i+1}$ will be identical and hence $\underline{\mathbb{X}}_q^{i+1}$ will not be affected by the transformations (3.52). In fact (3.58) shows the invariance property of the GEVD, which is indeed of great importance and verifies that an $R \times R$ node-specific transformation of the compression matrix $\underline{\mathbf{F}}_k^i$ at all nodes $k \in \mathcal{K}\backslash q$ (at iteration i) has no impact on the first M_q rows of $\widetilde{\underline{\mathbb{X}}}_q^{i+1}$ updated at node q from the GEVD for the *next* iteration (Step 3 of Table 3.2).

It is now again reiterated that the first M_q rows of $\widetilde{\mathbf{y}}_q^i$ and $\widetilde{\underline{\mathbf{y}}}_q^i$ at the updating node $q \in \mathcal{K}$ are the same in both the GEVD-based DANSE algorithm and the VA, while they only differ in the part corresponding to the received compressed signals $\mathbf{z}_{-q}^i$ and $\underline{\mathbf{z}}_{-q}^i$ (compare Table 3.1 and Table 3.2). Since in the GEVD-based DANSE algorithm, the compression matrix at node k is the same as $\mathbb{X}_k^i$ up to the $R \times R$ transformation $\widetilde{\mathbf{\Psi}}_k^i$ (see (3.39)), it turns out from (3.58) that at iteration i, we have $\underline{\mathbb{X}}_q^i = \mathbb{X}_q^i$. Now using an induction argument and based on (3.58), when the two algorithms are initialized with the same values, it can be concluded that the VA and the GEVD-based DANSE algorithm will always use the same compression matrices up to an $R \times R$ transformation on the columns, i.e., their column space will always be the same over all iterations. More formally, we will have that $\underline{\mathbb{X}}_q^i = \mathbb{X}_q^i$, $\forall i \in \mathbb{N}$, and from (3.39) that $\text{Colspace}(\mathbf{F}_q^i) = \text{Colspace}(\underline{\mathbb{X}}_q^i)$, $\forall i \in \mathbb{N}$. With this and comparing (3.52) and (3.39), we can link the correlation matrices in the VA and the GEVD-based DANSE algorithm as follows (see also (3.53)-(3.55))

$$\mathbf{R}^i_{\tilde{y}_q \tilde{y}_q} = \widetilde{\mathbf{J}}_q^i \, \underline{\mathbf{R}}^i_{\tilde{y}_q \tilde{y}_q} \, \widetilde{\mathbf{J}}_q^{i\,H} \tag{3.59}$$

$$\mathbf{R}^i_{\tilde{n}_q \tilde{n}_q} = \widetilde{\mathbf{J}}_q^i \, \underline{\mathbf{R}}^i_{\tilde{n}_q \tilde{n}_q} \, \widetilde{\mathbf{J}}_q^{i\,H} \tag{3.60}$$

[6] It is noted that the diagonal blocks are not square here, i.e., in this case Blkdiag(.) is not truly a block-diagonal matrix in the strict sense.

where the transformation matrix $\widetilde{\mathbf{J}}_q^i$ is defined as

$$\widetilde{\mathbf{J}}_q^i = \left[\begin{array}{c|c|c} \mathbf{I}_{M_q} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \boldsymbol{\Psi}_{<q}^i & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \boldsymbol{\Psi}_{>q}^i \end{array}\right] \tag{3.61}$$

with

$$\boldsymbol{\Psi}_{<q}^i \triangleq \text{Blkdiag}((\widetilde{\boldsymbol{\Psi}}_1^i)^H, \ldots, (\widetilde{\boldsymbol{\Psi}}_{(q-1)}^i)^H) \tag{3.62}$$

$$\boldsymbol{\Psi}_{>q}^i \triangleq \text{Blkdiag}((\widetilde{\boldsymbol{\Psi}}_{(q+1)}^i)^H, \ldots, (\widetilde{\boldsymbol{\Psi}}_{(K)}^i)^H). \tag{3.63}$$

Now based on Lemma I we can write

$$\widetilde{\mathbf{X}}_q^i = (\widetilde{\mathbf{J}}_q^i)^{-H} \underline{\widetilde{\mathbf{X}}}_q^i \tag{3.64}$$

and hence $\widetilde{\mathbb{X}}_q^i = (\widetilde{\mathbf{J}}_q^i)^{-H} \underline{\widetilde{\mathbb{X}}}_q^i$. This is indeed the key result that links the VA in Table 3.2 with the GEVD-based DANSE algorithm in Table 3.1. The relationship (3.64) then allows to rewrite (3.37) as

$$\widetilde{\mathbf{W}}_q^{i+1} = \left(\widetilde{\mathbf{J}}_q^i\right)^{-H} \underline{\widetilde{\mathbb{X}}}_q^i \widetilde{\boldsymbol{\Psi}}_q^i. \tag{3.65}$$

In order to prove the convergence of (3.65), all ingredients of it must converge as $i \to \infty$. So far, based on Theorem II, we only know that $\underline{\widetilde{\mathbb{X}}}_q^i, \forall q \in \mathcal{K}$ converges. Hence the next step is to verify that $\widetilde{\boldsymbol{\Psi}}_q^i, \forall q \in \mathcal{K}$ converges. Based on the definition (3.38), this requires that the first R diagonal elements of $\widetilde{\mathbf{L}}_q^i$, i.e., $\widetilde{\mathbb{L}}_q^i$, as well as the first R rows of $\widetilde{\mathbf{Q}}_q^{iH} \widetilde{\mathbf{E}}_{d_q}, \forall q \in \mathcal{K}$ converge.

Similar to (3.36), we define $\hat{\mathbb{Q}}$ as the first R columns of the network-wide matrix $\hat{\mathbf{Q}} = \hat{\mathbf{X}}^{-H}$ with the partitioning

$$\hat{\mathbb{Q}} \triangleq \hat{\mathbf{Q}} \left[\mathbf{I}_R \ \mathbf{0}\right]^T \triangleq \begin{bmatrix} \hat{\mathbb{Q}}_1 \\ \vdots \\ \hat{\mathbb{Q}}_K \end{bmatrix}. \tag{3.66}$$

Moreover we define a $M_k \times R$ submatrix $\mathbb{Q}_k^i$ as the first M_k rows and the first R columns of $\widetilde{\mathbf{Q}}_k^i$. The concatenation of all $\mathbb{Q}_k^i$ is defined as $\mathbb{Q}^i$, i.e.[7],

$$\mathbb{Q}_k^i \triangleq \left[\mathbf{I}_{M_k} \ \mathbf{0}\right] \widetilde{\mathbf{Q}}_k^i \left[\mathbf{I}_R \ \mathbf{0}\right]^T, \ \mathbb{Q}^i \triangleq \begin{bmatrix} \mathbb{Q}_1^i \\ \vdots \\ \mathbb{Q}_K^i \end{bmatrix}. \tag{3.67}$$

[7] Corresponding notations are also considered in the VA, e.g, $\underline{\mathbb{Q}}_k^i$ and $\underline{\mathbb{Q}}^i$.

Since $\widetilde{\mathbf{X}}_q^i = (\widetilde{\mathbf{Q}}_q^i)^{-H}$ and $\underline{\widetilde{\mathbf{X}}}_q^i = (\underline{\widetilde{\mathbf{Q}}}_q^i)^{-H}$, from (3.64) it follows that

$$\widetilde{\mathbf{Q}}_q^i = \widetilde{\mathbf{J}}_q^i \underline{\widetilde{\mathbf{Q}}}_q^i. \tag{3.68}$$

This means that the first M_q rows of $\widetilde{\mathbb{Q}}_q^i$ are also independent of the transformations that the matrix $\widetilde{\mathbf{J}}_q^i$ in (3.61) applies in the GEVD-based DANSE algorithm, i.e, $\mathbb{Q}_q^i = \underline{\mathbb{Q}}_q^i$. Moreover, it has been shown in [120] that the concatenation of the local signal subspace estimates $\underline{\mathbb{Q}}_k^i$ obtained based on the inversion of $\underline{\widetilde{\mathbf{X}}}_k^i$ at each node $k \in \mathcal{K}$, i.e, the matrix $\underline{\mathbb{Q}}^i$, converges to $\hat{\mathbb{Q}}$, or in particular $\lim_{i\to\infty} \underline{\mathbb{Q}}_k^i = \hat{\mathbb{Q}}_k, \forall k \in \mathcal{K}$ (see (3.66)-(3.67)). Based on (3.68) this also holds in the GEVD-based DANSE algorithm, i.e., $\lim_{i\to\infty} \mathbb{Q}^i = \hat{\mathbb{Q}}, \forall k \in \mathcal{K}$, or likewise $\lim_{i\to\infty} \mathbb{Q}_k^i = \hat{\mathbb{Q}}_k$. Finally since $R < M_k, \forall k \in \mathcal{K}$, it follows that after convergence we have

$$\lim_{i\to\infty} \widetilde{\mathbb{Q}}_k^{iH} \widetilde{\mathbf{E}}_{d_k} = \hat{\mathbb{Q}}_k^H \hat{\mathbf{E}}_{d_k}, \ \forall k \in \mathcal{K}. \tag{3.69}$$

By relying on Theorem II (see also Theorem A.III in Appendix B.1), we have that $\lim_{i\to\infty} \underline{\widetilde{\mathbb{L}}}_k^i = \hat{\mathbb{L}}$, $\forall k \in \mathcal{K}$. Additionally based on (3.59)-(3.61) and the result of Lemma I, it follows that we can further link the two algorithms in terms of the local GEVLs such that $\widetilde{\mathbf{L}}_k^i = \underline{\widetilde{\mathbf{L}}}_k^i$, $\forall k \in \mathcal{K}$. With this we can conclude that $\widetilde{\mathbb{L}}_k^i$, $\forall k \in \mathcal{K}$ also converges, i.e.,

$$\lim_{i\to\infty} \widetilde{\mathbb{L}}_k^i = \hat{\mathbb{L}}, \ \forall k \in \mathcal{K}. \tag{3.70}$$

Therefore considering the definitions of the $R \times R$ matrices $\hat{\mathbf{\Psi}}_k$ and $\widetilde{\mathbf{\Psi}}_k^i$ in (3.35) and (3.38), we can readily conclude from (3.69) and (3.70) that

$$\widetilde{\mathbf{\Psi}}_k^\infty \triangleq \lim_{i\to\infty} \widetilde{\mathbf{\Psi}}_k^i = \hat{\mathbf{\Psi}}_k, \ \forall k \in \mathcal{K}. \tag{3.71}$$

Now plugging (3.71) and (3.61) into (3.65), and considering the fact that after convergence of the VA, we have that $\lim_{i\to\infty} \underline{\widetilde{\mathbb{X}}}_k^i = \left[\hat{\mathbb{X}}_k^T \mathbf{I}_R \dots \mathbf{I}_R\right]^T, \forall k \in \mathcal{K}$

(see also (B.8) in Appendix B.1), gives

$$\lim_{i\to\infty} \widetilde{\mathbf{W}}_k^i = \begin{bmatrix} \hat{\mathbb{X}}_k \\ \hline (\widetilde{\mathbf{\Psi}}_1^\infty)^{-1} \\ \vdots \\ (\widetilde{\mathbf{\Psi}}_{(k-1)}^\infty)^{-1} \\ (\widetilde{\mathbf{\Psi}}_{(k+1)}^\infty)^{-1} \\ \vdots \\ (\widetilde{\mathbf{\Psi}}_K^\infty)^{-1} \end{bmatrix} \hat{\mathbf{\Psi}}_k, \ \forall k \in \mathcal{K}. \tag{3.72}$$

Comparing (3.72) with (3.32) we have that $\lim_{i\to\infty} \mathbf{G}_{kn}^i = (\widetilde{\mathbf{\Psi}}_n^\infty)^{-1}\hat{\mathbf{\Psi}}_k, \forall k \in \mathcal{K}, \forall n \in \mathcal{K}\backslash k$. Hence the network-wide parameterization (3.33) at each node $k \in \mathcal{K}$ can be written as

$$\lim_{i\to\infty} \mathbf{W}_k^i = \begin{bmatrix} \mathbf{F}_1^i \mathbf{G}_{k1}^i \\ \vdots \\ \mathbf{W}_{kk}^i \\ \vdots \\ \mathbf{F}_K^i \mathbf{G}_{kK}^i \end{bmatrix} = \begin{bmatrix} \hat{\mathbb{X}}_1 \widetilde{\mathbf{\Psi}}_1^\infty (\widetilde{\mathbf{\Psi}}_1^\infty)^{-1} \hat{\mathbf{\Psi}}_k \\ \vdots \\ \hat{\mathbb{X}}_k \hat{\mathbf{\Psi}}_k \\ \vdots \\ \hat{\mathbb{X}}_K \widetilde{\mathbf{\Psi}}_K^\infty (\widetilde{\mathbf{\Psi}}_K^\infty)^{-1} \hat{\mathbf{\Psi}}_k \end{bmatrix} = \hat{\mathbb{X}} \hat{\mathbf{\Psi}}_k. \tag{3.73}$$

Since the righthand side of (3.73) is equal to the righthand side of (3.34), it follows that, after the convergence of all nodes, $\lim_{i\to\infty} \mathbf{W}_k^i = \hat{\mathbf{W}}_k, \forall k \in \mathcal{K}$ and hence $\widetilde{\mathbf{d}}_k^i = \hat{\mathbf{d}}_k, \forall k \in \mathcal{K}$, which proves Theorem I.

□

Remark 4. It is noted that the original DANSE algorithm in [67] requires strict conditions on the data model, i.e., the node-specific desired signals $\mathbf{d}_k$ should share a common latent signal subspace of dimension S, where then the compression matrices are $(M_k \times S)$-dimensional matrices. However, in the GEVD-based DANSE algorithm, the rank-R approximation effectively redefines (imposes) a common latent signal subspace of dimension R, where then the compression matrices are $(M_k \times R)$-dimensional matrices. This means that although the existence of the common signal subspace that contains all the $\mathbf{d}_k$'s motivates the low-rank approximation of $\mathbf{R}_{ss}$ (see (3.8)), there is no need to strictly assume this conditions on the desired signals $\mathbf{d}_k$ in the convergence Theorem I in order to let the GEVD-based DANSE algorithm converge to the network-wide GEVD-based MWF. This will be further verified in the simulation section (see Fig. 3.2 and Fig. 3.5 in Section 3.5), where it is observed that, even under a data model mismatch (i.e., when $R < S$), the algorithm still converges

to the network-wide estimator, while a better noise reduction is still observed compared to the original DANSE algorithm (see Fig. 3.4 and Fig. 3.5).

Remark 5. The convergence analysis shows that there is a fundamental link between the proposed GEVD-based DANSE algorithm and the DACGEE algorithm proposed in [1] (see Footnote 5), despite the fact that both algorithms are derived in a different framework, i.e., distributed node-specific signal estimation on the one hand, and distributed computation of the network-wide GEVD on the other hand. In retrospect however, this link between both algorithms may not be a complete surprise, since the MWF with GEVD-based rank-R approximation implicitly also computes a GEVD. Nevertheless, there are several differences between the algorithms (e.g. the transmission of the $\underline{\mathbf{G}}_{-q}$-coefficients), and the link between them is due to the following three non-trivial key properties which have been identified in Section 3.4.6:

- The invariance-property of the GEVD under a joint congruence transformation of the defining matrix pair (Lemma I).
- The fact that the compression matrices of the GEVD-based DANSE algorithm at each node k based on $\widetilde{\mathbf{W}}_k^i$ in (3.28) can be written as a node-specific $R \times R$ column transformation of $\mathbb{X}_k^i$ (see (3.39)).
- The fact that $\lim_{i\to\infty} \mathbb{Q}^i = \lim_{i\to\infty} \underline{\mathbb{Q}}^i = \hat{\mathbb{Q}}$ holds, which does not immediately follow from the convergence proof of DACGEE, as the latter only shows that $\lim_{i\to\infty} \underline{\mathbb{X}}^i = \hat{\mathbb{X}}$.

If one of these properties would not hold, the link between both algorithms would indeed be lost.

Remark 6. As discussed in Section 3.4.1 cases where $L > R$ may happen. For such cases, the network-wide GEVD-based MWF estimates the L-channel desired signal $\hat{\mathbf{d}}_k = \hat{\mathbf{W}}_k^H \mathbf{y}$ (see (3.17)). This can be partitioned as

$$\hat{\mathbf{d}}_k \triangleq \begin{bmatrix} \hat{\mathbf{d}}_k^R \\ \hat{\mathbf{d}}_k^{(L-R)} \end{bmatrix} = \left[\hat{\mathbf{W}}_k^R \ \hat{\mathbf{W}}_k^{(L-R)}\right]^H \mathbf{y} \tag{3.74}$$

where $\hat{\mathbf{d}}_k^R$ is estimated by the GEVD-based DANSE algorithm. Now based on (3.34), we can write $\hat{\mathbf{W}}_k^R = \hat{\mathbb{X}}\hat{\mathbf{\Psi}}_k^R$ and $\hat{\mathbf{W}}_k^{(L-R)} = \hat{\mathbb{X}}\hat{\mathbf{\Psi}}_k^{(L-R)}$, where $\hat{\mathbf{\Psi}}_k^R$ and $\hat{\mathbf{\Psi}}_k^{(L-R)}$ are $R \times R$ and $R \times (L-R)$ transformation matrices, respectively. It follows then that $\hat{\mathbf{W}}_k^{(L-R)} = \hat{\mathbf{W}}_k^R \hat{\mathbf{L}}_k$, with $\hat{\mathbf{L}}_k = (\hat{\mathbf{\Psi}}_k^R)^{-1}\hat{\mathbf{\Psi}}_k^{(L-R)}$ an $R \times (L-R)$ transformation matrix. This also implies that $\hat{\mathbf{d}}_k^{(L-R)} = \hat{\mathbf{L}}_k^H \hat{\mathbf{d}}_k^R$, which means that at each node $k \in \mathcal{K}$, the signal $\hat{\mathbf{d}}_k^{(L-R)}$ is merely a linear combination of

the channels of $\hat{\mathbf{d}}_k^R$ and hence, given the latter, the former can be computed accordingly. This property in fact allows each node $k \in \mathcal{K}$ in the GEVD-based DANSE algorithm to compute the extra $(L-R)$ channels of the desired signal based on the estimated R-channel signal $\widetilde{\mathbf{d}}_k^i$ in (3.30). The unknown $R \times (L-R)$ transformation matrix can then be computed using, e.g., an MWF, with $\widetilde{\mathbf{d}}_k^i$ as the inputs. Note that in practice however, one can alternatively select the $(L-R)$ columns in (3.24) that correspond to the $(L-R)$ channels in $\hat{\mathbf{d}}_k^{(L-R)}$, by changing $\widetilde{\mathbf{E}}_{d_k}$ accordingly. These columns can be used as an MWF with $\widetilde{\mathbf{y}}_k^i$ as an input, and will indeed have $\hat{\mathbf{d}}_k^{(L-R)}$ as an output (after convergence of the algorithm). This can be easily proven from (3.34)-(3.35).

3.5 Simulation results

In this section, we provide Monte-Carlo (MC) simulations, which allow us to further demonstrate the convergence and performance behavior of the GEVD-based DANSE algorithm in different conditions. A setup with different positions of nodes/sources and with different signals is created in each independent MC run. Each node k then collects observations of a different 15-channel stochastic signal $\mathbf{y}_k$, i.e., $M_k = 15$, $\forall k \in \mathcal{K}$. It is noted that all plots in this section show the median of the MSE measures over 200 MC runs and averaged over all nodes.

Two different mean squared error (MSE-) based performance measures are used to asses the behavior of the GEVD-based DANSE algorithm. The first measure evaluates the algorithm from the convergence perspective, i.e., it considers the MSE between the solutions of the network-wide GEVD-based MWF and those of the GEVD-based DANSE algorithm. This measure, namely MSE1, at each node k will be applied both to the estimated filters as well as the desired signals, i.e.,

$$\text{MSE1}_{W_k}^i = \frac{1}{MR}\|\mathbf{W}_k^i - \hat{\mathbf{W}}_k\|_F^2 \tag{3.75}$$

$$\text{MSE1}_{d_k}^i = \frac{1}{RN}\sum_{n=0}^{N-1}\|\widetilde{\mathbf{d}}_k^i(n) - \hat{\mathbf{d}}_k(n)\|_F^2 \tag{3.76}$$

with $\hat{\mathbf{d}}_k(n)$ the observation of $\hat{\mathbf{d}}_k$ at time n. Note that when the GEVD-based DANSE algorithm converges to the network-wide GEVD-based MWF, $\text{MSE1}_{W_k}^i$ and $\text{MSE1}_{d_k}^i$ will theoretically decrease to 0, i.e., to the machine precision in practice.

The second measure, namely MSE2, aims at evaluating the output performance of the GEVD-based DANSE algorithm from a noise reduction perspective. This

will further allow to compare the output performance of the GEVD-based DANSE algorithm with that of the original DANSE algorithm. We define the following MSE2s as the MSE between the estimated and true values of the desired signals for both the network-wide and GEVD-based DANSE output solutions:

$$\text{MSE2}_{\hat{d}_k} = \frac{1}{RN}\sum_{n=0}^{N-1} \|\hat{\mathbf{d}}_k(n) - \mathbf{d}_k(n)\|_F^2 \tag{3.77}$$

$$\text{MSE2}^i_{d_k} = \frac{1}{RN}\sum_{n=0}^{N-1} \|\widetilde{\mathbf{d}}^i_k(n) - \mathbf{d}_k(n)\|_F^2. \tag{3.78}$$

We first perform batch-mode simulations in Section 3.5.1, where the required statistics are computed over the full length signals. Note that in the batch version of the GEVD-based DANSE algorithm, we use the entire signal length to estimate the correlation matrices used in the algorithm. In Section 3.5.2, a more practical scenario with finite-window sensor signal observations is considered. Note that in this case each set of sensor signal observations is only broadcast once, i.e., subsequent iterations are performed over different observation sets.

3.5.1 Batch-mode simulations

In this section batch-mode simulations are performed, where in each MC run six localized point sources are present with equal powers (unless otherwise stated), from which $S = 3$ are treated as desired sources, and the other 3 sources treated as noise sources, leading to spatially correlated noise components over all sensors. The number of observations in these scenarios is set to $N = 15000$ samples. The $S = 3$ target sources have an ON-OFF behavior which are active over the intervals [1000 4000] and [6000 9000], while the noise sources are continuously active. The observations of the latent 3-channel signal $\check{\mathbf{s}}$, and the entries of the 15×3 steering matrix $\mathbf{A}_k$ (see (3.1)), $\forall k \in \mathcal{K}$ are both independently drawn from a uniform distribution over the interval $[-0.5; 0.5]$. The network-wide noise signal $\mathbf{n}$ is generated as $\mathbf{n} = \mathbf{B}\mathbf{v} + \mathbf{q}$, where $\mathbf{B}$ is the $15K \times 3$ steering matrix corresponding to noise sources, $\mathbf{v}$ contains the 3 noise source signals, and $\mathbf{q}$ contains the spatially uncorrelated white noise signals (sensor noise) where the power at each node is equal to 10% of the power of the mixed desired source signals as observed by the first sensor of the node. The observations of $\mathbf{v}$ and the entries of $\mathbf{B}$ are also drawn from a uniform distribution over the interval $[-0.5; 0.5]$. Note that in these experiments, in order to exclude the effects of overlapping so-called 'noise-only' and 'signal-plus-noise' segments, noise correlation matrices are always estimated over the perfect 'noise-only'

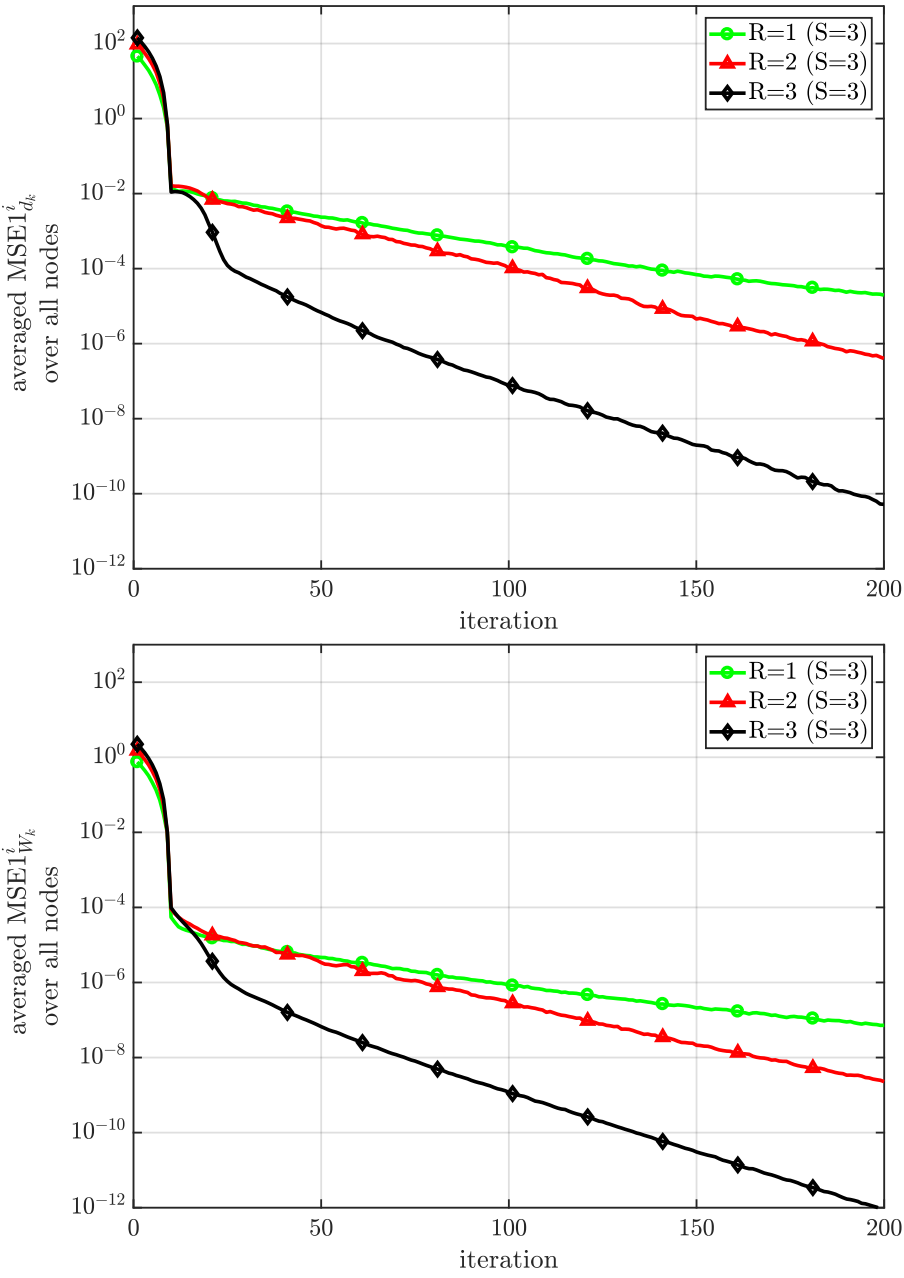


Figure 3.2: Convergence of the GEVD-based DANSE algorithm for different low-rank approximations R.

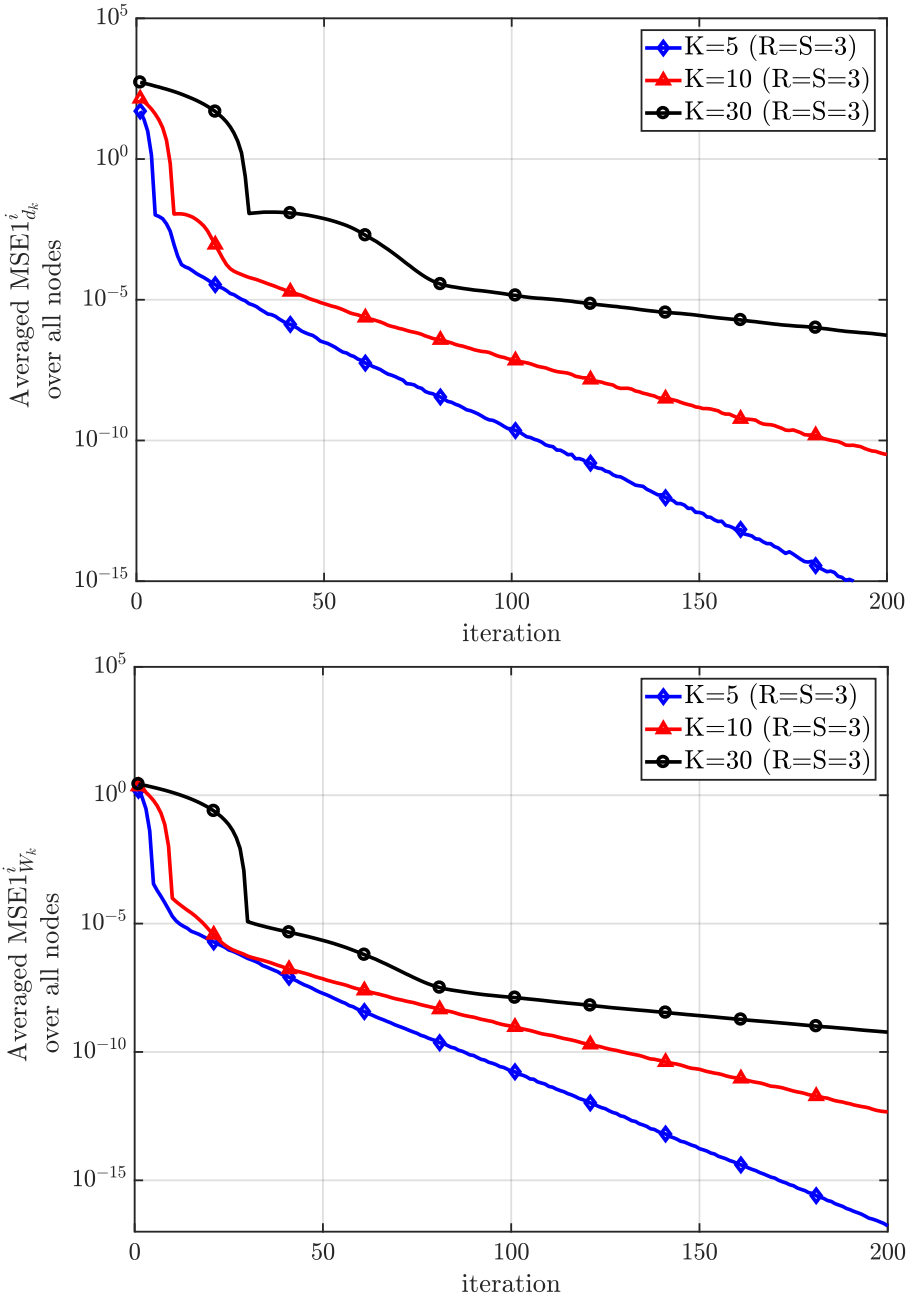


Figure 3.3: Convergence of the GEVD-based DANSE algorithm for different network sizes K.

segments, i.e., from the samples over the intervals [1 1000], [5000 6000] and [9000 15000].

We first consider the performance of the algorithm for different values of the rank approximation order R, both when $R = S$ and when $R < S$ (cases where S is either not known or wrongly estimated). Here we assume $K = 10$, i.e., $M = 150$. Fig.3.2 illustrates the convergence results for different values of R, i.e. $R = 3$, $R = 2$ and $R = 1$ when $S = 3$. In the upper part, the averaged $\text{MSE1}_{d_k}^i$ over all nodes is shown as a function of the different iterations of the GEVD-based DANSE algorithm. Similarly, the bottom part illustrates the averaged $\text{MSE1}_{W_k}^i$'s. First it is observed that for all three cases, the GEVD-based DANSE algorithm converges (with a random initialization of its parameters) to the network-wide GEVD-based MWF with the same value of R. Note that it has been observed that all these figures will eventually decrease to the machine precision values. It is noted that in the same scenario, the original DANSE algorithm would converge to the network-wide LMMSE-based MWF only if $R = S = 3$. It is also observed that among these cases, the case of $R = 3$ delivers the fastest convergence rate, followed by cases of $R = 2$ and $R = 1$. This can be explained by the larger number of degrees of freedom in each update step in the former case.

Fig. 3.3 shows the convergence of the GEVD-based DANSE algorithm in terms of both $\text{MSE1}_{d_k}^i$ and $\text{MSE1}_{W_k}^i$, for different network sizes, i.e., $K = 5$, $K = 10$ and $K = 30$. In this scenario we have $R = S = 3$. Since the updating is done in a sequential fashion, it is not surprising that the convergence speed decreases as the number of nodes increases.

In the rest of this section we will compare the noise reduction performance of the GEVD-based DANSE algorithm with the performance of the original DANSE algorithm.

We first compare the output of the two algorithms in scenarios with different noise powers. To achieve this, we define the overall input signal to noise ratio (iSNR) as the iSNR with respect to each individual desired source as observed at the first sensor of each node, averaged over all the desired sources. Similarly the overall output SNR (oSNR) is defined as the oSNR with respect to each individual desired source, averaged over all desired sources. In order to manipulate the overall iSNR, we will equally increase the power of the 3 noise sources in each of these scenarios. With this, we achieve simulation scenarios with overall iSNR values of 4.35 dB and -2.8 dB. Fig. 3.4 then compares the overall oSNR obtained from the GEVD-based DANSE ($R = 3$) and the original DANSE algorithms. Furthermore, the overall oSNRs of the network-wide GEVD-based MWF ($R = 3$) and the network-wide LMMSE-based MWF are added as a reference. These results demonstrate that, with the same

network communication cost, the GEVD-based DANSE algorithm outperforms the original DANSE algorithm in terms of the overall oSNR (see also Fig. 3.6).

Although an overall oSNR as a measure provides insights on the performance of the algorithms in terms of the amount of noise which has been removed, it does not reflect the amount of distortion or other imperfections on the desired signal estimates. Hence in order to better evaluate the output performance of the algorithms, we now consider the $\text{MSE2}_{d_k}^i$ measure defined in (3.78). Fig. 3.5 shows the averaged MSE2 over all $K = 10$ nodes (200 MC runs). Different values of R are considered in the case of the GEVD-based DANSE algorithm. Moreover the MSE2_{d_k} values of the corresponding network-wide MWF solutions are shown by dashed lines as a reference. This figure verifies again the significant difference between the GEVD-based DANSE algorithm and the original DANSE algorithm. Considering the same experiment setup, Fig. 3.6 shows the averaged MSE2_{d_k} as a function of the network communication cost (defined in Subsection 3.4.5) up to iteration i of the distributed algorithm, i.e., $K^2.R.N.i$. This figure shows that the GEVD-based DANSE algorithm with an underestimated rank ($R = 1$) still delivers a slightly better MSE2-based performance than the original DANSE algorithm, which also has a higher communication cost.

So far the nodes update their parameters in a sequential round-robin fashion, which may yield a slow convergence of the estimators, especially so when the number of nodes in the network is large. Note that the convergence time of this case increases linearly with the number of nodes. In [68], a modification of the original DANSE algorithm with simultaneously updating nodes has been described, where it was observed that a faster convergence and faster adaptation is obtained. We can use a similar simultaneous node-updating strategy for the GEVD-based DANSE algorithm, to investigate how the dynamics of the GEVD-based DANSE algorithm change in cases where the nodes are able to update simultaneously in each iteration. Note that the computational load of the latter will be higher in comparison with the former. Fig. 3.7 considers the convergence of the GEVD-based DANSE algorithm with $R = 3$ when nodes update simultaneously. The top part of this figure shows the averaged $\text{MSE1}_{d_k}^i$ and $\text{MSE1}_{W_k}^i$ over $K = 10$ nodes (200 MC runs). Furthermore, the bottom part of Fig. 3.7 compares the convergence speed of the GEVD-based DANSE algorithm with $R = 3$ for both cases where nodes update sequentially and simultaneously. These results verify that when nodes are able to update simultaneously, the algorithm converges significantly faster than the case of sequential updating. Note that this is stated here as an observation based on extensive simulation on this particular simulation scenario, but without a formal proof.

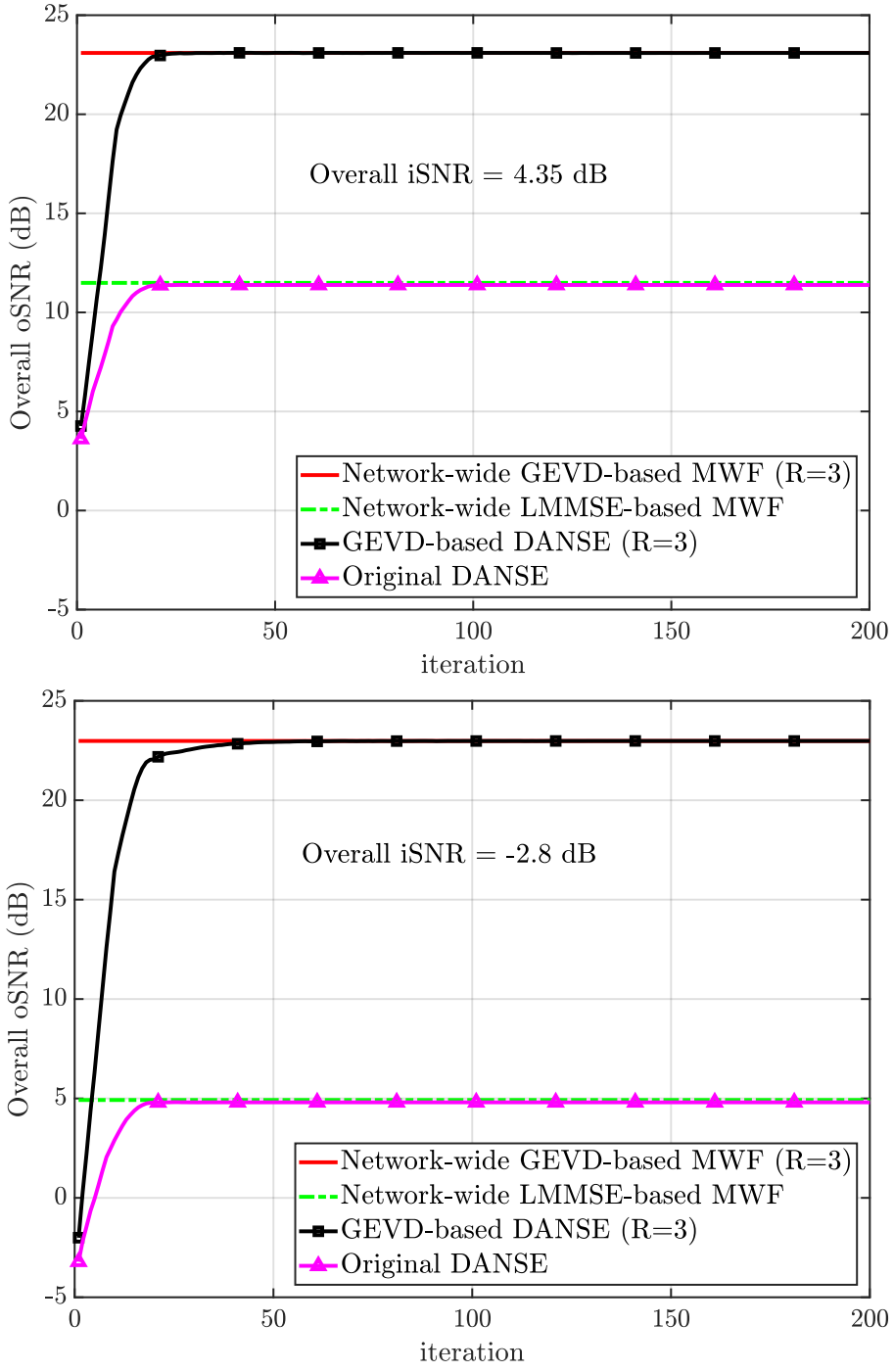


Figure 3.4: Comparison of overall oSNR between GEVD-based DANSE and original DANSE in scenarios with different iSNRs.

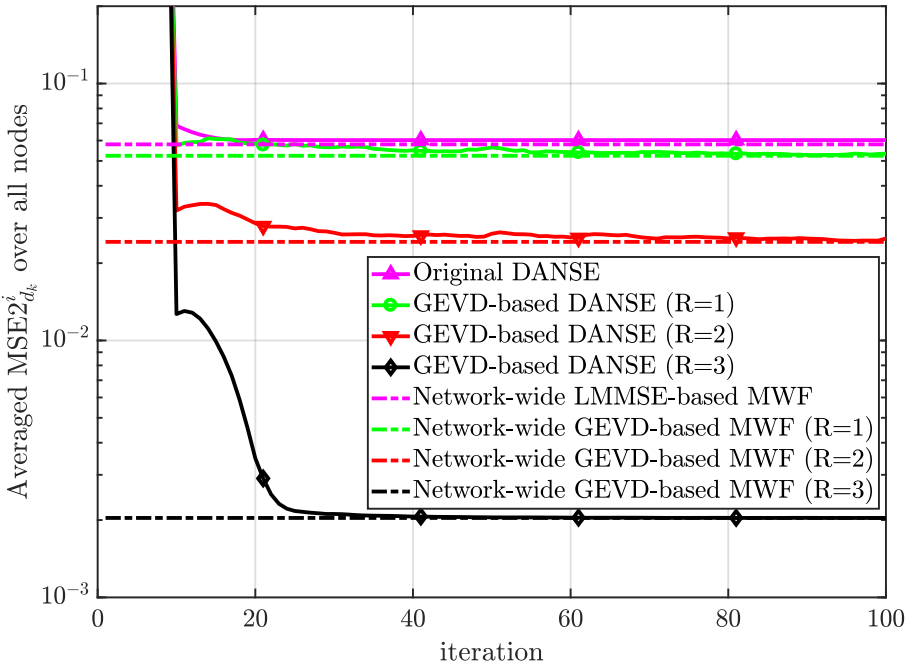


Figure 3.5: Comparison of MSE2$^i_{d_k}$ between GEVD-based DANSE with different values of R and original DANSE. Network-wide values are also added as a reference (dashed lines).

3.5.2 Finite-window simulations

In this section we evaluate the performance of the GEVD-based DANSE algorithm when the required statistics are estimated over finite windows of length N (instead of the full signal), where the window shifts over time in each iteration. This will result in larger estimation errors in the correlation matrices that are used in the algorithm. In this case, the theoretical convergence analysis only approximately hold as it does not take estimation errors in the correlation matrices into account. In practice, the window length N introduces a trade-off between the tracking performance and estimation errors in the correlation matrices, where the latter leads to a larger convergence offset. In this section signals with a full length of 100000 samples are generated where the desired source signals have an ON-OFF behavior with a duty cycle of 50%, with an active period of 1000 samples. Fig. 3.8 shows the convergence of the GEVD-based DANSE algorithm where the correlation matrices are estimated over different values of finite-window length N (200 MC runs and $R = S = 3$).

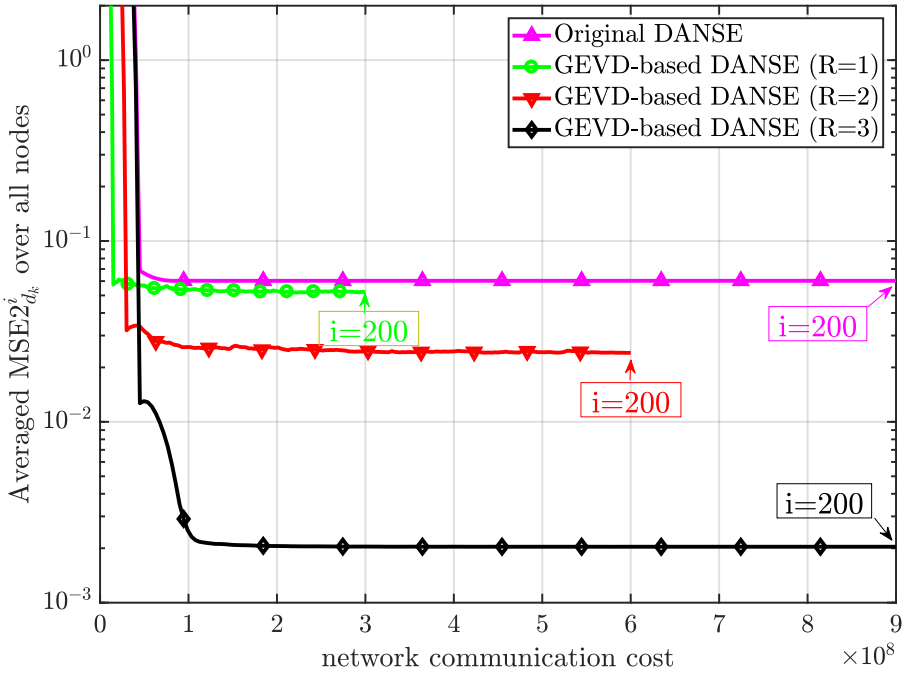


Figure 3.6: Comparison of network communication cost (total number of transmitted and received scalars in the network up to iteration i) between GEVD-based DANSE with different values of R and original DANSE. Number of algorithm iterations in all cases is $i = 200$.

The upper part shows the averaged MSE1$^i_{d_k}$ over all nodes as a function of the different iterations of the GEVD-based DANSE algorithm. The bottom part of Fig. 3.8 shows the MSE between the signal estimate of the GEVD-based DANSE and the network-wide GEVD-based MWF algorithm (both with finite window N), with respect to the signal estimate of the network-wide GEVD-based MWF in batch mode (computed over the full signal of $N = 100000$ samples). This figure verifies that, although in both the network-wide GEVD-based MWF and the GEVD-based DANSE algorithm the performance drops depending on the window length N, the discrepancy between them is negligible when N is chosen sufficiently large.

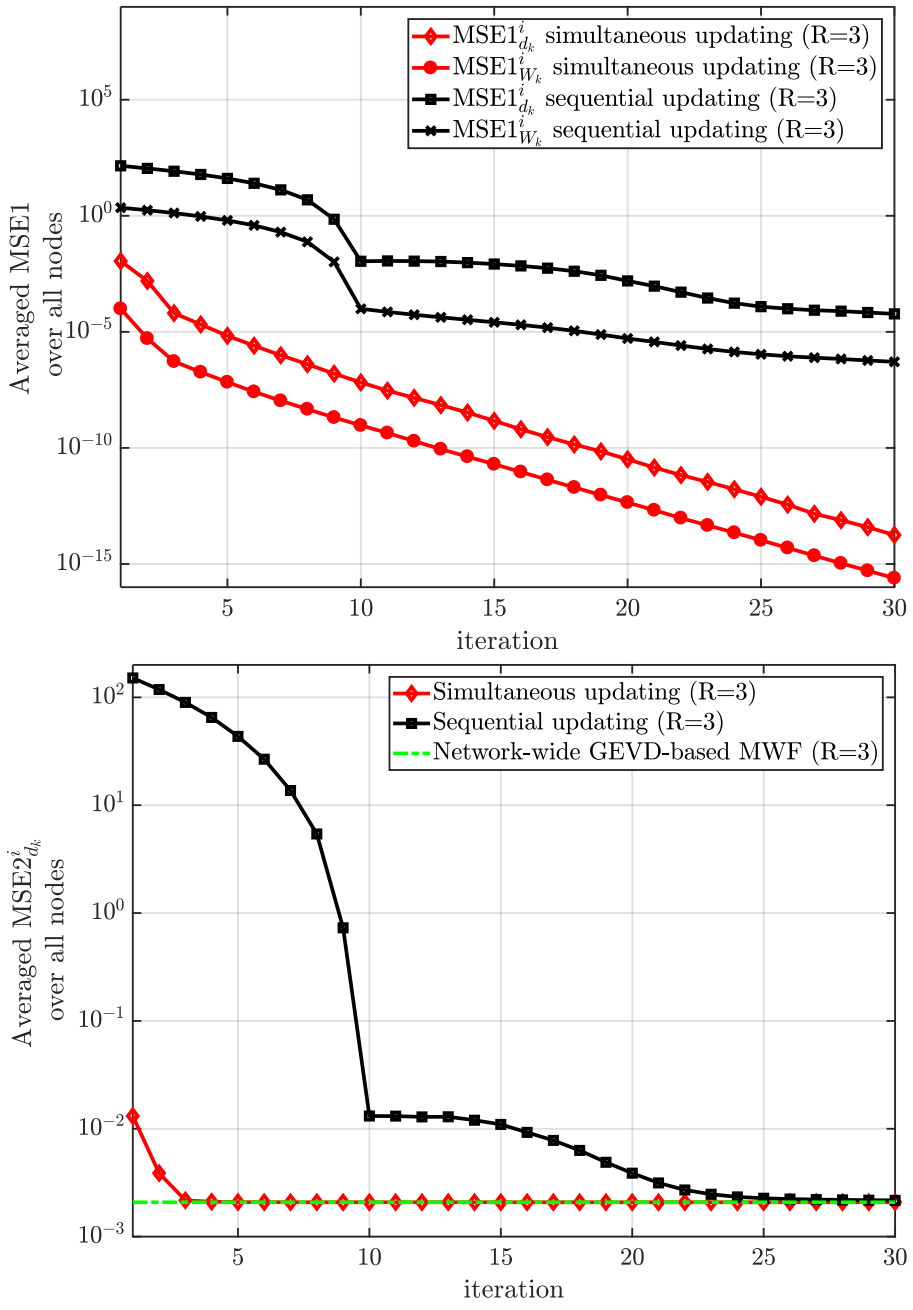


Figure 3.7: Comparison of GEVD-based DANSE ($R = 3$) between the cases where nodes update simultaneously and sequentially.

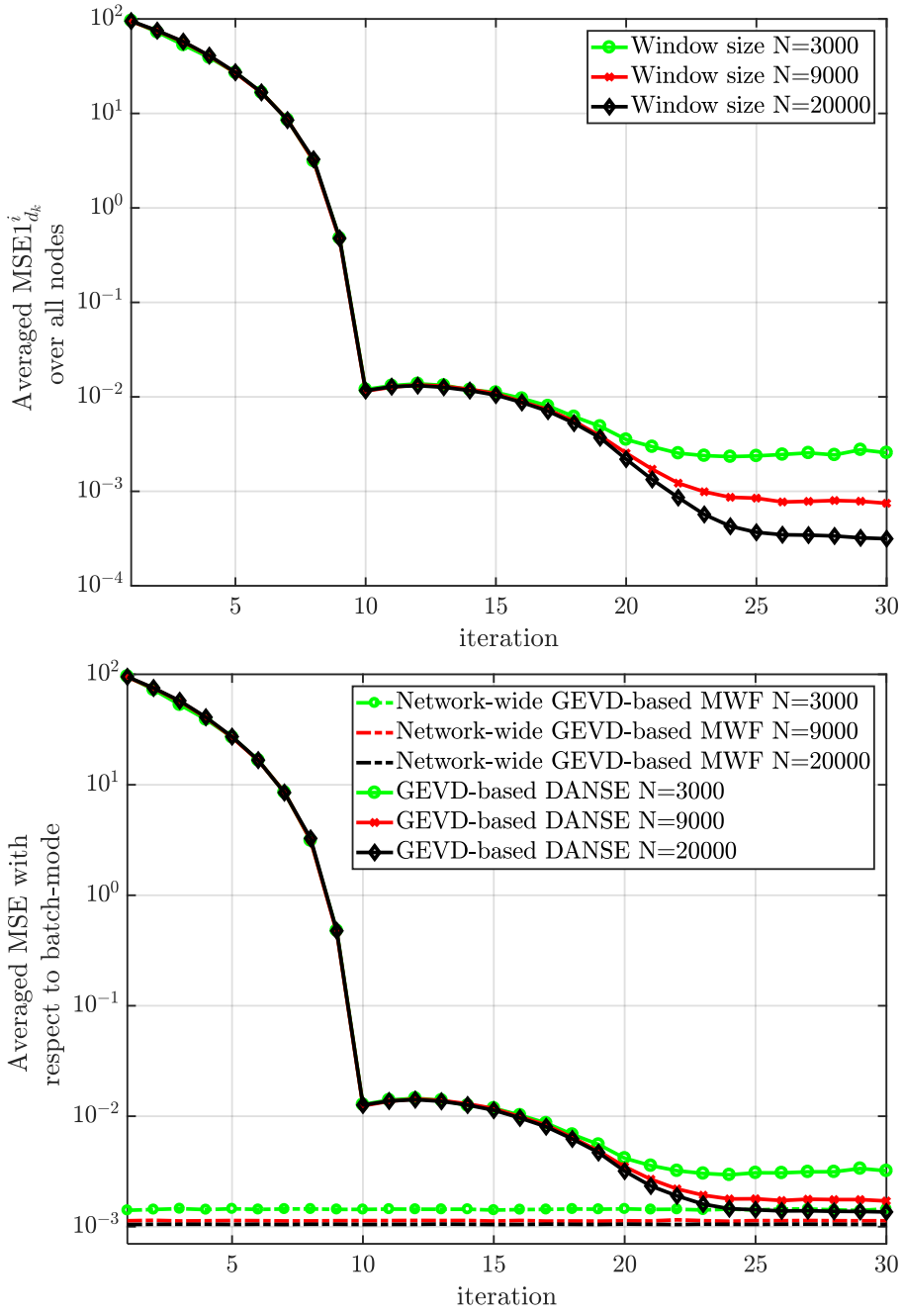


Figure 3.8: Convergence of the GEVD-based DANSE algorithm ($R = 3$) where the correlation matrices are estimated over a finite-window of N samples.

3.6 Conclusion and future work

In this chapter, we have proposed a distributed algorithm for the estimation of node-specific desired signals in wireless sensor networks. The estimation has been based on a GEVD-based low-rank approximation of a correlation matrix within the MWF that is locally computed at each node. The resulting GEVD-based DANSE algorithm significantly compresses the signals transmitted between the nodes compared to a centralized approach with a fusion center. We have shown theoretically and via simulations that the GEVD-based DANSE algorithm converges to the network-wide GEVD-based MWF as if each node would have access to all the sensor signal observations.

Our future work focuses on the development of a GEVD-based DANSE algorithm in which nodes are able to converge to the network-wide GEVD-based MWF, when the topology changes in between iterations, e.g., because of link failures, without having to re-converge to a new set of parameters for the new topology.

Chapter 4

Distributed integrated noise reduction and node-specific DOA estimation

Cooperative integrated noise reduction and node-specific direction-of-arrival estimation in a fully connected wireless acoustic sensor network

Amin Hassani, Alexander Bertrand and Marc Moonen

Published in *Signal Processing*, vol. 107, Feb. 2015, pp. 68-81, DOI: 10.1016/j.sigpro.2014.09.001.

Abstract

In this chapter, we consider cooperative node-specific direction-of-arrival (DOA) estimation in a fully connected wireless acoustic sensor network (WASN). We consider a scenario where each node is equipped with a local microphone array with a known geometry, but where the position of the nodes, as well as their relative geometry and hence the between-nodes signal coherence model is unknown. The local array geometry in each node defines node-specific DOAs with respect to a set of target speech sources and the aim is to estimate these in each node. We assume a noisy environment with localized and/or diffuse noise sources, i.e., the noise can be correlated over the different microphones. A distributed noise reduction algorithm can then be applied as a preprocessing step to denoise all the microphone signals of the WASN, based on the distributed adaptive node-specific signal estimation (DANSE) algorithm. The denoised local microphone signals can then be used in each node to estimate the node-specific DOAs by using a subspace-based DOA estimation, involving a (generalized) eigenvalue decomposition of the local microphone signal correlation matrices. It is seen that the fused microphone signals that are exchanged between the nodes in the DANSE algorithm can also be included in these correlation matrices to obtain improved DOA estimates, leading to a cooperative integrated noise reduction and DOA estimation scheme, where the noise reduction can actually be shortcut. The improved performance achieved by this cooperative DOA estimation is demonstrated by means of numerical simulations for two different subspace-based DOA estimation methods (MUSIC and ESPRIT).

4.1 Introduction

Microphone arrays facilitate spatiotemporal processing in acoustic applications and allow to exploit the spatial characteristics of the acoustic scenario to estimate a parameter or signal of interest. For example, they allow to estimate the direction from which target sound signals originate, and/or to perform spatial filtering to suppress undesired sound signals coming from other directions. Microphone arrays have been widely used in hearing aids, teleconferencing systems, automatic speech recognition, hands-free telephony, etc. [2], [135]. In general, the estimation performance improves when more microphones are used, and often also when the spacing between the microphones is increased. However, due to limitations in terms of space, power and processing capabilities of devices with embedded microphone arrays, it is not always possible to have an array with these desired characteristics [30].

One remedy could be to use a so-called wireless acoustic sensor network (WASN) [30]. A WASN consists of spatially distributed nodes, which are each equipped with a microphone array, a digital signal processing (DSP) unit and with wireless communication facilities to exchange data with other nodes in the WASN. As a result, the nodes can cooperate to solve certain acoustic signal processing tasks by exchanging relevant information amongst each other.

Direction-of-arrival (DOA) estimation of a target sound signal with respect to a given microphone array plays a crucial role in many applications. For example, based on the estimated DOA, one can control the look direction of a camera, or design an adaptive spatial filter to steer a beam towards the actual location of the target sound source and steer nulls towards the location of noise sources [3,4]. In this chapter, we consider subspace-based DOA estimation techniques, which rely on the estimation of a so-called signal and noise subspace from the (generalized) eigenvalue decomposition of the microphone signal correlation matrices. For example, in the case of a narrowband signal and a fully calibrated microphone array, e.g., multiple signal classification (MUSIC) [114], maximum likelihood methods (MODEs) [136] or weighted subspace fitting (WSF) [137] can be used for DOA estimation. In this category, perhaps MUSIC is the most popular super resolution algorithm which can be applied to an array with an arbitrary but known geometry. However, MUSIC has a relatively high computational complexity. To reduce the computational cost, the so-called estimation of signal parameters via the rotational invariance technique (ESPRIT) [115] may be used as an alternative, which is also more robust with respect to array imperfections compared to MUSIC [115]. However, ESPRIT can only be applied to arrays with specific geometries [115].

When a wideband signal (such as a speech signal) is considered, a wideband

extension of a narrowband DOA estimation should be utilized. Methods such as steered covariance matrix (STCM) [138] and spatial smoothing or array manifold interpolation (AMI) [139], are based on coherent focusing, i.e., they perform a narrowband method on a single frequency-steered coherent covariance matrix [138], [140]. The class of so-called incoherent methods, applies a narrowband method on each frequency bin independently and averages them all in the end [141], [142]. It has been demonstrated in [143] that for sources with non-flat spectra (such as a speech signal), incoherent wideband MUSIC (IWM) leads to more accurate results compared to results of the coherently steered MUSIC. Therefore, but without loss of generality (w.l.o.g.), we consider incoherently averaged wideband methods in the sequel.

Although we will focus on MUSIC and ESPRIT in this chapter, it is noted that there are several other subspace-based DOA estimation techniques for (partially) calibrated microphone arrays, e.g., rank reduction (RARE) [144], multiple invariance ESPRIT [145] and multiple invariances MUSIC and MODE [146].

In this chapter, we consider cooperative node-specific direction-of-arrival (DOA) estimation in a fully connected wireless acoustic sensor network (WASN). We consider a noisy scenario where the position of the nodes as well as the relative geometry between them are unknown, but where each node is equipped with a local microphone array, with a known local geometry. The local geometry defines node-specific DOAs with respect to a set of target speech sources and the aim is to estimate these in each node[1]. This means that, unlike e.g., [48], [147], the aim is to take benefit from the correlation between the microphone signals of the different arrays without modeling the unknown coherence structure between them. In practice, this is of great importance since even theoretical modeling of the spatial coherence cannot perfectly describe the environmental impacts (turbulence) that disturb the natural spherical propagation of wavefronts, especially when nodes are placed far apart [148], [48].

We assume a noisy environment where the noise can be spatially correlated, i.e., due to localized and/or diffuse noise sources, which may deteriorate the performance of the DOA estimation. Therefore a multi-channel noise reduction algorithm can be applied as a preprocessing step to denoise all the microphone signals of the WASN. However, it is important that this noise reduction does not remove the spatial information associated to the target speech signal in the individual microphone signals. Furthermore, due to the unknown node and source positions, the noise reduction must rely on a blind beamforming technique, e.g., the multi-channel Wiener filter (MWF) [112]. In essence, MWF adopts a minimum mean square error (MMSE) criterion to estimate the desired

[1] One application could be a video conferencing in which on top of the noise reduction for speech enhancement, we are also interested in steering each node's built-in camera towards the location of a certain speaker.

target speech signal as it is observed in the microphones and therefore allows to preserve the spatial characteristics of the target speech signal in the individual microphones such that DOA estimation can be performed on the denoised signals.

In order to apply a network-wide MWF, all the microphone signals of the WASN must be centralized and processed in a fusion center which may however demand a large communication bandwidth and computational power. An alternative could be a decentralized processing which is inherently scalable in terms of the communication bandwidth and computational complexity. The distributed adaptive node-specific signal estimation (DANSE) algorithm [67], [68], is an iterative algorithm that distributes the processing task of the centralized MWF amongst the nodes. In the case of DANSE, the nodes broadcast fused microphone signals which, assuming a fully connected topology, can be captured by all other nodes in the network. Under mild conditions, DANSE converges to the centralized MWF solution as if all microphone signal signals were available in each node [67], [68], allowing each node to optimally denoise all of its local microphone signals.

Because of the node-specific nature of DANSE, it is well suited to be applied in conjunction with the node-specific DOA estimation. The denoised local microphone signals can then be used in each node to estimate the node-specific DOAs by using a subspace-based DOA estimation, involving a (generalized) eigenvalue decomposition of the local microphone signal correlation matrices. It will be demonstrated that the fused microphone signals that are exchanged between the nodes in the DANSE algorithm can also be included in these correlation matrices to obtain improved DOA estimates, leading to a cooperative integrated noise reduction and DOA estimation scheme, where the DANSE final filtering stage can actually be shortcut.

The chapter is organized as follows. The data model and problem statement are presented in Section 4.2. Section 4.3 briefly reviews three subspace-based DOA estimation methods. Section 4.4 first describes the MWF algorithm for noise reduction and then outlines its distributed implementation based on the DANSE algorithm. Section 4.5 presents DANSE-based node-specific DOA estimation. Section 4.6 first addresses some evaluation aspects and then presents the simulation results. Finally, conclusions are drawn in Section 4.7.

4.2 Data model, problem statement and preview

We consider a WASN with K nodes in which each node $k \in \{1, \ldots, K\}$ is equipped with M_k microphones forming a uniform linear array (ULA), a set of

M_k collinear microphones with equal spacing. The ULA geometry is selected here for the sake of an easy exposition, but w.l.o.g., i.e., other geometries may be considered as well as long as a DOA estimation procedure is used that can handle general geometries (e.g., MUSIC). The topology of the network is assumed to be fully connected which means that data broadcast by one node can be received by all other nodes in the network. The signal of microphone m at node k (frequency domain representations) can be decomposed as

$$y_{km}(\omega) = s_{km}(\omega) + n_{km}(\omega) \tag{4.1}$$

where $s_{km}(\omega)$ and $n_{km}(\omega)$ are the target speech component and undesired noise component, respectively, and ω is the discrete frequency domain variable, where the resolution is defined by the discrete Fourier transform (DFT) of size L. In the sequel, whenever it is possible, we omit ω for the sake of brevity. By stacking (4.1) for $m = 1, \ldots, M_k$, we obtain $\mathbf{y}_k = [y_{k1} \ldots y_{kM_k}]^T = \mathbf{s}_k + \mathbf{n}_k$. All the $\mathbf{y}_k$'s are stacked in the full M-dimensional signal vector $\mathbf{y} = [\mathbf{y}_1^T \ldots \mathbf{y}_K^T]^T$ in which $M = \sum_{k=1}^{K} M_k$. Considering $\check{\mathbf{s}}$ as the signal generated by S target speech sources, we have $\mathbf{s}_k = \mathbf{A}_k(\boldsymbol{\theta}_k)\,\check{\mathbf{s}}$ in which the steering matrix $\mathbf{A}_k(\boldsymbol{\theta}_k)$ is defined as

$$\mathbf{A}_k(\boldsymbol{\theta}_k) = [\mathbf{a}_{k1}(\theta_{k1}) \ldots \mathbf{a}_{kS}(\theta_{kS})] \tag{4.2}$$

where $\mathbf{a}_{ks}(\theta_{ks})$ is the node-specific M_k-dimensional steering vector which is composed of the acoustic transfer functions (including room acoustics and microphone characteristics) from the s-th target speech source to the microphones of node k, and where $\boldsymbol{\theta}_k = [\theta_{k1} \ldots \theta_{kS}]^T$ is the set of corresponding node-specific DOAs with respect to the ULA of node k. For a ULA, the so-called array steering (response) vector $\mathbf{g}_k(\omega, \theta)$, which expresses the relative phase shifts of the target speech signal s in all microphones at node k with respect to the first microphone of its local ULA for a given DOA θ, can be generally modeled as [4],

$$\mathbf{g}_k(\omega, \theta) = \begin{bmatrix} 1 \\ e^{-j\omega d\, cos(\theta) f_s / c} \\ \vdots \\ e^{-j\omega (M_k - 1) d\, cos(\theta) f_s / c} \end{bmatrix} \tag{4.3}$$

where f_s is the sampling frequency, c is the speed of sound, and d is the inter-microphone distance of the ULA of node k. Note that (4.3) assumes that all microphones have the same ideal omni-directional directivity response, that the relative attenuation factors are neglectable, and that far-field conditions are satisfied. These are common assumptions in DOA estimation algorithms [4], and they are a reasonable approximation when the inter-microphone distances are small. This is indeed the case for the local ULAs that are embedded in

a sensor node, as envisaged in this chapter. It is noted that we only impose these assumptions locally on a per-node basis, but not with respect to the network-wide array.

It is reiterated that each node $k \in \{1, \ldots, K\}$ observes node-specific DOAs $\boldsymbol{\theta}_k$ originating from the same set of target speech sources, and that the goal for each node is to estimate its node-specific DOAs $\boldsymbol{\theta}_k$ in a noisy acoustic environment. To this end, the nodes can first cooperate to denoise their local microphone signals, where each node broadcasts observations of fused microphone signals, as defined by the DANSE algorithm. In a second step, each node can use its M_k denoised microphone signals, as well as the denoised fused signals that are exchanged between the nodes in the noise reduction step, as inputs to a subspace estimation and a subspace-based DOA estimation algorithm. A schematic diagram of this approach with cascaded noise reduction and subspace estimation is depicted in Figure 4.1 where the different blocks will be explained later in more detail. It will be demonstrated, however, that the DANSE algorithm allows to integrate the subspace estimation into the noise reduction and that the DANSE final filtering stage can then be shortcut. A schematic diagram of this approach with integrated noise reduction and subspace estimation is depicted in Figure 4.2.

For the sake of brevity, throughout Sections 4.3-4.5 we only consider the special case of a single target speech source, i.e., $S = 1$. The multi-source case can be derived straightforwardly, where all vector-variables can be replaced by their matrix equivalent. A multi-source scenario is considered in Section 4.6 to further show the effectiveness of the proposed cooperative method in the general case.

4.3 Subspace-based DOA estimation

Subspace-based DOA estimation algorithms essentially extract the so-called signal and noise subspace from the microphone signal correlation matrices and estimate the DOAs based on them. In this section, we briefly review the MUSIC and ESPRIT algorithms and their incoherent wideband extensions. It is noted that other subspace-based DOA estimation algorithms can be used as well. We first consider the case where the noise at each microphone is independent and identically distributed (i.i.d.), and we will later consider the more general case. The microphone signal correlation matrix at node k is then equal to

$$\mathbf{R}_{\mathbf{y}_k\mathbf{y}_k} = E\{\mathbf{y}_k\mathbf{y}_k^H\} = \mathbf{R}_{\mathbf{s}_k\mathbf{s}_k} + \sigma_{n_k}^2\mathbf{I}_{M_k} \tag{4.4}$$

where $\mathbf{R}_{\mathbf{s}_k\mathbf{s}_k} = E\{\mathbf{s}_k\mathbf{s}_k^H\}$, $\sigma_{n_k}^2$ is the noise power on each microphone of node k, $E\{\cdots\}$ denotes the expected value operator, the superscript H indicates the conjugate transpose operator, and $\mathbf{I}_{M_k}$ is the $M_k \times M_k$ identity matrix. It is

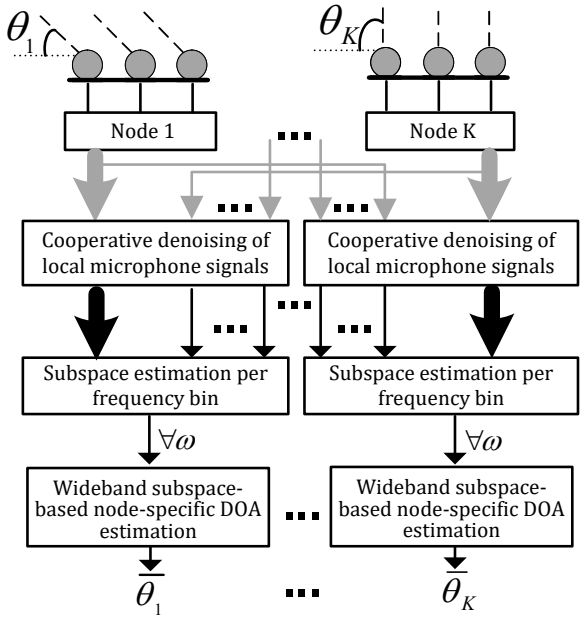


Figure 4.1: Node-specific DOA estimation scheme with cascaded noise reduction and subspace estimation.

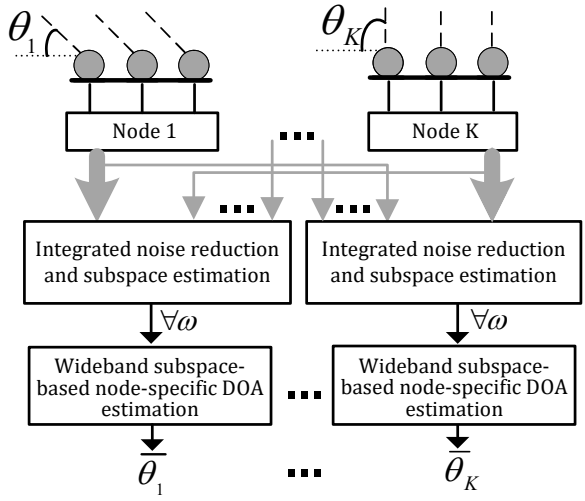


Figure 4.2: Node-specific DOA estimation scheme with integrated noise reduction and subspace estimation.

noted that in (4.4), $\mathbf{R}_{\mathbf{y}_k\mathbf{y}_k}$ and $\mathbf{R}_{\mathbf{s}_k\mathbf{s}_k}$ have the same eigenvectors, which is due to the i.i.d. assumption on the noise. In case of a single target speech source, the correlation matrix of the target speech signal component of the microphone signals, $\mathbf{s}_k$, can be written as

$$\mathbf{R}_{\mathbf{s}_k\mathbf{s}_k} = \sigma_s^2 \mathbf{a}_k \mathbf{a}_k^H \tag{4.5}$$

where $\sigma_s^2 = E\{|\check{s}|^2\}$ is the power of the target speech source signal.

4.3.1 MUSIC

In this section, we provide a very brief outline of the MUSIC algorithm, and we refer to [114] for further details. Basically, MUSIC decomposes the correlation matrix $\mathbf{R}_{\mathbf{y}_k\mathbf{y}_k}$ at each frequency ω, into a signal and noise subspace which are orthogonal to each other, e.g., by means of an eigenvalue decomposition (EVD). In the case of a single target speech source, the signal subspace is defined by the eigenvector corresponding to the largest eigenvalue of $\mathbf{R}_{\mathbf{y}_k\mathbf{y}_k}$, and the noise subspace is spanned by the remaining $(M_k - 1)$ eigenvectors. The matrices containing the basis vectors for the signal and noise subspace are then denoted as

$$\mathbf{E}_{\mathbf{s}_k} = [\mathbf{q}_{k1}] \tag{4.6}$$

$$\mathbf{E}_{\mathbf{n}_k} = [\mathbf{q}_{k2} | \dots | \mathbf{q}_{kM_k}] \tag{4.7}$$

where $\mathbf{q}_{k1}$ is the eigenvector corresponding to the largest eigenvalue, and $\mathbf{E}_{\mathbf{n}_k}^H \mathbf{q}_{k1} = \mathbf{0}$. Note that these subspaces are different at each frequency ω. For a narrowband signal with a central frequency ω, we define the so-called pseudospectrum as

$$\frac{1}{|\mathbf{g}_k^H \mathbf{E}_{\mathbf{n}_k} \mathbf{E}_{\mathbf{n}_k}^H \mathbf{g}_k|} \tag{4.8}$$

where $\mathbf{g}_k(\omega, \theta)$ is defined in (4.3). It is noted that the denominator will be close to zero if θ equals the true DOA, since then $\mathbf{g}_k \approx \mathbf{a}_{k1} \approx q_{k1}$. Therefore, the θ_k for which the wideband pseudospectrum[2] is maximized, will be the estimated DOA, i.e. (we use an overline (bar) to denote an estimate),

$$\bar{\theta}_k = \arg\max_{\theta_k} \frac{1}{\sum_\omega |\mathbf{g}_k^H \mathbf{E}_{\mathbf{n}_k} \mathbf{E}_{\mathbf{n}_k}^H \mathbf{g}_k|} \tag{4.9}$$

where an exhaustive search over all possible θ_k is performed. Note that although we are considering a ULA, MUSIC also works with other array topologies as long as the array geometry (and hence the array steering vector $\mathbf{g}_k(\omega, \theta)$) is known and fully calibrated.

[2]Note that there are also other alternatives to combine the different frequencies (incoherent averaging) in a wideband pseudospectrum [149].

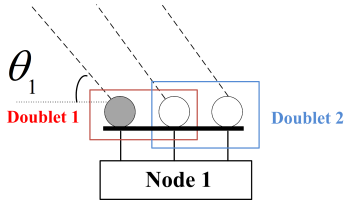


Figure 4.3: Two sub-arrays (doublets) which are used by ESPRIT for a three-element array

4.3.2 ESPRIT

ESPRIT [115] is an alternative subspace-based DOA estimation algorithm, which does not require an exhaustive search over all possible DOAs, leading to a computational complexity that is typically lower compared to MUSIC. ESPRIT essentially operates on a doublet structure which means that it decomposes the array into several two-element sub-arrays (doublets) with a known identical displacement vector, i.e. all doublets are identically oriented with the same local inter-microphone distance. Figure 4.3 shows how ESPRIT splits a three-microphone ULA into two overlapping doublets.
In the rest of this section we briefly describe the ESPRIT algorithm for a ULA and the reader is referred to [115] for further details.

Given the signal subspace $\mathbf{E}_{\mathbf{s}_k}$ in (4.6) which is extracted from the correlation matrix $\mathbf{R}_{\mathbf{y}_k\mathbf{y}_k}$ by means of an EVD, ESPRIT defines two subvectors $\mathbf{v}_{1_k}$ and $\mathbf{v}_{2_k}$ with the first M_k-1 and the last M_k-1 entries of $\mathbf{E}_{\mathbf{s}_k}$. The i-th component in $\mathbf{v}_{1_k}$ and $\mathbf{v}_{2_k}$ then corresponds to the i-th doublet in the array. Whenever $\mathbf{E}_{\mathbf{s}_k}$ exactly matches the array steering vector expressed in (4.3) we can write

$$\mathbf{v}_{2_k} = \mathbf{v}_{1_k}\psi \tag{4.10}$$

where for an estimated $\mathbf{E}_{\mathbf{s}_k}$, the ψ is estimated by using a least squares solver, i.e.,

$$\bar{\psi} = \left(\mathbf{v}_{1_k}^H \mathbf{v}_{1_k}\right)^{-1} \mathbf{v}_{1_k}^H \mathbf{v}_{2_k}. \tag{4.11}$$

Since both $\mathbf{v}_{1_k}$ and $\mathbf{v}_{2_k}$ are often noisy, a total least squares (TLS) solver can be used alternatively which is based on the singular value decomposition (SVD) of the matrix $[\mathbf{v}_{1_k}|\mathbf{v}_{2_k}]$. Considering the array manifold vector expressed in (4.3), we can then write

$$\bar{\theta}_k = \cos^{-1}\left(\frac{\angle\bar{\psi}}{\omega d f_s/c}\right). \tag{4.12}$$

For the case of a wideband signal, the DOA estimation is then obtained from the average over the DOA estimates for different frequencies (incoherent averaging).

4.4 Noise reduction preprocessing

In the previous section we have considered the problem of DOA estimation in an acoustic environment with i.i.d. microphone noise, i.e. the noise signals are spatially uncorrelated and identically distributed. In the sequel, we consider the general case where the noise signals in the different microphones may be correlated, e.g., due to the presence of localized acoustic noise sources. In such an environment and in order to estimate the DOAs using one of the subspace-based algorithms explained so far, we can perform a noise reduction as a preprocessing step to denoise the local microphone signals in each node. The target speech correlation matrix $\mathbf{R}_{\mathbf{s}_k\mathbf{s}_k}$ (see (4.5)) is then estimated based on the denoised local microphone signals. For the noise reduction we consider multichannel Wiener filtering (MWF) [112] which in contrast to standard beamforming, does not require prior information on the microphone or source positions. In Subsection 4.4.1, we briefly review the MWF, and in Subsection 4.4.2, we explain how the nodes can cooperate to improve the overall noise reduction performance.

4.4.1 Multi-channel Wiener filter

The goal of MWF is to estimate the target speech signal s_{km} as it is observed in the m-th microphone of node k. MWF performs a filter-and-sum operation in which the filter coefficients $\mathbf{w}_{km}$ are selected such that the following mean square error (MSE) cost function is minimized

$$\min_{\mathbf{w}_{km}} E\{\left|\mathbf{e}_m^H\mathbf{s}_k - \mathbf{w}_{km}^H\mathbf{y}_k\right|^2\} \tag{4.13}$$

where $\mathbf{e}_m = [0\ldots0\,1\,0\ldots0]^T$, where the 1 is the m-th coefficient. The solution to this minimum MSE (MMSE) problem, assuming independence between $\mathbf{s}_k$ and $\mathbf{n}_k$, is given as [112]

$$\mathbf{w}_{km} = \left(\mathbf{R}_{\mathbf{y_k}\mathbf{y_k}}\right)^{-1}\mathbf{R}_{\mathbf{s_k}\mathbf{s_k}}\mathbf{e}_m. \tag{4.14}$$

Again assuming independence between $\mathbf{s}_k$ and $\mathbf{n}_k$, we can write

$$\mathbf{R}_{\mathbf{s_k}\mathbf{s_k}} = \mathbf{R}_{\mathbf{y_k}\mathbf{y_k}} - \mathbf{R}_{\mathbf{n_k}\mathbf{n_k}} \tag{4.15}$$

where $\mathbf{R}_{\mathbf{n_k}\mathbf{n_k}} = E\{\mathbf{n}_k\mathbf{n}_k^H\}$. Estimation of the covariance matrices ($\mathbf{R}$ matrices) can be done by time averaging in the short-time-Fourier-transform (STFT) domain. $\mathbf{R}_{\mathbf{y_k}\mathbf{y_k}}$ can be estimated during "speech-and-noise" signal segments and $\mathbf{R}_{\mathbf{n_k}\mathbf{n_k}}$ can be estimated during "noise-only" signal segments. To distinguish between "noise-only" and "speech-and-noise" signal segments, a voice activity detection (VAD) mechanism must be applied [112].

In the case of a single speech source, $\mathbf{R}_{\mathbf{s_k s_k}}$ is given by (4.5) and hence is a rank-1 matrix. In practice, however, due to (a) the finite DFT size in the STFT analysis, (b) the non stationarity of the noise and (c) the finite observation set (which leads to estimation errors), the rank of the estimated $\bar{\mathbf{R}}_{\mathbf{s_k s_k}}$ will be greater than one. Moreover, in low input signal to noise ratio (iSNR) conditions, we have that $\bar{\mathbf{R}}_{\mathbf{y_k y_k}} \approx \bar{\mathbf{R}}_{\mathbf{n_k n_k}}$, such that the estimation of $\mathbf{R}_{\mathbf{s_k s_k}}$ via subtraction in (4.15) may result in a covariance matrix $\bar{\mathbf{R}}_{\mathbf{s_k s_k}}$ which is not positive (semi-)definite and this may result in an unstable noise reduction performance [150]. A remedy for this problem is to choose a rank-1 approximation based on either the EVD of $\bar{\mathbf{R}}_{\mathbf{s_k s_k}}$ or the generalized EVD (GEVD) of $\bar{\mathbf{R}}_{\mathbf{y_k y_k}}$ and $\bar{\mathbf{R}}_{\mathbf{n_k n_k}}$ [103]. GEVD-based rank-1 approximation has been shown to deliver the best performance, as it effectively selects the "mode" corresponding to the highest SNR [103]. Therefore the GEVD based rank-1 approximation is utilized in the sequel[3].

Given the matrices $\bar{\mathbf{R}}_{\mathbf{y_k y_k}}$ and $\bar{\mathbf{R}}_{\mathbf{n_k n_k}}$, their joint diagonalization can be written as

$$\begin{aligned} \bar{\mathbf{R}}_{\mathbf{y_k y_k}} &= \bar{\mathbf{V}}_k \bar{\mathbf{\Sigma}}_{y_k} \bar{\mathbf{V}}_k^H \\ \bar{\mathbf{R}}_{\mathbf{n_k n_k}} &= \bar{\mathbf{V}}_k \bar{\mathbf{\Lambda}}_{n_k} \bar{\mathbf{V}}_k^H \end{aligned} \tag{4.16}$$

so that

$$\bar{\mathbf{R}}_{\mathbf{n_k n_k}}^{-1} \bar{\mathbf{R}}_{\mathbf{y_k y_k}} = \bar{\mathbf{V}}_k^{-H} (\bar{\mathbf{\Lambda}}_{n_k}^{-1} \bar{\mathbf{\Sigma}}_{y_k}) \bar{\mathbf{V}}_k^H = \bar{\mathbf{V}}_k^{-H} \bar{\mathbf{\Sigma}}_k \bar{\mathbf{V}}_k^H \tag{4.17}$$

where $\bar{\mathbf{V}}_k$ is an invertible matrix (not necessarily orthogonal) and the columns of $\bar{\mathbf{V}}_k^{-H}$ are the generalized eigenvectors, $\bar{\mathbf{\Sigma}}_{y_k} = diag\{\bar{\sigma}_1 \cdots \bar{\sigma}_{M_k}\}, \bar{\mathbf{\Lambda}}_{n_k} = diag\{\bar{\lambda}_1 \cdots \bar{\lambda}_{M_k}\}$ and the real-valued generalized eigenvalues are defined by the diagonal matrix $\bar{\mathbf{\Sigma}}_k = diag\{\frac{\bar{\sigma}_1}{\bar{\lambda}_1} \cdots \frac{\bar{\sigma}_{M_k}}{\bar{\lambda}_{M_k}}\}$ [122], [112]. The GEVD-based rank-1 approximation of $\mathbf{R}_{\mathbf{s_k s_k}}$, is then given by

$$\bar{\mathbf{R}}_{\mathbf{s_k s_k}} = \bar{\mathbf{v}}_k (\bar{\sigma}_1 - \bar{\lambda}_1) \bar{\mathbf{v}}_k^H \tag{4.18}$$

where $\bar{\mathbf{v}}_k$ is the first column of $\bar{\mathbf{V}}_k$. The MWF formula then becomes (compare with (4.14))

$$\bar{\mathbf{w}}_{km} = \left(\bar{\mathbf{R}}_{\mathbf{y_k y_k}}\right)^{-1} \bar{\mathbf{v}}_k \bar{v}_{km}^* (\bar{\sigma}_1 - \bar{\lambda}_1) \tag{4.19}$$

where $\bar{v}_{km}$ is the m-th component of $\bar{\mathbf{v}}_k$. Finally, the denoised version of the m-th microphone of node k is computed as

$$\bar{d}_{km} = \bar{\mathbf{w}}_{km}^H \mathbf{y}_k. \tag{4.20}$$

After denoising all the microphone signals in each node k, the resulting denoised microphone signals $\bar{\mathbf{d}}_k = [\bar{d}_{k1} \cdots \bar{d}_{kM_k}]$ can be fed to the DOA estimation algorithm.

[3]This is w.l.o.g. since EVD-based rank-1 approximation can also be utilized for the proposed cooperative DOA method.

4.4.2 DANSE-based cooperative noise reduction

In Subsection 4.4.1 we have assumed that each node operates on its own in order to denoise its local microphone signals. If a node k would also have access to the microphone signals of all the other nodes, i.e., the entire M-dimensional signal vector $\mathbf{y}$, it could compute the network-wide MWF to obtain a substantially better noise reduction. However, this would require a large communication bandwidth and computational power in each node. An alternative could be a decentralized processing which is inherently scalable in terms of the communication bandwidth and computational complexity. This is achieved by the DANSE algorithm [67], [68], which can be viewed as a distributed implementation of the network-wide MWF. The computational cost is then shared between the different nodes, and each node only broadcasts one fused signal to the other nodes, rather than its full M_k-dimensional signal vector $\mathbf{y}_k$. Consequently and compared to the centralized network-wide MWF (based on the full M-dimensional signal vector $\mathbf{y}$), the algorithm reduces the required per-node communication bandwidth by a factor M_k as well as the number of input channels in each node which results in a significant computational complexity reduction. It has been shown in [67] that the DANSE algorithm is able to denoise the microphone signals in each node as if each node would have access to all the WASN microphone signals, despite the fact that only one signal per node is broadcast. In Section 4.5, we will explain that the fused microphone signals that are exchanged between the nodes in the DANSE algorithm can also be exploited to improve the subspace estimation in each node and hence the DOA estimation. In the rest of this section we briefly review the DANSE algorithm for a single target speech source in a fully-connected WASN. It is noted that this is only a very concise review to give an idea of the underlying principles. For more details, as well as extensions to multiple speakers and other network topologies, we refer to [67], [68], [71].

In DANSE (for a single source), each node k creates one fused microphone signal z_k by means of a filter-and-sum operation on its own microphone signals and then broadcasts it to all other nodes. The signal z_k at node k is computed as

$$z_k = \mathbf{f}_k^H \mathbf{y}_k \tag{4.21}$$

where the fusion vector $\mathbf{f}_k$ will be defined later. We define $\mathbf{z} = [z_1 \ \dots \ z_K]^T$ and we write $\mathbf{z}_{-k}$ to denote the vector $\mathbf{z}$ in which z_k is excluded.

Node k's own microphone signals together with the z_k-signals received from the other nodes, are stacked in a vector

$$\widetilde{\mathbf{y}}_k = [\mathbf{y}_k^T \ \mathbf{z}_{-k}^T]^T \tag{4.22}$$

For the sake of an easy exposition, we first assume that each node only estimates the target speech signal in its first microphone, and we later extend this for the other microphones. Each node k then computes the local MWF (compare with (4.19)) as

$$\widetilde{\mathbf{w}}_{k1} = \mathbf{R}^{-1}_{\widetilde{\mathbf{y}}_k\widetilde{\mathbf{y}}_k}\widetilde{\mathbf{v}}_k\widetilde{v}^*_{k1}(\widetilde{\sigma}_1 - \widetilde{\lambda}_1) \tag{4.23}$$

where the $\widetilde{\cdot}$ notation is used for quantities that are computed based on the extended signal $\widetilde{\mathbf{y}}_k$ rather than $\mathbf{y}_k$, and we also replace $\widetilde{\widetilde{\cdot}}$ with $\widetilde{\cdot}$ in the sequel for the sake of conciseness. The $\widetilde{\mathbf{w}}_{k1}$ is then partitioned into two parts, one applied to $\mathbf{y}_k$ and one applied to $\mathbf{z}_{-k}$, i.e.,

$$\widetilde{\mathbf{w}}_{k1} = \begin{bmatrix}\mathbf{h}_{k1}\\ \mathbf{g}_{k1}\end{bmatrix} \tag{4.24}$$

and the denoised signal of the first microphone at node k can then be written as

$$\widetilde{d}_{k1} = \widetilde{\mathbf{w}}^H_{k1}\widetilde{\mathbf{y}}_k = \mathbf{h}^H_{k1}\mathbf{y}_k + \mathbf{g}^H_{k1}\mathbf{z}_{-k}. \tag{4.25}$$

In DANSE, the $\mathbf{f}_k$ in (4.21) is then set to $\mathbf{h}_{k1}$, i.e. for all $k \in \{1, \ldots, K\}$

$$\mathbf{f}_k = \mathbf{h}_{k1} \tag{4.26}$$

Note that the fusion vector $\mathbf{f}_k$ is not only a compressor in each node k to generate the z_k signal from the local microphone signals, but also is a part of the MWF for the first microphone signal in (4.25). However, this is a chicken-and-egg problem since to obtain $\mathbf{f}_k$ we have to compute (4.23)-(4.26) first, which in turn require the $\mathbf{z}_{-k}$ from the other nodes. Starting with random entries for the $\mathbf{f}_k$s, $\forall k \in \{1, \ldots, K\}$, DANSE lets each node k iteratively update first its $\mathbf{R}_{\widetilde{\mathbf{y}}_k\widetilde{\mathbf{y}}_k}$ and $\mathbf{R}_{\widetilde{\mathbf{n}}_k\widetilde{\mathbf{n}}_k}$ and then $\widetilde{\mathbf{w}}_{k1}$ and $\mathbf{f}_k$ (using (4.23)-(4.26)) based on the most recent microphone signals of $\widetilde{\mathbf{y}}_k$. The updating procedure can be done in a sequential round-robin fashion [67], or all the nodes can update simultaneously (requiring some minor modifications) [68]. In [67] it is demonstrated that DANSE converges to the network-wide MWF, as if all microphone signals were available in each node.

So far we have only denoised the first microphone of each node $k \in \{1, \ldots, K\}$ with the DANSE algorithm. It can be shown that the other microphone signals can also be optimally denoised based on the same z_k-signals, even though the fusion vectors that generate these z_k-signals are based on the MWF problems corresponding to the first microphones in each node. For example to denoise the second microphone signal of node k, $\widetilde{\mathbf{w}}_{k2}$ is computed with (4.23), where $\widetilde{v}_{k1}$ is merely replaced by $\widetilde{v}_{k2}$.

4.5 Cooperative integrated noise reduction and DOA estimation

4.5.1 Cooperative DOA estimation

In the previous section, we have introduced the DANSE algorithm to denoise the local microphone signals in each node where the nodes cooperate with each other by exchanging fused microphone signals. This preprocessing step allows to reduce the effect of noise in the DOA estimation in each node. As depicted in Figure 4.1, the next step is to estimate the node-specific DOA at each node k, based on all available denoised signals. Now the objective is to re-use the fused microphone signals that are broadcast in the DANSE algorithm to further improve the node-specific DOA estimation performance, leading to a cooperative integrated noise reduction and DOA estimation scheme. To achieve this, the $\mathbf{z}_{-k}$ signals should first also be denoised by the local MWF, i.e., all the signals in $\widetilde{\mathbf{y}}_k$ are first denoised and then fed to the DOA estimation. By stacking the M_k denoised microphone signals together with $K-1$ denoised z_k signals, we can define $\widetilde{\mathbf{d}}_k$ as

$$\widetilde{\mathbf{d}}_k = [\widetilde{d}_{k1} \cdots \widetilde{d}_{kM_k}, \cdots \widetilde{d}_{k(M_k+K-1)}] \tag{4.27}$$

and its corresponding correlation matrix as

$$\mathbf{R}_{\widetilde{\mathbf{d}}_k\widetilde{\mathbf{d}}_k} = E\{\widetilde{\mathbf{d}}_k\widetilde{\mathbf{d}}_k^H\} \tag{4.28}$$

which will be used in the sequel for the node-specific DOA estimation. In order to extract the local signal and noise subspace at node k, an EVD of $\mathbf{R}_{\widetilde{\mathbf{d}}_k\widetilde{\mathbf{d}}_k}$ is performed. If $\bar{\mathbf{u}}_{k,max}$ is the eigenvector corresponding to the largest eigenvalue, then it is noted that since the relative geometry between the nodes is unknown, only the first M_k entries of $\bar{\mathbf{u}}_{k,max}$, defined as $\bar{\mathbf{u}}_k$ can be used for the DOA estimation. Although this means that we throw away information, there is still implicit cooperation between the nodes as the EVD indeed also relies on the fused microphone signals from other nodes, allowing to exploit more correlation structure in the subspace estimation[4]. Figure 4.4 visualizes the dimension of the correlation matrix $\mathbf{R}_{\widetilde{\mathbf{d}}_k\widetilde{\mathbf{d}}_k}$ in the proposed cooperative approach, compared with two other approaches. The first is a "centralized" approach where each node has access to all M microphone signals throughout the entire network, i.e. $\mathbf{y}_k = \mathbf{y}$ in (4.13). In this case we can estimate the full M-dimensional correlation matrix which indeed leads to a better node-specific subspace estimation and hence DOA estimation, but which has a high communication cost. Secondly we will consider an "isolated" approach where each node has only access to its own

[4]We will later provide some more motivation for this claim

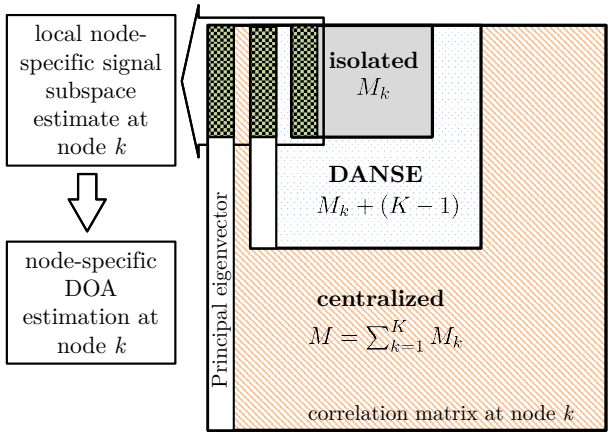


Figure 4.4: Dimension of the correlation matrix $\mathbf{R}_{\tilde{\mathbf{d}}_k \tilde{\mathbf{d}}_k}$ when extra signals are included in the centralized and DANSE approaches, compared to the dimension of the isolated approach.

microphone signals and where there is no cooperation, i.e. the input of each local MWF is merely the $\mathbf{y}_k$ already introduced in Section 4.2. As can be seen in Figure 4.4, the DANSE-based node-specific DOA estimation uses more data than the isolated approach, but less data than the centralized approach. This figure also shows that each node k estimates its node-specific DOA only based on the M_k local entries corresponding to its local array.

Intuitively, there are three effects which explain why the proposed cooperative approach results in a better node-specific DOA estimation at each node:

1. The distributed noise reduction allows the DOA estimation to perform better for a given input SNR and number of microphones.

2. An enhanced subspace estimation is obtained by exploiting more structure due to extending the covariance matrix (see also Subsection 4.5.2). Our proposed cooperative approach exploits coherence between nodes, but without using a model for this coherence based on relative geometry etc., because such models with large inter-microphone distances are typically inaccurate in the first place due to the environmental impacts (turbulence) that disturb the natural spherical propagation of wavefronts.

3. The proposed approach uses the broadcasting signals of DANSE to extend the correlation matrix, which typically have better SNR than when the

nodes would merely transmit raw microphone signals (see also Section 4.6).

According to (4.5), $\bar{\mathbf{u}}_k$ can be treated as a normalized estimate of the steering vector, i.e. $\bar{\mathbf{a}}_k \approx \beta \bar{\mathbf{u}}_k$, where β is an unknown complex number.

To assess the performance of the cooperative DOA estimation in conjunction with the DANSE algorithm, we can apply any of the subspace-based DOA estimation algorithms explained in Section 4.3. In MUSIC, referring to (4.6), we will have $\mathbf{q}_{k1} = \bar{\mathbf{u}}_k$ and then $\mathbf{E}_{\mathbf{n}_k}$ in (4.7), is computed as the $(M_k - 1)$-dimensional subspace orthogonal to $\bar{\mathbf{u}}_k$. Likewise, in ESPRIT, $\mathbf{E}_{\mathbf{s}_k} = \mathbf{q}_{k1} = \bar{\mathbf{u}}_k$.

Remark: In terms of the computational complexity of the proposed cooperative approach, the following items should be considered:

1. When the noise reduction part is taken into account, we have the inversion (see (4.23)) of K times (one per node) an $(M_k + K - 1) \times (M_k + K - 1)$ matrix , versus one $M \times M$ matrix (in the centralized case) .
2. In the subspace-based DOA estimation part, we compute an EVD (see (4.28)) of K times an $(M_k + K - 1) \times (M_k + K - 1)$ matrix , versus one $M \times M$ matrix.

Therefore in both cases, there is a significant benefit since both inversion and EVD are $O(N^3)$ procedures, where N is the dimension of the matrix. However it is known that in practice the communication unit of a WASN node (often battery-powered) consumes much more energy than its DSP unit. Therefore, even more important than the computational gain, as mentioned in Section 4.4.2, there is a reduction in communication cost with a factor M_k per node.

4.5.2 Theoretical motivation

In this section we provide a brief theoretical motivation that explains why the proposed cooperative method improves the performance of the node-specific DOA estimation method. For the sake of an easy exposition, we consider a single-node WASN with M microphones in which all the microphones receive the signal of a target source at the same time (corresponds to a DOA of 90 degrees in far-field conditions). Moreover, for the sake of mathematical tractability, we consider i.i.d. noise components. Therefore, in this case we can write the normalized steering vector as $\hat{\mathbf{a}} = \frac{1}{\sqrt{M}} \mathbf{1}_M$, where $\mathbf{1}_M$ is a M-dimensional vector with all entries equal to 1. Similar to (4.4)-(4.5), we can here write

$$\mathbf{R}_{\mathbf{yy}} = E\{\mathbf{y}\mathbf{y}^H\} = \mathbf{a}\sigma_s^2\mathbf{a}^H + \sigma_n^2\mathbf{I}_M. \tag{4.29}$$

In practice, $\mathbf{R}_{\mathbf{yy}}$ is estimated via time averaging. By defining the $M \times N$ matrix $\mathbf{Y}$ in which each column corresponds to an observation of $\mathbf{y}$ at a certain time instant, we can approximate $\mathbf{R}_{\mathbf{yy}}$ as

$$\mathbf{R}_{\mathbf{yy}} \approx \overline{\mathbf{R}}_{\mathbf{yy}} = \frac{1}{N}\mathbf{Y}\mathbf{Y}^H. \tag{4.30}$$

Based on an EVD we have

$$\mathbf{R}_{\mathbf{yy}}\hat{\mathbf{a}} = \lambda_{max}\hat{\mathbf{a}} \tag{4.31}$$

where $\lambda_{max} = \sigma_s^2 M + \sigma_n^2$. The other eigenvalues $\lambda_m, m = 2, \ldots, M$ are equal to σ_n^2 and correspond to the $(M-1)$-dimensional noise subspace. Let $\bar{\mathbf{a}}$ denote the normalized steering vector estimate computed from the sample covariance matrix $\overline{\mathbf{R}}_{\mathbf{yy}}$. Define the estimation error then as $\Delta\mathbf{a} = \bar{\mathbf{a}} - \hat{\mathbf{a}}$. The second order statistic of $\Delta\mathbf{a}$ can then be described as (see, .e.g., formula (4) in [151])

$$E\{\Delta\mathbf{a}\ \Delta\mathbf{a}^H\} = \frac{\lambda_{max}}{N}\sum_{m=2}^{M}\frac{\lambda_m}{(\lambda_{max}-\lambda_m)^2}\hat{\mathbf{a}}_m\hat{\mathbf{a}}_m^H. \tag{4.32}$$

By plugging $\sum_{m=2}^{M}\hat{\mathbf{a}}_m\hat{\mathbf{a}}_m^H = \mathbf{I}_M - \hat{\mathbf{a}}\hat{\mathbf{a}}^H$ in (4.32), and setting $\lambda_{max} = \sigma_s^2 M + \sigma_n^2$ and $\lambda_m = \sigma_n^2, m = 2, \ldots, M$, we can write

$$E\{\Delta\mathbf{a}\ \Delta\mathbf{a}^H\} = \frac{M\sigma_s^2\sigma_n^2 + \sigma_n^4}{M^2 N\sigma_s^4}(\mathbf{I}_M - \hat{\mathbf{a}}\hat{\mathbf{a}}^H). \tag{4.33}$$

Now the objective is to determine how adding more signals (increasing M) affects the steering vector estimation performance. To this end, we examine the MSE of the estimation error $\Delta\mathbf{a}$. Since $\hat{\mathbf{a}} = \frac{1}{\sqrt{M}}\mathbf{1}_M$, and with some straightforward simplifications, we find that

$$\begin{aligned} E\Big\{\|\Delta\mathbf{a}\|^2\Big\} &= E\Big\{Tr\{\Delta\mathbf{a}\Delta\mathbf{a}^H\}\Big\} = Tr\Big\{E\{\Delta\mathbf{a}\Delta\mathbf{a}^H\}\Big\} \\ &= \frac{M\sigma_s^2\sigma_n^2 + \sigma_n^4}{M^2 N\sigma_s^4}(M-1). \end{aligned} \tag{4.34}$$

Finally we define MSE(M) as the MSE per entry of $\Delta\mathbf{a}$ (hence independent of the length of M), i.e.,

$$\begin{aligned} \text{MSE}(M) &= \frac{E\Big\{\|\Delta\mathbf{a}\|^2\Big\}}{M} \\ &= \frac{1}{MN}\left[(\frac{M-1}{M})\frac{\sigma_n^2}{\sigma_s^2} + (\frac{M-1}{M^2})\frac{\sigma_n^4}{\sigma_s^4}\right]. \end{aligned} \tag{4.35}$$

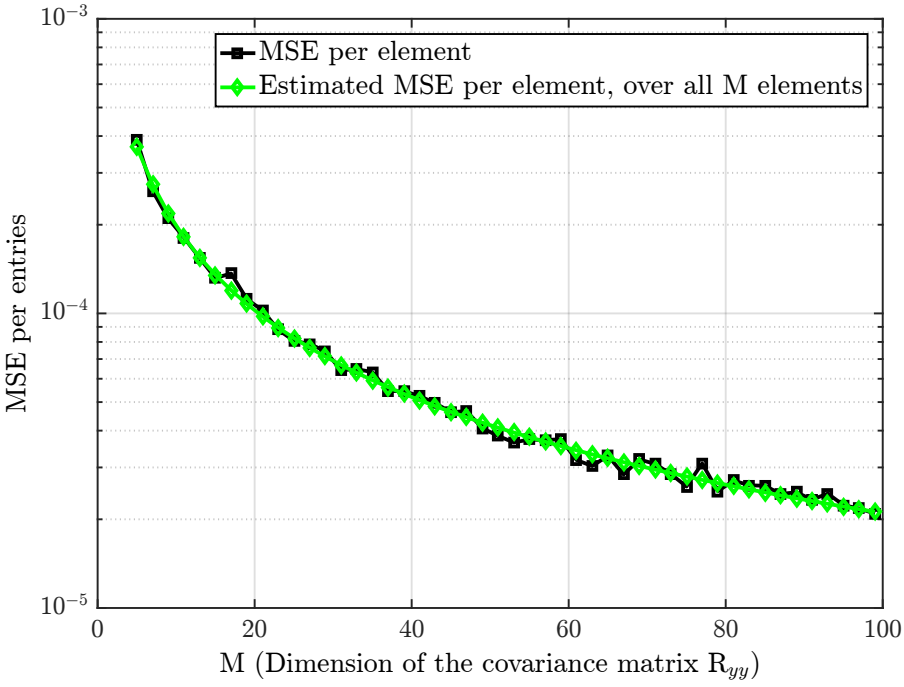


Figure 4.5: MSE of the entries of $\mathbf{\Delta a}$

As can be seen, $\lim_{M\to\infty} \mathrm{MSE}(M) = 0$, i.e., the performance of the steering vector estimation is improved as the dimension of the sample covariance matrix increases, i.e., if extra signals are added. Indeed, this also leads to a better DOA estimation when a subspace-based DOA estimation is considered. Figure 4.5 is provided to better clarify the relationship between the dimension of the sample covariance matrix, i.e., M, and the steering vector estimation performance. This simulation is carried out with $\sigma_s = 2$, $\sigma_n = 1.3$, $N = 200$ over different values of M, and averaged over 200 Monte Carlo runs. The data for the stochastic matrix $\mathbf{Y}$ is drawn from a zero-mean normal distribution based on the covariance matrix described by (4.29). The figure clearly shows how the MSE of the entries of $\mathbf{\Delta a}$ in this case is reduced, as M increases. The results of the Monte Carlo simulations are compared to the theoretical results by plotting (4.35) as a function of M. It is observed that the theoretical prediction is very close to the simulated values. This verifies that increasing the dimension of the sample covariance matrix decreases the per-entry MSE of the steering vector estimation.

Remark: Note that for the case where also correlated noise components exist, it can be shown again that $\lim_{M\to\infty} \mathrm{MSE}(M) = 0$.

4.5.3 Shortcutting the noise reduction

Until now, a cascaded scheme has been proposed in which the first step is to denoise the microphone signals by DANSE and the second step is to estimate the node-specific DOAs based on the denoised signals (Figure 4.1). However it will be shown in this section that exactly the same DOA estimates can be obtained, without explicitly computing the signals in $\widetilde{\mathbf{d}}_k$ and the EVD of its resulting correlation matrix (see (4.28)), which effectively leads to a cooperative integrated noise reduction and DOA estimation scheme, where the noise reduction is shortcut (Figure 4.2).

From (4.23) we can define $\widetilde{\widetilde{\mathbf{W}}}_k$ as

$$\widetilde{\widetilde{\mathbf{W}}}_k = \bar{\mathbf{R}}_{\widetilde{\mathbf{y}}_k\widetilde{\mathbf{y}}_k}^{-1} \widetilde{\mathbf{v}}_k(\widetilde{\sigma}_1 - \widetilde{\lambda}_1)\widetilde{\mathbf{v}}_k^H \tag{4.36}$$

where the m-th column of the $\widetilde{\widetilde{\mathbf{W}}}_k$ corresponds to the MWF to estimate the target speech signal in the m-th component of $\widetilde{\mathbf{y}}_k$ at node k. Note that $\widetilde{\mathbf{d}}_k$ in (4.27) can then be written as $\widetilde{\widetilde{\mathbf{W}}}_k^H\widetilde{\mathbf{y}}_k$. Considering (4.28) and (4.36) we can write

$$\begin{aligned}\bar{\mathbf{R}}_{\widetilde{\mathbf{d}}_k\widetilde{\mathbf{d}}_k} &= \widetilde{\widetilde{\mathbf{W}}}_k^H \bar{\mathbf{R}}_{\widetilde{\mathbf{y}}_k\widetilde{\mathbf{y}}_k} \widetilde{\widetilde{\mathbf{W}}}_k \\ &= \widetilde{\mathbf{v}}_k(\widetilde{\sigma}_1 - \widetilde{\lambda}_1)^* \widetilde{\mathbf{v}}_k^H \bar{\mathbf{R}}_{\widetilde{\mathbf{y}}_k\widetilde{\mathbf{y}}_k}^{-H} \bar{\mathbf{R}}_{\widetilde{\mathbf{y}}_k\widetilde{\mathbf{y}}_k} \bar{\mathbf{R}}_{\widetilde{\mathbf{y}}_k\widetilde{\mathbf{y}}_k}^{-1} \widetilde{\mathbf{v}}_k(\widetilde{\sigma}_1 - \widetilde{\lambda}_1)\widetilde{\mathbf{v}}_k^H\end{aligned} \tag{4.37}$$

and by taking $\rho = |(\widetilde{\sigma}_1 - \widetilde{\lambda}_1)|^2 \widetilde{\mathbf{v}}_k^H \bar{\mathbf{R}}_{\widetilde{\mathbf{y}}_k\widetilde{\mathbf{y}}_k}^{-H} \widetilde{\mathbf{v}}_k$, we have

$$\bar{\mathbf{R}}_{\widetilde{\mathbf{d}}_k\widetilde{\mathbf{d}}_k} = \rho\widetilde{\mathbf{v}}_k\widetilde{\mathbf{v}}_k^H \tag{4.38}$$

which means that $\bar{\mathbf{R}}_{\widetilde{\mathbf{d}}_k\widetilde{\mathbf{d}}_k}$ is immediately a rank-1 matrix and hence the eigenvector corresponding to the largest (non-zero) eigenvalue is equal to $\widetilde{\mathbf{v}}_k$, which is already available from the GEVD of $\bar{\mathbf{R}}_{\widetilde{\mathbf{y}}_k\widetilde{\mathbf{y}}_k}$ and $\bar{\mathbf{R}}_{\widetilde{\mathbf{n}}_k\widetilde{\mathbf{n}}_k}$. As a result, we can shortcut the final filtering stage of the DANSE algorithm that computes the denoised signals of $\widetilde{\mathbf{d}}_k$ which clearly leads to a substantial reduction in computational complexity. Note that this shortcut only holds if an EVD- or GEVD-based rank-1 approximation of $\bar{\mathbf{R}}_{\widetilde{\mathbf{s}}_k\widetilde{\mathbf{s}}_k}$ is used in the local MWFs of the DANSE algorithm.

Remark: As mentioned in Section 4.4, the use of a GEVD for the rank-1 approximation of $\bar{\mathbf{R}}_{\mathbf{s}_k\mathbf{s}_k}$ in the MWF often improves the noise reduction performance compared to the use of an EVD [103,150]. However, in view of the DOA estimation, there is an additional benefit in using a GEVD rather than an EVD. If a random scaling is applied to one of the z_k-signals in $\widetilde{\mathbf{y}}_k$, this results in a similar scaling of the corresponding row and column of the correlation

matrices $\bar{\mathbf{R}}_{\tilde{\mathbf{y}}_k\tilde{\mathbf{y}}_k}$, $\bar{\mathbf{R}}_{\tilde{\mathbf{s}}_k\tilde{\mathbf{s}}_k}$ and $\bar{\mathbf{R}}_{\tilde{\mathbf{n}}_k\tilde{\mathbf{n}}_k}$. This scaling then actually changes the eigenvectors of $\bar{\mathbf{R}}_{\tilde{\mathbf{s}}_k\tilde{\mathbf{s}}_k}$ and therefore also a steering vector estimate based on the EVD. This is undesired, i.e., a simple scaling of the fused microphone signals in one node should not have any effect on the steering vector estimate (and the resulting DOA estimate) in other nodes. It can be shown that the GEVD does not have this effect, i.e., the scaling of a z_k-signal only affects the component in the generalized eigenvectors corresponding to the scaled signal, while the other components remain the same up to a common scaling. As a result, the local steering vector estimate is never affected, as the scaled component is not part of it (remember that the steering vector only consists of the components in the generalized eigenvector that correspond to the microphone signals, and not to the z_k-signals).

4.6 Simulation Results

4.6.1 Evaluation aspects

To demonstrate the effectiveness of the proposed cooperative node-specific DOA estimation, it will be compared with the centralized case and the case where each node performs a local noise-reduction and DOA estimation on its own (the 'isolated' case). Moreover, we consider an approach where each node k merely broadcasts one of its raw microphone signals to the other nodes (instead of the z_k-signal defined in the DANSE) and where these signals are then directly used as additional inputs to the local MWFs, followed by the subspace-based DOA estimation in each node. This is similar to the DANSE-based node-specific DOA estimation, but it relies on a suboptimal cooperative noise reduction scheme instead of the (optimal) DANSE algorithm. This approach does not only result in a reduced noise reduction performance[1], it also results in a slightly worse DOA estimation performance, as will be demonstrate with the simulations. All experiments are performed in a simulated cubic room with dimensions $5\,\mathrm{m} \times 5\,\mathrm{m} \times 5\,\mathrm{m}$ and with wall reflection coefficients $\beta = 0.2$ using the image method [152]. First we consider a WASN with four nodes ($K = 4$), each having a ULA with 3 microphones ($M_k = 3$ for each node k and $M = 12$), with a single target speech source placed at the center of the room (The acoustic scenario is depicted in Figure 4.6). In Subsection 4.6.4, we will also consider the multi-source case. We perform Monte-Carlo simulations (using different speech signals in each run), in a room . Four localized multi-talker noise sources are placed in the room with equal noise power, which is varied to manipulate the

[1] This follows from the fact that the DANSE algorithm always results in an optimal noise reduction [67]

input SNR. We use a sampling frequency of $f_s = 16\,\text{kHz}$, a Hann-windowed DFT with size $L = 512$ and with 50% Hann-window overlaps. An ideal VAD is used to exclude the effect of VAD errors. The target speech source produces short English sentences with a silence period between each two consecutive sentences. Sensor noise and all other spatially uncorrelated noise sources are modeled as uncorrelated white Gaussian noise with 20% of the power of the target speech signal as observed at the microphones. We simulate DANSE in batch mode which means that the required correlation matrices are estimated over the full signal length in each iteration and the DOA estimation is performed after convergence of the DANSE algorithm.

All results in Subsections 4.6.2 and 4.6.3 are averaged over 56 independent Monte Carlo runs and over all nodes. All the figures are plotted as a function of the input SNR at a reference node k which is defined in the time-domain as the power ratio of the speech and noise component in the first microphone signal of node k, i.e.,

$$iSNR_k = \frac{E\left\{|s_{k1}|^2\right\}}{E\left\{|n_{k1}|^2\right\}} = \frac{E\left\{|s_{k1}|^2\right\}}{E\left\{|n_{k1}^m + n_{k1}^p|^2\right\}} \tag{4.39}$$

where n_{k1}^m and n_{k1}^p are assumed to be the signal components corresponding to the uncorrelated sensor noise and the localized noise sources, respectively.

Since the actual subspace estimation performance plays an import role in all the subspace-based DOA estimation algorithms, a proper assessment of the subspace estimation can give a better insight into the merits of our proposed technique. Since only the relative phase differences between the microphones are important, we should define a measure for the subspace estimation performance that is independent of phase or sign ambiguities. To achieve this goal, we again consider the overlapping doublets structure for a ULA as explained in Subsection 4.3.2, and we compute the difference between the phase of the two doublets' estimated steering vectors and of the true array manifold vector at each frequency bin. For node k, this yields (see (4.11) and Figure 4.3)[5]

$$e_k = \frac{1}{\Omega} \sum_{\omega} |\angle\bar{\psi} - \angle\psi| \tag{4.40}$$

where $\Omega = L/2 + 1$ (50% Hann-window overlaps) is the number of the DFT bins.

[5]It could be necessary to add or subtract multiples of 2π to ensure that the absolute phase e_k is in the interval $[0, \pi]$.

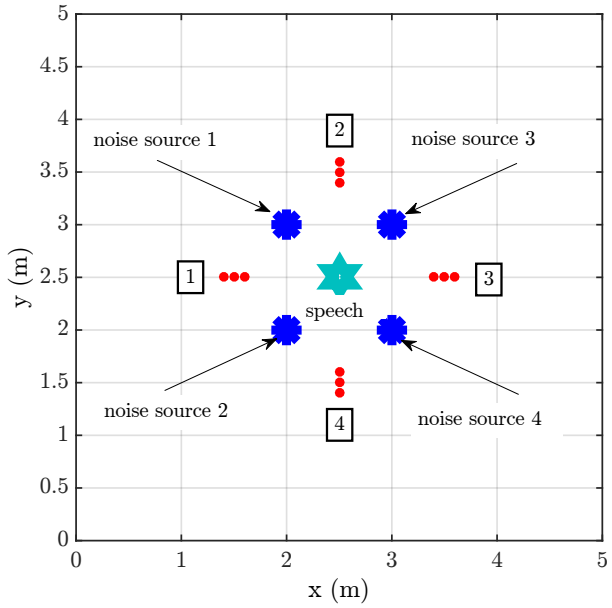


Figure 4.6: Acoustic scenario

The performance of the node-specific DOA estimation by using the DANSE algorithm is evaluated with the subspace-based DOA estimation algorithms outlined earlier in Section 4.3, i.e., MUSIC and ESPRIT.

4.6.2 Scenario 1

We first assume a symmetric scenario in which the true value of the DOA in each node is chosen to be 0 degree (this corresponds to so-called end-fire arrays). Due to this symmetry, iSNR is identical at each node and hence all the nodes are equally important. To change the iSNRs, we change the power of the four localized noise sources uniformly and identically, while keeping the uncorrelated noise level on the microphones unchanged. Figure 4.7 compares the subspace estimation performance based on the measure (4.40) as a function of iSNR when averaged over all frequency bins and all MC runs. As can be seen, DANSE achieves a better subspace estimation than the isolated approach and the approach where nodes merely broadcast their first microphone signal to the other nodes (note that the plot corresponding to the proposed cooperative distributed DOA estimation method based on DANSE almost fully overlaps with the plot for the centralized method).

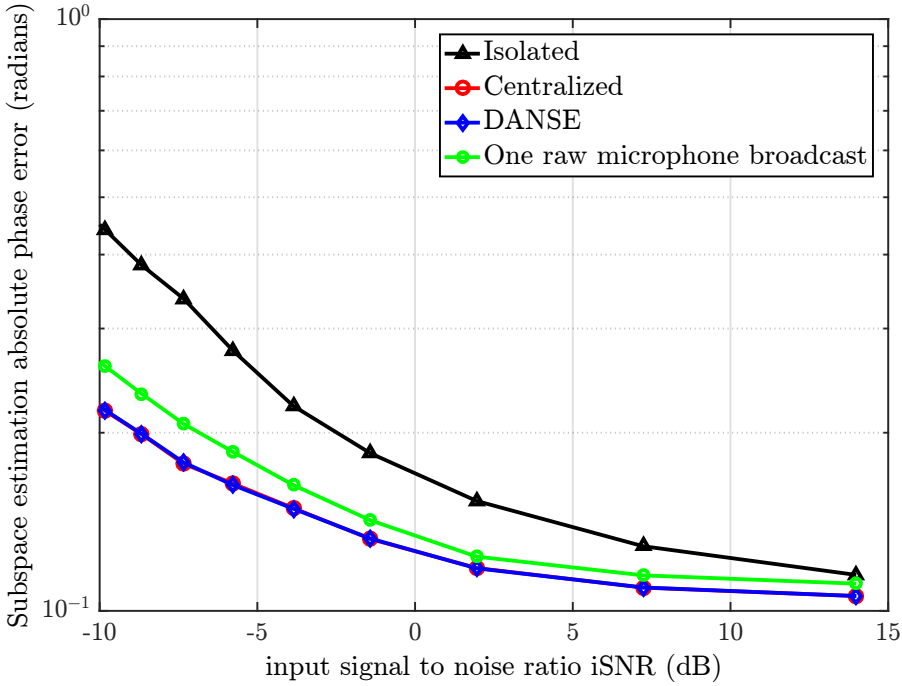
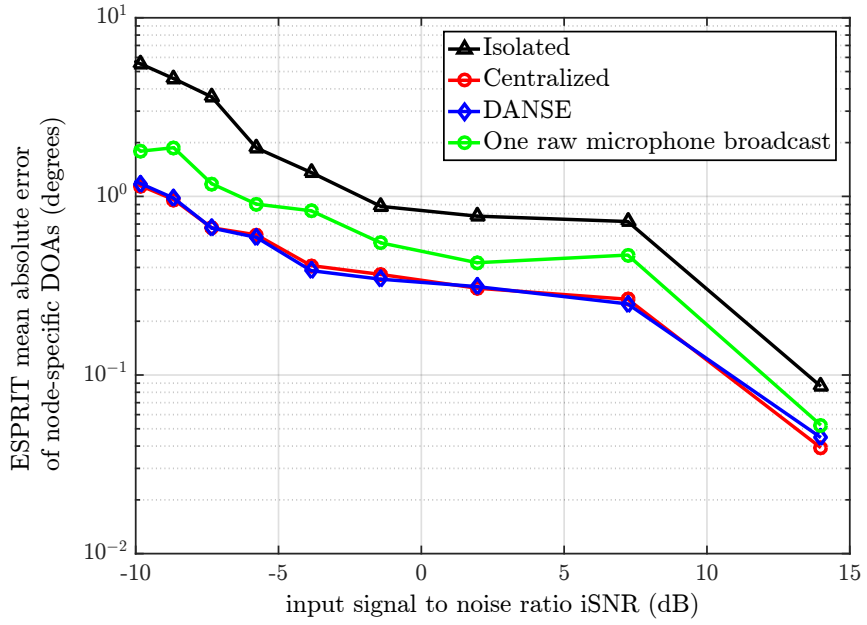


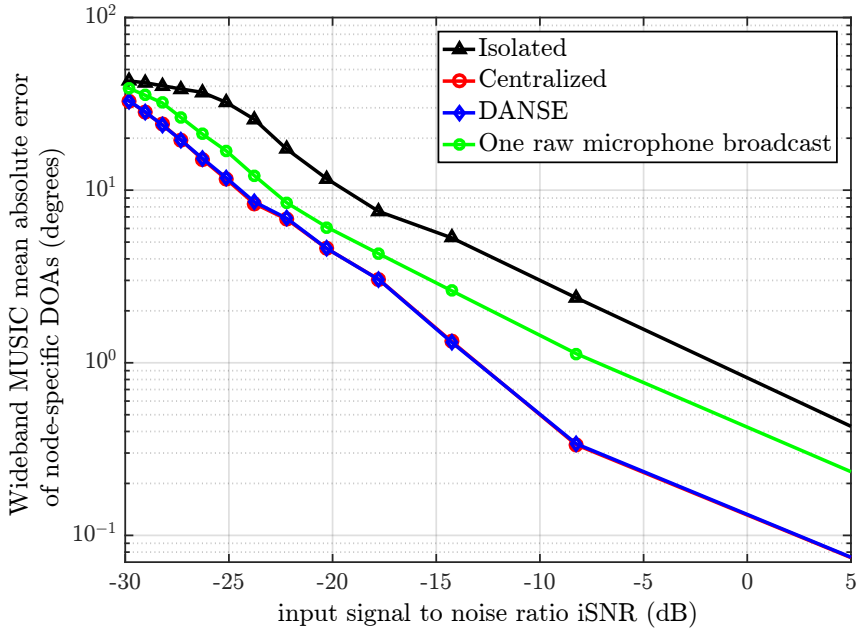
Figure 4.7: Subspace estimation performance

Figure 4.8a shows the averaged absolute values of the DOA estimation errors using ESPRIT. Moreover, the results for DOA estimation with MUSIC are illustrated in Figure 4.8b. As can be seen in these figures, there is a clear benefit in terms of DOA estimation when there is cooperation between the nodes, compared to the isolated approach. If this cooperation is based on the z_k-signals of the DANSE algorithm, the performance of the DOA estimation is closer to the centralized performance compared to the approach where the nodes merely broadcast one microphone signal. The results also show that MUSIC is more robust than ESPRIT. This comes with a significantly higher computational complexity due to the exhaustive searches, which might be impractical in WASNs with limited power supply.

It is noted that the obtained results are better than those in a preliminary study [87]. This is partly due to the fact that the present simulations use a GEVD, rather than an EVD based rank-1 approximation (as used in [87]). As explained in the remark in Section 4.5.3, such a GEVD-based approach is more robust and less dependent on the differences in signal power between the fused microphone signals that are exchanged between the nodes.



(a)



(b)

Figure 4.8: Absolute DOA estimation errors based on (4.8a) wideband ESPRIT and (4.8b) wideband MUSIC.

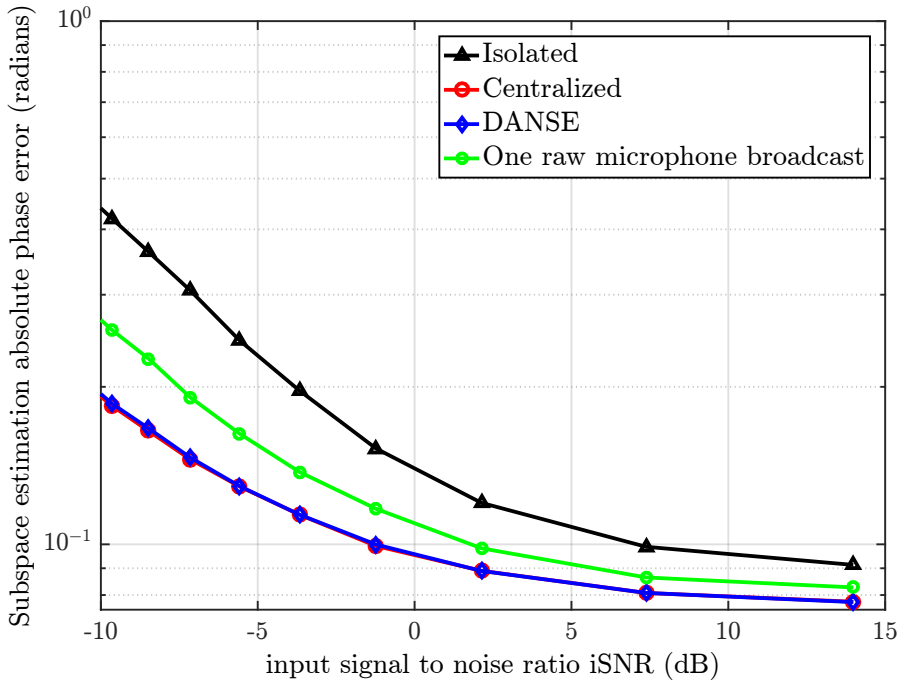
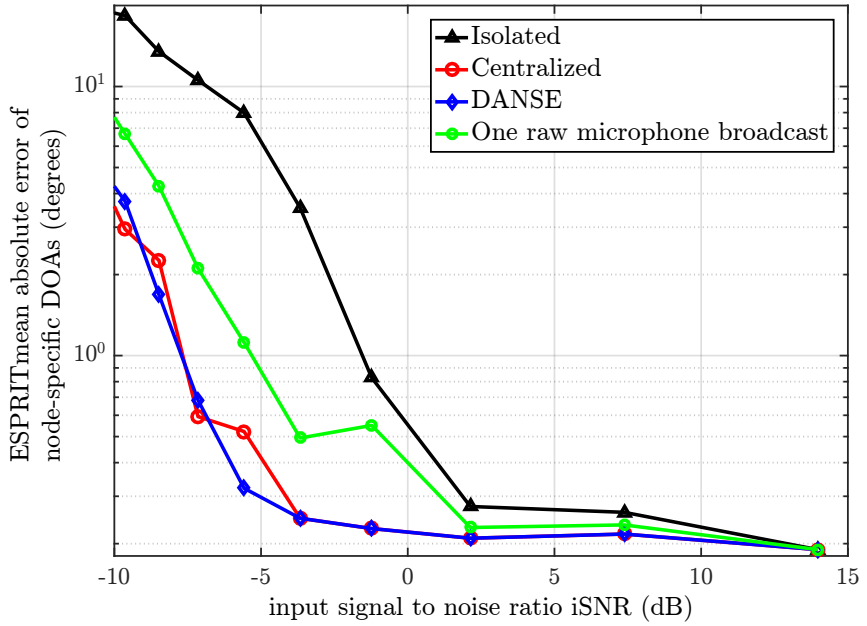


Figure 4.9: Subspace estimation performance with random rotation of the arrays

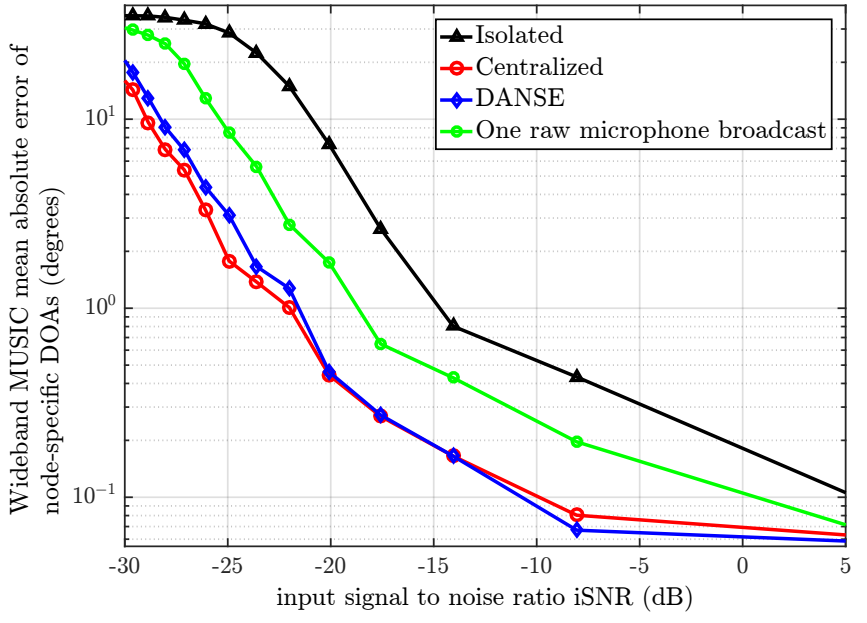
4.6.3 Scenario 2

In order to further investigate the effectiveness of the proposed cooperative node-specific DOA estimation, we now rotate the microphone array in each node independently with a random angle in each MC run. Figure 4.9 compares the subspace estimation performance based on the measure in (4.40) as a function of iSNR when averaged over all the frequency bins and all the MC runs with different true DOAs. Moreover, Figures 4.10a and 4.10b show the averaged absolute values of the DOA estimation errors in degrees for ESPRIT and MUSIC, respectively.

Again, we observe that cooperation between nodes results in a better subspace estimation and hence a better DOA estimation.



(a)



(b)

Figure 4.10: Absolute DOA estimation errors based on (4.10a) wideband ESPRIT and (4.10b) wideband MUSIC, with random rotation of the arrays.

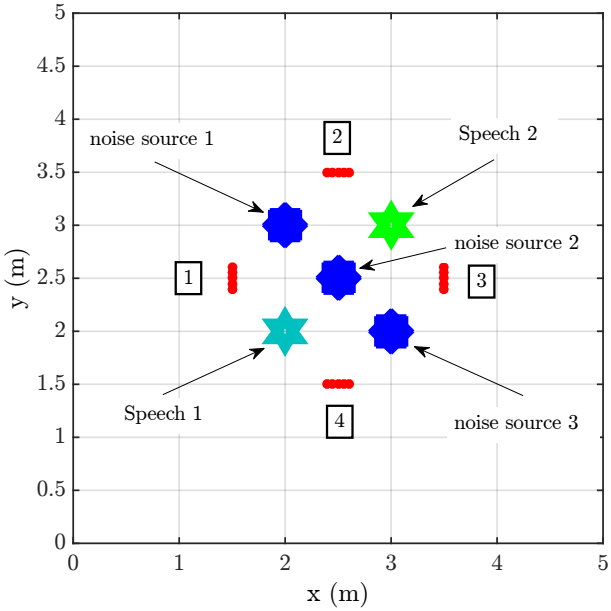
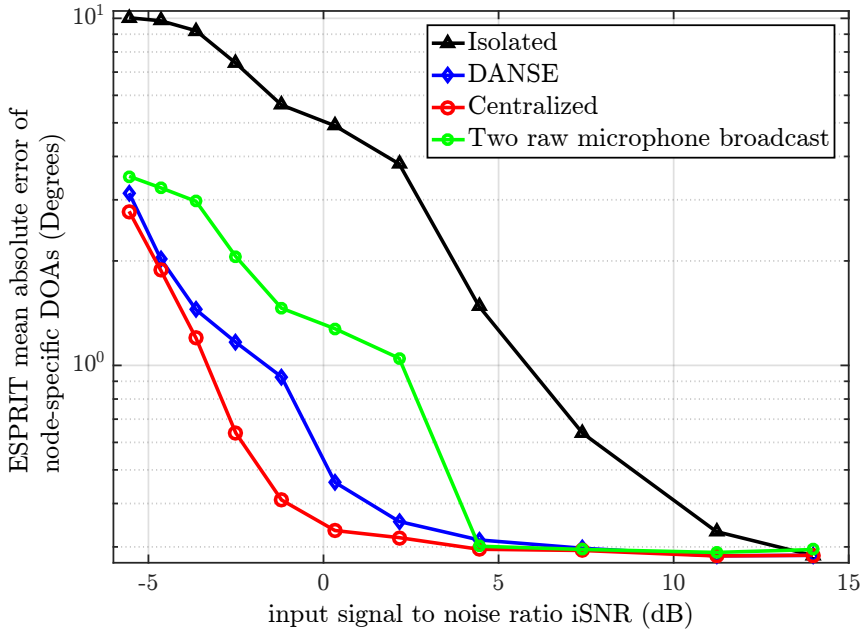


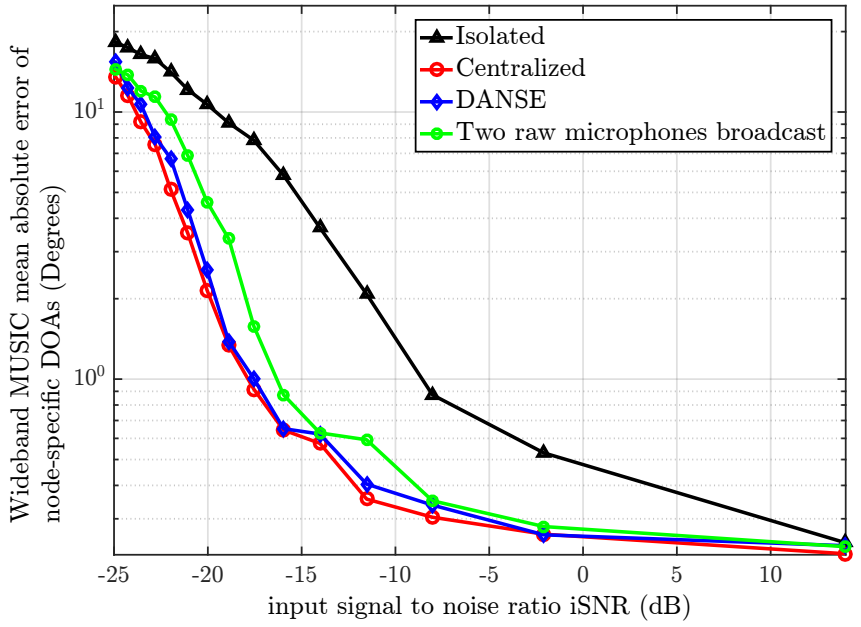
Figure 4.11: Multi-source acoustic scenario

4.6.4 Scenario 3: multi-source case

In this section we consider a multi-source scenario with two target speech sources, while three localized multi-talker noise sources contaminate the captured target speech signals. The multi-source acoustic arrangement is depicted in Figure 4.11. To change the iSNRs, we again increase the power of the three localized noise sources uniformly and identically, while keeping the uncorrelated noise level on the microphones unchanged. Moreover, to have a noise subspace with a higher dimension, here we consider $M_k = 5$ for each node k, hence $M = 20$. While each source consists of a different speech signal, there are some silent intervals for both sources to let nodes estimate the noise statistics. The true DOAs of $K = 4$ nodes with respect to the first and the second (see Figure 4.11) target speech sources are $[45^\circ\ 71^\circ\ 71^\circ\ 45^\circ]$ and $[108^\circ\ 135^\circ\ 135^\circ\ 108^\circ]$, respectively. The simulations are performed by averaging first over absolute estimation errors of 28 Monte Carlo runs, and then over the two estimated DOAs at each node k, and finally over all $K = 4$ nodes. It has been shown in [67] that for multi-source cases, DANSE converges to the centralized MWF performance when each node k broadcasts $min\{S, M_k\}$ fused signal to the other nodes (resulting in a per-node compression factor of $max\{M_k/S, 1\}$ for the data to be sent). Therefore, and since here $S = 2$, DANSE compresses the 5-channel



(a)



(b)

Figure 4.12: Absolute multi-source DOA estimation errors based on (4.12a) wideband ESPRIT and (4.12b) wideband MUSIC, with random rotations.

microphone signal of each node k into a 2-channel signal that is broadcast to the other nodes. Figures 4.12a and 4.12b show the resulting DOA estimation error when ESPRIT and wideband MUSIC are used, respectively. Although the performance plots are now slightly different, the general trend remains the same, i.e., these figures verify that in the general case of estimating DOAs for multiple target speech sources, cooperation between nodes again leads to significantly better DOA estimation. As a reference, the case where nodes merely exchange two of their raw microphone signals is also considered. As can be clearly seen, the performance of the DANSE case is again substantially closer to the centralized case.

4.7 Conclusion

In this chapter, we have studied a cooperative node-specific DOA estimation algorithm in a fully connected WASN in a noisy environment where the position of the nodes as well as the relative geometry or coherence models between them are unknown. The DANSE algorithm is employed as a preprocessing step to first denoise all the WASN microphone signals in a distributed fashion where in each node GEVD-based MWFs are applied for the filtering process. In addition to achieving an optimal noise reduction, the fused microphone signals that are exchanged between the nodes in DANSE are also exploited to improve the node-specific subspace estimation in the DOA estimation algorithm, resulting in a cooperative integrated noise reduction and DOA estimation where the computational cost can be reduced by shortcutting the DANSE final filtering stage. An incoherent wideband version of MUSIC and ESPRIT has been employed to show the effectiveness of the proposed cooperative node-specific DOA estimation. Monte-Carlo simulations have demonstrated that the cooperation between the nodes indeed improves the subspace estimation, and therefore also the multi-source DOA estimation in each node.

Part II

Distributed signal processing algorithms for heterogeneous multi-task WSNs

Chapter 5

Multi-task WSN for signal enhancement, MVDR beamforming and DOA estimation: single source case

Multi-task wireless acoustic sensor network for node-specific speech enhancement and DOA estimation

Amin Hassani, Jorge Plata-Chaves, Alexander Bertrand and Marc Moonen

Published in proceedings of the *IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM)*, Rio De Janeiro, Brazil, Jul. 2016, DOI: 10.1109/SAM.2016.7569718.

Abstract

We consider the design of a distributed algorithm that is suitable for a wireless acoustic sensor network formed by nodes solving multiple tasks (MDMT). In the network, some of the nodes aim at estimating the node-specific direction-of-arrival of some desired sources. Additionally, there are other nodes that aim at implementing either a multi-channel Wiener filter or a minimum variance distortionless response beamformer in order to estimate node-specific desired signals as they impinge on their microphones. By using compressive filter-and-sum operations that incorporate a low-rank approximation of the sensor signal correlation matrix, the proposed MDMT algorithm lets the nodes cooperate to achieve the network-wide centralized solution of their node-specific estimation problems without any knowledge about the tasks of other nodes. Finally, the effectiveness of the algorithm is shown through computer simulations.

5.1 Introduction

Recently, there has been an increasing interest in distributed algorithms that can be implemented over so-called wireless acoustic sensor networks (WASNs). A WASN consists of a multitude of wireless nodes which are equipped with a microphone or an array of microphones. Traditionally, the design of distributed algorithms has focused on networks where the nodes observe the same phenomenon and/or are interested in the same network-wide signal processing task [15]- [16]. However, motivated by the heterogeneity of today's digital networks, recent advances in distributed adaptive signal processing and communication networking are currently enabling a novel paradigm where the networks are formed by Multiple Devices cooperating in Multiple Tasks (MDMT) [153], [154].

Unlike distributed parameter estimation algorithms for single-task networks, (e.g. [75]- [78]), under the MDMT paradigm the design of distributed parameter estimation algorithms assumes that the nodes are interested in estimating different but coupled parameters. Toward this goal, all the resulting distributed parameter estimation algorithms rely on novel node-specific implementations of a particular adaptive filtering technique such as least mean squares (LMS), recursive least squares (RLS) or the affine projection algorithm (APA). For instance, in the context of cognitive radio networks and smart grids, there are node-specific incremental [82], [83] and diffusion [84], [85] algorithms that solve a distributed parameter estimation problem where the nodes are interested in different but overlapping vectors of parameters. Similarly, several diffusion-based algorithms were also derived to facilitate the cooperation among subsets of nodes with similar estimation interests [155]- [158]. The aforementioned works are all focused on regression-based parameter estimation problems. Other works have focused on other parameter estimation problems such as node-specific direction of arrival (DOA) estimation [118]- [120]. In this setting, each node has a different orientation and hence a different DOA with respect to the target source [118].

Another class of distributed MDMT algorithms focuses on node-specific signal estimation (NSSE) problems, which rely on a network-wide spatial filtering. Most of these algorithms rely on compressive filter-and-sum operations on the sensor signals in order to let the nodes solve their NSSE problems with the same performance as a central unit that collects and processes all the sensor signals. These compressive filter-and-sum operations were used to solve NSSE problems under different beamformer criteria where the nodes are interested in estimating samples of desired signals that share a common latent signal subspace, e.g., for speech enhancement in WASNs. Based on the multi-channel Wiener filter (MWF), a distributed algorithm was derived to obtain the centralized

linear minimum mean square error estimates of the node-specific desired signals in binaural hearing aids [34] or in wireless networks with a fully-connected topology [67], a tree topology [71] and combinations thereof [106]. To run all these algorithms over networks that operate under non-stationary and low-SNR conditions, in [123] the estimation of each node-specific desired signal is performed by a MWF in which a low-rank approximation based on a generalized eigenvalue decomposition (GEVD) is incorporated. Moreover, the authors in [33] derived a distributed algorithm under which the estimation of the node-specific signals is undertaken through two different but coupled blind minimum variance distortionless response (MVDR) beamformers. Unlike the MWF-based node-specific signal estimation algorithm, under the blind MVDR criterion the goal of each node is to minimize the output power of a filter subject to linear constraints to avoid distortion of a desired (e.g., speech) signal.

All the aforementioned distributed node-specific estimation algorithms consider a setting where all nodes cooperate to obtain different but coupled solutions of the *same* signal processing (SP) task (e.g., signal enhancement, spectrum estimation, DOA estimation etc.,). Moreover, when obtaining these node-specific estimates, all the existing works assume that all nodes apply the *same* SP technique, e.g., a particular adaptive filter (e.g., LMS, RLS or APA), beamformer (e.g., MWF or MVDR) or subspace-based DOA estimation method. However, in heterogeneous multi-task networks, the nodes may be interested in solving *different* but interrelated SP tasks. Furthermore, each node may apply *different* SP techniques (filters or beamformers) in order to fulfill the particular performance requirement of its application layer. Motivated by these facts, we first define in this chapter an instance of an MDMT network where some nodes aim at obtaining estimates of the node-specific DOAs of some desired sources, while others are interested in enhancing node-specific desired sources by using different beamformer criteria, i.e., a GEVD-based MWF or blind MVDR. Next, after describing the corresponding centralized problems of the different SP tasks, we rely on compressive linear estimation techniques to design a distributed MDMT-based algorithm that lets the nodes cooperate while solving their node-specific SP tasks as if they had access to all the sensor signals available in the network. In this particular setting, the nodes do not even have to be aware of the tasks that other nodes are solving, i.e., MWF, MVDR or DOA estimation. Finally, we provide simulation results for such a multi-task WASN to illustrate its effectiveness.

5.2 Data model and problem statement

We consider a fully-connected WASN with K nodes observing a single target speech source[1]. Each node $k \in \mathcal{K} = \{1, \ldots, K\}$ is equipped with a microphone array consisting of M_k microphones, where its M_k-channel microphone signal is denoted as $\mathbf{y}_k$. We consider frequency domain processing, where the microphone signal $\mathbf{y}_k$ can be modeled as (ω is the frequency variable)

$$\mathbf{y}_k(\omega) = \mathbf{a}_k(\theta_k, \omega)s(\omega) + \mathbf{n}_k(\omega) \tag{5.1}$$

where s is the signal of the target speech source, $\mathbf{a}_k$ is a node-specific M_k-dimensional steering vector (acoustic transfer function from the source to the microphones of node k), θ_k is the DOA at node k, and $\mathbf{n}_k$ is additive noise which includes both spatially correlated (e.g., due to localized noise sources) and uncorrelated (e.g., microphone self-noise) noise contributions. We assume that each node's microphone array has a different orientation, and hence the DOA θ_k is different for each node. In the sequel, we always omit ω for the sake of brevity. We also define $\mathbf{s}_k \triangleq \mathbf{a}_k s$ as the M_k-channel speech component of the noisy microphone signals at node k. By stacking all $\mathbf{y}_k$, $\mathbf{n}_k$ and $\mathbf{s}_k$, we obtain the network-wide M-channel signals $\mathbf{y}$, $\mathbf{s}$ and $\mathbf{n}$, respectively, where $M = \sum_{k=1}^{K} M_k$. With this we can write $\mathbf{y} = \mathbf{s} + \mathbf{n} = \mathbf{a}s + \mathbf{n}$, where $\mathbf{a}$ denotes the network-wide M-dimensional steering vector.

Each node is then tasked to attain a node-specific goal from the following three cases, i.e., $\mathcal{K} \triangleq \{\mathcal{K}^{\mathrm{MWF}} \bigcup \mathcal{K}^{\mathrm{MVDR}} \bigcup \mathcal{K}^{\mathrm{DOA}}\}$. Each node $k \in \mathcal{K}^{\mathrm{MWF}}$ estimates its node-specific desired speech signal d_k from the available noisy microphone signals using MWF [112]. Each node $k \in \mathcal{K}^{\mathrm{MVDR}}$ minimizes the output power of its beamformer under a single linear constraint that steers the beam towards the location of the target speech source such that the target speech signal (as received at its *reference* microphone) is processed without distortion [33]. Finally, each node $k \in \mathcal{K}^{\mathrm{DOA}}$ estimates its node-specific DOA θ_k from the target speech source [118]. We assume that the local microphone array geometry of the nodes $k \in \mathcal{K}^{\mathrm{DOA}}$ is known, but the position of these nodes as well as the relative geometry between them and the other nodes are unknown.

5.3 Centralized estimation

We first consider the centralized estimation problems where we assume that each node k transmits its unprocessed M_k-channel microphone signal $\mathbf{y}_k$ to all

[1] These assumptions are mainly for the sake of easy exposition, as all results can be extended to networks with nearest-neighbor topology or scenarios where multiple target speech sources are present [67]- [106].

other nodes. Therefore the objective for node k is to carry out its node-specific task based on the network-wide M-channel microphone signal $\mathbf{y}$.

5.3.1 MWF

Under the MWF criterion, the goal for each node $k \in \mathcal{K}^{\mathrm{MWF}}$ is to process the network-wide noisy microphone signal $\mathbf{y}$ in order to obtain the linear minimum mean square error (LMMSE) estimate of its node-specific desired signal $d_k = a_{k,r}s$, where $a_{k,r}$ is the r-th element of $\mathbf{a}_k$ corresponding to the reference microphone. Toward this goal, each node $k \in \mathcal{K}^{\mathrm{MWF}}$ then applies an M-dimensional linear estimator $\hat{\mathbf{w}}_k$ to estimate d_k as $\hat{d}_k = \hat{\mathbf{w}}_k^H \mathbf{y}$, with

$$\hat{\mathbf{w}}_k = \min_{\mathbf{w}_k} E\{|d_k - \mathbf{w}_k^H \mathbf{y}|^2\} \tag{5.2}$$

where the hat ($\hat{\cdot}$) refers to the fact that the centralized estimation is considered and where $E\{\cdot\}$ and the superscript H denote the expected value operator and the conjugate transpose operator, respectively. Assuming that $\mathbf{R}_{yy} = E\{\mathbf{y}\mathbf{y}^H\}$ is a full rank matrix, the unique solution of (5.2) is [112]

$$\hat{\mathbf{w}}_k = \mathbf{R}_{yy}^{-1}\mathbf{R}_{ss}\mathbf{e}_k \quad \text{with} \quad \mathbf{R}_{ss} \triangleq E\{\mathbf{s}\mathbf{s}^H\} = P_s\mathbf{a}\mathbf{a}^H \tag{5.3}$$

where $\mathbf{e}_k$ selects the column of $\mathbf{R}_{ss}$ corresponding to the reference microphone of node k, and where $P_s = E\{|s|^2\}$ is the power of the target speech source signal s.

Note that $\mathbf{R}_{yy}$ can be estimated using sample averaging during '*speech-and-noise*' segments. In addition, the noise correlation matrix, defined as $\mathbf{R}_{nn} = E\{\mathbf{n}\mathbf{n}^H\}$, is assumed to be either known or to be estimated in the '*noise-only*' segments when the target speech source is silent. To distinguish between such segments, a Voice Activity Detection (VAD) is required (as explained in [112, 118]). In the sequel, we use an overline to denote a correlation matrix that is estimated from the data, i.e., $\bar{\mathbf{R}}$.

When $\mathbf{s}$ and $\mathbf{n}$ are uncorrelated, we have $\mathbf{R}_{ss} = \mathbf{R}_{yy} - \mathbf{R}_{nn}$, where $\mathbf{R}_{ss}$ is a rank-1 matrix (see (5.3)). In practice, however, the rank of $\bar{\mathbf{R}}_{ss} = \bar{\mathbf{R}}_{yy} - \bar{\mathbf{R}}_{nn}$ is often greater than one, which is due to the finite window size in the short-time Fourier transform (STFT) analysis and/or nonstationarity of the noise. Moreover, in low-SNR conditions, $\bar{\mathbf{R}}_{ss}$ may even lose its positive semi-definiteness, leading to suboptimal or even unstable filters [103]. A GEVD-based rank-1 approximation of $\bar{\mathbf{R}}_{ss}$ can be alternatively incorporated in the MWF solution (5.3) to increase the estimation performance in such cases (more discussion in [123], [103]).

A GEVD of the matrix pair $(\bar{\mathbf{R}}_{yy}, \bar{\mathbf{R}}_{nn})$ is defined as [122]

$$\bar{\mathbf{R}}_{yy}\hat{\mathbf{X}} = \bar{\mathbf{R}}_{nn}\hat{\mathbf{X}}\hat{\mathbf{\Lambda}} \quad \text{s.t} \quad \hat{\mathbf{X}}^H\bar{\mathbf{R}}_{nn}\hat{\mathbf{X}} = \mathbf{I}_M \tag{5.4}$$

where $\hat{\mathbf{\Lambda}}$ and $\hat{\mathbf{X}}$ contain the Generalized EigenVaLues (GEVLs) and their corresponding Generalized EigenVeCtors (GEVCs), respectively. It is assumed w.l.o.g. that the GEVLs are sorted in descending order and that the GEVCs are scaled such that their $\bar{\mathbf{R}}_{nn}$-weighted norm is 1 (as expressed in (6.6)). Assuming that $\bar{\mathbf{R}}_{nn}$ is invertible, the GEVD problem (6.6) is equivalent to a joint diagonalization of $\bar{\mathbf{R}}_{yy}$ and $\bar{\mathbf{R}}_{nn}$, i.e., it can be verified from (6.6) that

$$\bar{\mathbf{R}}_{yy} = \hat{\mathbf{Q}}\hat{\mathbf{\Lambda}}\hat{\mathbf{Q}}^H, \ \bar{\mathbf{R}}_{nn} = \hat{\mathbf{Q}}\hat{\mathbf{Q}}^H \tag{5.5}$$

where $\hat{\mathbf{Q}} = \hat{\mathbf{X}}^{-H}$, with $\hat{\mathbf{Q}}$ a full-rank $M \times M$ matrix (not necessarily orthogonal). We can then write $\bar{\mathbf{R}}_{ss} = \bar{\mathbf{R}}_{yy} - \bar{\mathbf{R}}_{nn} = \hat{\mathbf{Q}}\big(\hat{\mathbf{\Lambda}} - \mathbf{I}_M\big)\hat{\mathbf{Q}}^H$. Comparing this with (5.3), the GEVD-based rank-1 approximation of $\bar{\mathbf{R}}_{ss}$ can be computed as

$$\bar{\mathbf{R}}_{ss} = (\hat{\lambda} - 1)\hat{\mathbf{q}}\hat{\mathbf{q}}^H \tag{5.6}$$

where $\hat{\lambda}$ is the first (i.e., largest) GEVL of $(\bar{\mathbf{R}}_{yy}, \bar{\mathbf{R}}_{nn})$, and where $\hat{\mathbf{q}}$ denotes the first column of $\hat{\mathbf{Q}}$. By plugging (5.6) into (5.3), the GEVD-based estimate of the node-specific desired signal at node $k \in \mathcal{K}^{\text{MWF}}$ is $\hat{d}_k = \hat{\mathbf{w}}_k^H \mathbf{y}$ with

$$\hat{\mathbf{w}}_k = \bar{\mathbf{R}}_{yy}^{-1}\hat{\mathbf{q}}(\hat{\lambda} - 1)\hat{q}_{k,r}^* \tag{5.7}$$

where $\hat{q}_{k,r}$ denotes the entry of $\hat{\mathbf{q}}$ corresponding to the r-th microphone of node k, , i.e., $\hat{q}_{k,r}^* = \hat{\mathbf{q}}^H \mathbf{e}_k$, where superscript $*$ is the complex conjugate operator. Note that by plugging (5.5) in (5.7), it can be easily shown that $\hat{\mathbf{w}}_k$ in (5.7) is equal to $\hat{\mathbf{x}}$ up to a scaling, where $\hat{\mathbf{x}}$ is the first column of $\hat{\mathbf{X}}$. The node-specific desired signal at node $k \in \mathcal{K}^{\text{MWF}}$ is then estimated as $\hat{d}_k = \hat{\mathbf{w}}_k^H \mathbf{y}$.

5.3.2 MVDR

The objective for each node $k \in \mathcal{K}^{\text{MVDR}}$ is to design an M-dimensional beamformer $\hat{\mathbf{w}}_k$ that minimizes the output noise power, subject to a unity gain constraint in the target speech source direction, i.e.,

$$\hat{\mathbf{w}}_k = \min_{\mathbf{w}_k} E\big\{|\mathbf{w}_k^H \mathbf{n}|^2\big\} \quad \text{s.t.} \quad \mathbf{a}^H \mathbf{w}_k = 1. \tag{5.8}$$

In practice, due to the fact that often the network-wide steering vector $\mathbf{a}$ is either unknown or difficult to estimate, the constraint is replaced by $\hat{\mathbf{q}}^H \mathbf{w}_k = \hat{q}_{k,r}^*$ [134]. This replacement is essentially motivated by the fact that $\hat{\mathbf{q}}$ is an estimate for the steering vector $\mathbf{a}$ (up to an unknown scaling), as can be verified by comparing (5.6) with (5.3). With the latter constraint, the optimization problem (5.8) alternatively preserves the target speech signal as it impinges on the reference microphone of node k. The closed-form solution of (5.8) is then given by [134]

$$\hat{\mathbf{w}}_k = \frac{\mathbf{R}_{nn}^{-1}\hat{\mathbf{q}}}{\hat{\mathbf{q}}^H \mathbf{R}_{nn}^{-1}\hat{\mathbf{q}}}\hat{q}_{k,r}^* \tag{5.9}$$

in which by plugging (5.5), it can be easily shown that $\hat{\mathbf{w}}_k$ is equal to $\hat{\mathbf{x}}$ up to a scaling. The node-specific desired signal at node $k \in \mathcal{K}^{\text{MVDR}}$ is then estimated as $\hat{d}_k = \hat{\mathbf{w}}_k^H \mathbf{y}$.

5.3.3 DOA estimation

The objective for each node $k \in \mathcal{K}^{\text{DOA}}$ is to estimate its node-specific DOA θ_k from the network-wide M-channel microphone signal $\mathbf{y}$. To achieve this, an estimate of the local steering vector (up to a complex scalar) can be fed into a subspace-based DOA estimation method, such as MUSIC [114], or ESPRIT [115]. We first partition $\hat{\mathbf{q}}$ as $\hat{\mathbf{q}} = [\hat{\mathbf{q}}_1^T \dots \hat{\mathbf{q}}_K]^T$, where $\hat{\mathbf{q}}_k^T$ contains the M_k entries of $\hat{\mathbf{q}}$ corresponding to the local array of node k. Note that the node-specific DOA estimation at each node $k \in \mathcal{K}^{\text{DOA}}$ is then carried out only using $\hat{\mathbf{q}}_k$, which is due to the fact that the relative geometry between the nodes is unknown. Although this means that we only partially exploit the information, cooperation between nodes has led to an improved GEVD-based estimate of $\hat{\mathbf{q}}$ (and hence its sub-vectors $\hat{\mathbf{q}}_k$'s) [118]. The obtained DOA estimate is denoted as $\hat{\theta}_k$.

5.4 Distributed MDMT-based algorithm

In the proposed distributed algorithm, each node $k \in \mathcal{K}$ first fuses its M_k-channel microphone signal $\mathbf{y}_k$ into a single-channel signal $z_k = \mathbf{f}_k^H \mathbf{y}_k$ with an M_k-dimensional linear compressor $\mathbf{f}_k$ (which will be defined later, see (5.14)) , and then broadcasts z_k to all other nodes. As a result, the required per-node communication bandwidth is reduced by a factor of M_k, compared to the centralized approach.

Considering a K-channel signal $\mathbf{z} = [z_1 \dots z_K]^T$, $\mathbf{z}_{-k}$ denotes the vector $\mathbf{z}$ with z_k excluded. Assuming a fully-connected WASN, each node k then has access to a P_k-channel signal $\widetilde{\mathbf{y}}_k$ which is defined as $\widetilde{\mathbf{y}}_k = [\mathbf{y}_k^T \ \mathbf{z}_{-k}^T]^T$, with $P_k = M_k + (K-1)$. In the sequel, we use the $\widetilde{\cdot}$ notation for quantities that are computed based on the extended signal $\widetilde{\mathbf{y}}_k = \widetilde{\mathbf{s}}_k + \widetilde{\mathbf{n}}_k$. Moreover, the corresponding P_k-dimensional correlation matrix estimates at each node k are denoted as $\bar{\mathbf{R}}_{\tilde{y}_k \tilde{y}_k}$, $\bar{\mathbf{R}}_{\tilde{s}_k \tilde{s}_k}$ and $\bar{\mathbf{R}}_{\tilde{n}_k \tilde{n}_k}$.

At iteration i, node q is the only updating node, which uses a block of L samples to locally estimate the required correlation matrices. In the next iteration the updating node q is changed, and a new block of L samples (over a different time window) is used, which means that the iterations are spread out over time in a block-wise fashion. Similar to (6.6)-(5.5) node q then computes a local

GEVD on the matrix pencil $(\bar{\mathbf{R}}^i_{\tilde{y}_k\tilde{y}_k}, \bar{\mathbf{R}}^i_{\tilde{n}_k\tilde{n}_k})$, leading to P_q-dimensional matrices containing the local GEVCs and GEVLs denoted as $\widetilde{\mathbf{X}}^i_q$, $\widetilde{\mathbf{\Lambda}}_q$ (with ordered as GEVLs in (6.6)) and $\widetilde{\mathbf{Q}}^i_q$, respectively, where $\widetilde{\mathbf{Q}}^i_q = (\widetilde{\mathbf{X}}^i_q)^{-H}$. The iteration index i will be dropped in the sequel for conciseness. We also define $\widetilde{\mathbf{q}}_q$ as the first column of $\widetilde{\mathbf{Q}}_q$. Regarding the update procedure at the updating node q, we first consider the following three single-SP task cases:

- If all nodes were MWF nodes, i.e., if $\mathcal{K} = \mathcal{K}^{\text{MWF}}$, one could run the GEVD-based distributed adaptive node-specific signal estimation (DANSE) algorithm [123], in which all nodes sequentially perform the following operations (compare to (5.2),(5.7)).

$$\widetilde{\mathbf{w}}_q = \min_{\widetilde{\mathbf{w}}_q} E\{|d_q - \widetilde{\mathbf{w}}_q^H \widetilde{\mathbf{y}}_q|^2\} \tag{5.10}$$

$$\widetilde{\mathbf{w}}_q = \bar{\mathbf{R}}^{-1}_{\tilde{y}_q\tilde{y}_q} \widetilde{\mathbf{q}}_q (\widetilde{\lambda}_q - 1)\widetilde{q}^*_{q,r} \tag{5.11}$$

  with $\widetilde{q}_{q,r}$ denoting the r-th entry of $\widetilde{\mathbf{q}}_q$ and $\widetilde{\lambda}_q$ equal to the largest GEVL in $\widetilde{\mathbf{\Lambda}}_q$.

- If all nodes were MVDR nodes, i.e., if $\mathcal{K} = \mathcal{K}^{\text{MVDR}}$, one could run the linearly constrained (LC-) DANSE algorithm [69], in which all nodes sequentially perform

$$\widetilde{\mathbf{w}}_q = \min_{\widetilde{\mathbf{w}}_q} E\{|\widetilde{\mathbf{w}}_q^H \widetilde{\mathbf{n}}_q|^2\} \quad \text{s.t.} \quad \widetilde{\mathbf{q}}_q^H \widetilde{\mathbf{w}}_q = \widetilde{q}^*_{q,r} \tag{5.12}$$

$$\widetilde{\mathbf{w}}_q = \frac{\bar{\mathbf{R}}^{-1}_{\tilde{n}_q\tilde{n}_q} \widetilde{\mathbf{q}}_q}{\widetilde{\mathbf{q}}_q^H \bar{\mathbf{R}}^{-1}_{\tilde{n}_q\tilde{n}_q} \widetilde{\mathbf{q}}_q} \widetilde{q}^*_{q,r}. \tag{5.13}$$

- If all nodes were DOA nodes, i.e., if $\mathcal{K} = \mathcal{K}^{\text{DOA}}$, one could run the algorithm in [120] to estimate the first M_q entries of $\widetilde{\mathbf{q}}_q$ corresponding to the local array at node q, defined as $\mathbf{q}_q$. As an estimate of the steering vector $\mathbf{a}_k$ (up to an unknown scaling), $\mathbf{q}_q$ is then fed into a subspace-based DOA estimation algorithm such as MUSIC or ESPRIT. The resulting DOA estimate is denoted as $\widetilde{\theta}_q$. Note that in this case node q updates an auxiliary vector $\widetilde{\mathbf{w}}_q$ via $\widetilde{\mathbf{w}}_q = \widetilde{\mathbf{x}}_q$, where $\widetilde{\mathbf{x}}_q$ is the first column of $\widetilde{\mathbf{X}}_q$, i.e., it is the largest principal GEVC of $(\bar{\mathbf{R}}^i_{\tilde{y}_k\tilde{y}_k}, \bar{\mathbf{R}}^i_{\tilde{n}_k\tilde{n}_k})$. This auxiliary parameter will define the linear compressor $\mathbf{f}_q$ (see (5.14)). The reason behind this choice is explained in [120], where it is shown that using (part of) the local principal GEVC as a compressor, eventually results in a local steering vector estimate $\mathbf{q}_q$ which is a subset of the network-wide steering vector $\hat{\mathbf{q}}$ (details omitted).

In all of the three aforementioned algorithms (i.e., [123], [69], [120]), the fusion rule is updated in a similar fashion, i.e., the updating node q updates $\mathbf{f}_q$ by replacing it with the first M_q rows of $\widetilde{\mathbf{w}}_q$, i.e.,

$$\mathbf{f}_q = [\mathbf{I}_{M_q}\mathbf{0}]\,\widetilde{\mathbf{w}}_q \tag{5.14}$$

where $\mathbf{I}_{M_q}$ is the M_q-dimensional identity matrix and $\mathbf{0}$ is an all-zero matrix with proper dimension. Nevertheless, note that the fusion update in (5.14) is different for each of these three single-SP cases, as the respective $\widetilde{\mathbf{w}}_q$'s are different. Using these ingredients, we now define the distributed MDMT algorithm case where we let the nodes use the fused signals of the other nodes, independent of how they locally update their $\widetilde{\mathbf{w}}_q$ with the fusion rule $\mathbf{f}_q$ based on GEVD-based DANSE, LC-DANSE or DOA estimation. The resulting distributed MDMT-based algorithm is described in Table 6.1. Note that nodes $k \in \{\mathcal{K}^{\text{MWF}} \bigcup \mathcal{K}^{\text{MVDR}}\}$ estimate their node-specific desired signal as $\widetilde{\mathbf{d}}_k = \widetilde{\mathbf{w}}_q^H \widetilde{\mathbf{y}}_q$ in each iteration. It is noted that the nodes are not aware of each others tasks, and hence perform the same operations as they would perform in a hypothetical homogeneous network where all the (other) nodes perform the same network-wide distributed algorithm (DANSE, LC-DANSE or DOA estimation). Remarkably, despite the fact that each node solves a different local task, it can be shown that all their local estimates converge to the corresponding centralized solution as if all nodes would have access to the microphone signals from all other nodes.

Theorem I: *If $\bar{\mathbf{R}}_{nn}$ is full rank, then the estimates obtained from the proposed distributed MDMT-based algorithm converge for any initialization of the fusion rules $\mathbf{f}_q$ to the corresponding estimates obtained from the centralized solutions, i.e., when $i \to \infty$, $\forall k \in \{\mathcal{K}^{MWF} \bigcup \mathcal{K}^{MVDR}\}$, $\widetilde{\mathbf{d}}_k = \hat{\mathbf{d}}_k$, and $\forall k \in \{\mathcal{K}^{DOA}\}$, $\widetilde{\theta}_k = \hat{\theta}_k$.*

We do not provide a rigorous proof here due to space constraints. The proof relies on the fact that, even though the $\widetilde{\mathbf{w}}_q$'s are estimated in a different manner at each node, there is an inherent compatibility, i.e., it can be shown that all the $\widetilde{\mathbf{w}}_q$'s represent a scaled version of the local GEVC $\widetilde{\mathbf{x}}_q$ (only for the DOA node, this link is explicit as we define $\widetilde{\mathbf{w}}_q$ as $\widetilde{\mathbf{x}}_q$). This makes the algorithm akin to Distributed Adaptive Covariance Generalized Eigenvector Estimation (DACGEE) [1], of which convergence and centralized optimality can be proven. The proof of Theorem I then follows from the convergence of the latter and the fact that the proposed algorithm is immune to the different scaling applied by each node's fusion rule $\mathbf{f}_k$.

Table 5.1: Distributed MDMT-based algorithm

1. Set $i \leftarrow 0$, $q \leftarrow 1$, and initialize all $\mathbf{f}_k^0$ and $\widetilde{\mathbf{w}}_k^0$, $\forall\, k \in \mathcal{K}$, with random entries.
2. Each node $k \in \mathcal{K}$ broadcasts L new observations of its single-channel compressed signal z_k^i:
$$z_k^i[iL+j] = \mathbf{f}_k^{i\,H}\mathbf{y}_k^i[iL+j], \qquad j = 1 \dots L \tag{5.15}$$
where the notation [.] denotes a time (STFT-frame) index.
3. **Updating node** q: first update $\bar{\mathbf{R}}^i_{\tilde{y}_q\tilde{y}_q}$ and $\bar{\mathbf{R}}^i_{\tilde{n}_q\tilde{n}_q}$ via sample averaging and then computes $\widetilde{\mathbf{X}}_q^i$ and $\widetilde{\mathbf{\Lambda}}_q^i$ from the GEVD of $(\bar{\mathbf{R}}^i_{\tilde{y}_q\tilde{y}_q}, \bar{\mathbf{R}}^i_{\tilde{n}_q\tilde{n}_q})$ from which $\widetilde{\mathbf{q}}_q^i$ is estimated.
   - if $q \in \mathcal{K}^{\mathrm{MWF}}$: compute the node-specific MWF $\widetilde{\mathbf{w}}_q^{i+1}$ as in (5.11).
   - if $q \in \mathcal{K}^{\mathrm{MVDR}}$: compute the node-specific MVDR $\widetilde{\mathbf{w}}_q^{i+1}$ as in (5.13).
   - if $q \in \mathcal{K}^{\mathrm{DOA}}$: use $\mathbf{q}_q^i$ and estimate the node-specific DOA $\widetilde{\theta}_q$, e.g., via ESPRIT or MUSIC and update $\widetilde{\mathbf{w}}_q^{i+1} = \widetilde{\mathbf{x}}_q^{i+1}$.
4. **Updating node** q: updates $\mathbf{f}_q^{i+1} = [\mathbf{I}_{M_q}\,\mathbf{0}]\,\widetilde{\mathbf{w}}_q^{i+1}$.
5. The other nodes $k \in \mathcal{K} \setminus q$ update their parameters as $\widetilde{\mathbf{w}}_k^{i+1} = \widetilde{\mathbf{w}}_k^i$ and $\mathbf{f}_k^{i+1} = \mathbf{f}_k^i$.
6. Each node $k \in \left\{\mathcal{K}^{\mathrm{MWF}} \bigcup \mathcal{K}^{\mathrm{MVDR}}\right\}$ estimates the next L samples of its single-channel signal d_k, as $\widetilde{\mathbf{d}}_k = (\widetilde{\mathbf{w}}_k^{i+1})^H\widetilde{\mathbf{y}}_k$. Each node $k \in \mathcal{K}^{\mathrm{DOA}} \setminus q$ keeps its latest node-specific DOA $\widetilde{\theta}_k$.
7. $i \leftarrow i+1$ and $q \leftarrow (q \mod K) + 1$ and return to step 2.

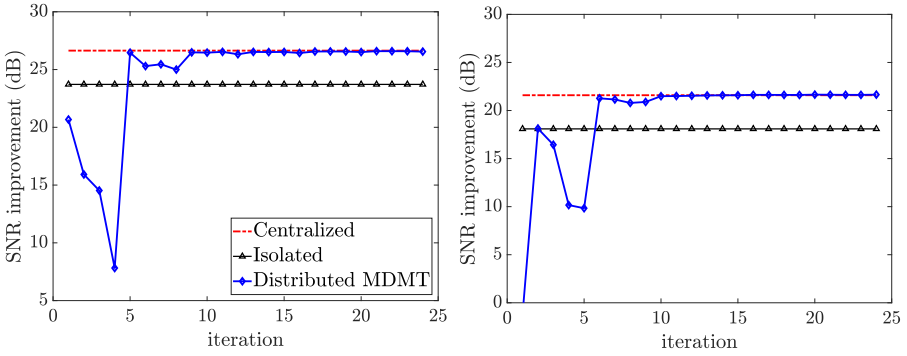
5.5 Numerical simulations

To investigate both the convergence and the performance of the proposed distributed MDMT-based algorithm, an acoustic scenario is simulated using the image method [152]. The room is rectangular 5 m × 5 m × 5 m, with reflection coefficients 0.2 for all surfaces. A WASN with 4 nodes ($K = 4$) is considered, where each node is equipped with a uniform linear array with 3 microphones ($M_k = 3, \forall k \in \mathcal{K}$) and where the inter-microphone distance is 10 cm. The target speech source produces seven speech sentences, with one second of silence between each two consecutive sentences. Four localized multi-talker noise sources (mutually uncorrelated) are placed in the room at the broadside direction of the nodes, with equal noise power. We use a sampling frequency of 16 kHz, a Hann-windowed DFT with window size 256 and with 50% overlaps. We assume a perfect VAD to exclude the effect of VAD errors. An uncorrelated white Gaussian noise is also added to each microphone signal to model the

microphone's self-noise and other possible isotropic noise contributions. The simulations are carried out in batch mode, which means that the signal statistics are estimated over the full signal length in each iteration. Nodes 1 and 3 are tasked with MWF, node 2 with MVDR and node 4 with DOA estimation (via wideband ESPRIT). As a performance measure, at MWF and MVDR nodes we utilize the SNR improvement (in dB), i.e., the difference between the input and the output SNR, and at the DOA node we use the absolute error of the estimates (in degrees). Figure A.1 illustrates both the convergence and the performance of the proposed distributed MDMT-based algorithm at all nodes. Cases where nodes estimates their node-specific tasks on their own, called as 'isolated', are added to also show the effectiveness of the algorithm. Results show that the estimates obtained with the distributed MDMT-based algorithm converges to the corresponding centralized estimates obtained in the centralized case.

5.6 Conclusion

We have studied a distributed multi-task problem in a WASN formed by three different groups of nodes. One of the groups is composed of nodes that aim at estimating the node-specific DOA of a desired speech source. The second and the third group are formed by nodes that aim at solving different node-specific speech enhancement problems by implementing either a MWF or a MDVR beamformer, respectively. We have derived a distributed algorithm that lets the nodes cooperate to attain the network-wide centralized solution of their estimation problems without any knowledge on the tasks solved by the other nodes. To do so, the proposed algorithm employs compressive filter-and-sum operations and a low-rank approximation of the sensor correlation matrix based on the GEVD. Finally, simulations have shown the efficiency of the proposed algorithm.



(a) Node 1: nodes-specific **MWF**.

(b) Node 2: node-specific **MVDR**.

(c) Node 3: nodes-specific **MWF**.

(d) Node 4: node-specific **DOA**.

Figure 5.1: Convergence of the distributed MDMT-based algorithm.

Chapter 6

Multi-task WSN for signal enhancement, LCMV beamforming and DOA estimation: multi-source case with node-specific constraints

Multi-Task wireless sensor network for joint distributed node-specific signal enhancement, LCMV beamforming and DOA estimation

Amin Hassani, Jorge Plata-Chaves, M. Hassan Bahari, Marc Moonen and Alexander Bertrand

Published in *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 3, Apr. 2017, pp. 518 - 533, DOI: 10.1109/JSTSP.2017.2676982.

Abstract

We consider a multi-task Wireless Sensor Network (WSN) where some of the nodes aim at applying a multi-channel Wiener filter to denoise their local sensor signals, while others aim at implementing a linearly constrained minimum variance beamformer to extract node-specific desired signals and cancel interfering signals, and again others aim at estimating the node-specific direction-of-arrival of a set of desired sources. For this multi-task WSN, by relying on distributed signal estimation techniques that incorporate a low-rank approximation of the desired signals correlation matrix, we design a distributed algorithm under which the nodes cooperate with reduced communication resources even though they are solving different signal processing tasks and do not know the tasks of the other nodes. Convergence and optimality results show that the proposed algorithm lets all the nodes achieve the network-wide centralized solution of their node-specific estimation problem. Finally, the algorithm is applied in a wireless acoustic sensor network scenario with multiple speech sources to show the effectiveness of the algorithm and support the theoretical results.

6.1 Introduction

A Wireless Sensor Network (WSN) consists of a collection of sensor nodes with sensing, processing, and wireless communication facilities. Often, these nodes are connected in an ad-hoc fashion and then cooperate to solve a Signal Processing (SP) task by means of a distributed SP algorithm. Such distributed algorithms avoid an energy-inefficient data centralization and allow to distribute the processing load over the different nodes. Traditionally, the design of distributed algorithms has focused on WSNs where all the nodes contribute to the same network-wide SP task and/or their sensor signals arise from the same data model [15]- [16]. However, resulting from the heterogeneity of the devices in the emerging Internet-of-Things (IoT), there is a need for a novel paradigm where the network is formed by Multiple Devices cooperating in Multiple Tasks (MDMT) [159], [153], [154]. We will refer to such multi-task WSNs as MDMT WSNs.

Unlike what is assumed in distributed parameter estimation algorithms for a single-task WSN (e.g. [75]- [78]), in MDMT WSNs the nodes may have a *Node-Specific Parameter Estimation* (NSPE) task, i.e., be interested in estimating different but inter-related parameters (e.g., [81]). In this setting, the design of distributed parameter estimation algorithms has relied on novel node-specific implementations of adaptive filtering techniques to facilitate the cooperation among nodes despite their different interests. For instance, based on novel extensions of the Least Mean Squares (LMS) and Recursive Least Squares (RLS) algorithm there are node-specific incremental [82], [83] and diffusion [84], [85] algorithms that solve a distributed parameter estimation problem where the parameters of interest are partially overlapping between the nodes. Similarly, based on regularized extensions of the LMS and the Affine Projection Algorithm (APA), several diffusion-based algorithms have also been derived to facilitate the cooperation among subsets of nodes with interests that are numerically similar [155]- [158]. In [118]- [120], the nodes cooperate with each other to estimate the node-specific Direction-of-Arrivals (DOAs) of a set of desired sources, which are node-specific due to the different position and orientation of the nodes.

Next to node-specific parameter estimation tasks, one can also consider MDMT WSNs where the nodes have a *Node-Specific Signal Estimation* (NSSE) task. Under different beamforming criteria and motivated by applications such as, e.g., Wireless Acoustic Sensor Networks (WASNs) [30] or body area networks [26], several distributed spatial filtering algorithms have been designed to solve NSSE problems where the nodes are interested in estimating desired signals that share a common latent signal subspace. For instance, based on the Multi-channel Wiener Filter (MWF), several distributed algorithms have been devised

to let the nodes obtain the centralized Linear Minimum Mean Square Error (LMMSE) estimate of their node-specific desired signals by exchanging linearly compressed versions of their sensor signals. These distributed algorithms have initially been designed for binaural hearing aids [34] and, afterwards, have been extended to WSNs with a fully-connected topology [67], a tree topology [71] and combinations thereof [106]. More recently, for non-stationary settings with low-SNR conditions, in [123] the estimation of node-specific desired signals has been performed by implementing different but coupled MWFs that incorporate a low-rank approximation based on a Generalized EigenValue Decomposition (GEVD). In addition to the MWF-based solutions, several distributed node-specific algorithms have been developed based on the Minimum Variance Distortionless Response (MVDR) criterion to estimate all the node-specific desired signals. Under the MVDR criterion, each node minimizes the output variance of a multi-channel spatial filter subject to a set of node-specific linear constraints to obtain a distortionless response for its desired signals [33]. For a setting where each device can have different desired signals as well as different interferers, a distributed node-specific implementation of a Linearly Constrained Minimum Variance (LCMV) beamformer has been proposed in [69]. In this case, generalizing the node-specific MVDR algorithms, the nodes cooperate with each other to minimize the output variance of their spatial filter with node-specific linear constraints, letting each node avoid distortion in its desired signals and cancel (fully or partially) its interferers.

To the best of our knowledge, all the aforementioned distributed node-specific estimation algorithms allow the cooperation among nodes interested in different but coupled versions of the *same* SP task (e.g., signal enhancement, spectrum estimation, DOA estimation etc.,). Furthermore, all the existing works assume that all nodes apply the *same* basic algorithm, e.g., a particular adaptive filter (e.g., LMS, RLS or APA), beamformer (e.g., MWF, MVDR, LCMV) or subspace-based DOA estimation method. Nonetheless, in heterogeneous networks, the nodes may have *different* interrelated SP tasks. Furthermore, each node may apply a different basic algorithm (filter or beamformer) in order to fulfill the particular performance requirements of its task. Motivated by these facts, by way of example, we consider an MDMT WSN where some of the nodes aim at applying an MWF to denoise their local sensor signals, while others aim at implementing an LCMV beamformer to extract node-specific desired signals and cancel interfering signals, and again others aim at estimating the node-specific DOAs of a set of desired sources. It is noted that in this MDMT WSN, the steering vectors are not known, and hence a GEVD-based MWF [103] and a GEVD-based LCMV [160] have been applied. Although a separate distributed algorithm has been proposed for each of these three SP tasks (MWF [123], LCMV [69], and DOA estimation [118]), we will show that they are inherently compatible, i.e., nodes from any of these three algorithms can connect to each

other, resulting in an MDMT WSN where each node indeed solves a different SP task. Nevertheless, we show that the nodes can still collaborate with significantly reduced communication resources, without even being aware of each other's SP task (be it MWF-based signal enhancement, LCMV beamforming or DOA estimation). Remarkably, as shown in the convergence and optimality proof of the proposed distributed MDMT algorithm, all the nodes can achieve the centralized solution of their corresponding node-specific estimation problem as if all nodes would have access to all sensor signals in the network. Finally, a validation of the theoretical expressions is provided via an application in a WASN scenario with multiple speech sources.

The chapter is organized as follows. In Section 6.2, the novel MDMT problem is mathematically formulated. In Section 6.3, the different estimation algorithms are explained, corresponding to the node-specific estimation problems. Section 6.4 is devoted to the derivation of the distributed MDMT algorithm that solves the MDMT problem of Section 6.2. Section 6.5 then provides the convergence and optimality analysis of the distributed MDMT algorithm. In Section 6.6, simulation results are provided. Finally, Section 6.7 summarizes the chapter.

6.2 Overview, data model and problem statement

6.2.1 Illustrative example

An example of an MDMT WASN is provided in Figure 6.1. Two speech sources as well as two noise sources are present in this scenario. Four nodes are considered, where each one is equipped with a microphone array and performs one of the following tasks:

1. an MWF which applies an LMMSE-based spatial filter to estimate the mixture of both speech signals as locally observed at its microphones, while suppressing the noise contributions.
2. an LCMV beamformer which applies a constrained spatial filter to extract the signal of one speech source while canceling out the signal of the other one.
3. a DOA estimation to estimate the DOAs with respect to both speech sources.

Note that tasks and interests of each node are node-specific, in the sense that (1) they inherently use different algorithms, (2) they may be interested in different

sources, and (3) they estimate the speech signals as locally observed at the microphones of the node. In the particular example of Figure 6.1, node 1 is an LCMV node and estimates the speech signal of source 1, while suppressing the speech signal of source 2. Node 4 is also an LCMV node, however with an interest opposite to node 1, i.e., it rejects the speech signal of source 1 and estimates the speech signal of source 2. Node 2 is an MWF node that treats both speech sources as desired sources and hence estimates the mixture of both speech signals in its microphone signals, while suppressing the noise contributions. Finally, node 3 estimates its node-specific DOAs of both speech sources.

6.2.2 Notation overview

In the sequel, a boldface capital letter, e.g., $\mathbf{Q}$, denotes a matrix quantity, while a boldface small letter, e.g., $\mathbf{q}$, denotes a vector quantity. Furthermore, a vector will be given a subscript to represent a certain column of a particular matrix, e.g., a vector $\mathbf{q}_1$ denotes the first column of the matrix $\mathbf{Q}$. The hat notation $(\hat{.})$ refers to *network-wide* centralized estimation, while the tilde notation $(\tilde{.})$ refers to *reduced-dimension* distributed estimation (their concrete meaning will be further explained later). The subscript $(.)_k$ indicates a node index and is used to refer to a quantity which can be equally applied to any general node k, while a notation with the node index $(.)_q$ is exclusively used to refer to the updating node q in the proposed MDMT algorithm. A blackboard notation will be used to denote a submatrix of a particular matrix, e.g., $\mathbb{Q}$ denotes a submatrix of $\mathbf{Q}$. This submatrix usually consists of the first S columns (unless stated otherwise), where S is the total number of signal sources. In addition, to refer to a projected version of a particular subspace (onto another subspace), a calligraphic letter will be used, e.g., $\boldsymbol{\mathcal{P}}$ denotes a projected version of the subspace $\mathbf{P}$ (see Section 6.3.2). Finally in few instances where a subscript is introduced to the node index k, a quantity corresponding to a particular source at node k is addressed, e.g., in $(.)_{k_s}$ a quantity at node k corresponding to source s is referred to.

6.2.3 Data model and problem statement

We consider a WSN with K multi-sensor nodes where a signal broadcast by a node can be received by all other $(K-1)$ nodes, i.e., the topology of the WSN is assumed to be fully-connected. It is noted that this fully-connected assumption is merely for the sake of an easy exposition, since all results can be extended to

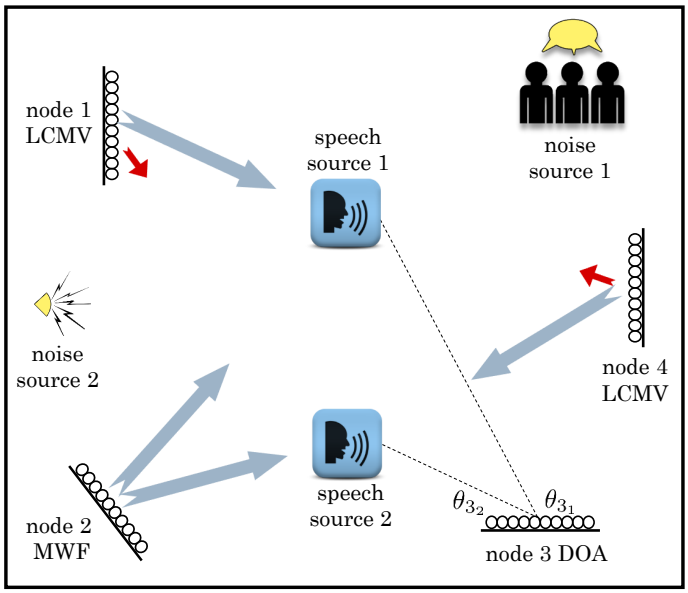


Figure 6.1: Example of an MDMT WASN.

networks with a nearest-neighbor topology[1], e.g., with a tree topology [71] or mixture of fully-connected and tree topologies [106]. For the extension of the algorithm to other topologies, we refer to the signal fusion strategies in [71, 106].

Each node $k \in \mathcal{K} = \{1, \ldots, K\}$ is equipped with a sensor array consisting of M_k sensors, where its (possibly complex-valued) M_k-channel sensor signal is denoted as $\mathbf{y}_k$, representing an M_k-variate stochastic variable. The sensor signal $\mathbf{y}_k$ is assumed to satisfy short-term stationarity and ergodicity conditions[2]. This structure can also be viewed as a hierarchical WSN where K master nodes collect sensor signals from M_k slave nodes each with a single sensor. It is assumed that the WSN observes S sources 'of interest' that are localized somewhere in space and mutually uncorrelated[3]. Each of these S sources can be considered as either a desired or interfering source, as explained later. The

[1]Note that this approach only works in certain topologies that allow to control the feedback paths in the network, such as, e.g., trees. Feedback components may harm convergence of the algorithm, because of similar reasons as explained in [71].

[2]By definition, a signal is short-term *ergodic* if over a finite segment, its time averages are equal to its statistical (ensemble) averages. This property allows the correlation matrices to be estimated via a time averaging. Moreover, the assumption of short-time *stationarity* ensures that the ensemble averages of a signal remain constant during a finite signal segment.

[3]To let the optimal solutions be defined in Section 6.3 and for notational convenience, we assume that S is known by all the nodes. However, this is without loss of generality and

M_k-channel sensor signal $\mathbf{y}_k$ can then be expressed by the linear model

$$\mathbf{y}_k = \mathbf{s}_k + \mathbf{n}_k = \mathbf{A}_k \check{\mathbf{s}} + \mathbf{n}_k \tag{6.1}$$

where $\check{\mathbf{s}}$ is the S-channel signal containing the S source signals and $\mathbf{n}_k$ is additive noise which includes both between-sensors correlated (due to e.g., the presence of localized noise sources) and uncorrelated (due to e.g., the sensor self-noise) noise contributions. In (6.1), $\mathbf{A}_k = [\mathbf{a}_{k_1}(\theta_{k_1}) \dots \mathbf{a}_{k_S}(\theta_{k_S})]$ is an unknown $M_k \times S$ complex-valued steering matrix, where $\mathbf{a}_{k_s}$ is the node-specific M_k-dimensional steering vector, and θ_{k_s} is the node-specific DOA for the s-th source, $s \in \{1, \dots, S\}$. Since each node is placed in a different location, and each node's sensor array has a different orientation, it is noted that nodes indeed observe different node-specific DOAs. It is also noted that since all signals in (6.1) are assumed to be complex-valued, this model also allows, e.g., convolutive time-domain mixtures, described as instantaneous per-frequency mixtures in the (short-term) Fourier transform domain.

Let $M = \sum_{k=1}^{K} M_k$, by stacking all $\mathbf{y}_k$, $\mathbf{n}_k$ and $\mathbf{s}_k$, we obtain the network-wide M-channel signals $\mathbf{y}$, $\mathbf{s}$ and $\mathbf{n}$, respectively, i.e.,

$$\mathbf{y} = \mathbf{s} + \mathbf{n} = \mathbf{A}\check{\mathbf{s}} + \mathbf{n} \tag{6.2}$$

where $\mathbf{A}$ is an $M \times S$ matrix, i.e., is a stacked version of all node-specific steering matrices $\mathbf{A}_k$.

Each node $k \in \mathcal{K}$ is then tasked with performing a *different* node-specific estimation problem from the following three cases: MWF-based signal enhancement, LCMV beamforming or DOA estimation, i.e., $\mathcal{K} \triangleq \mathcal{K}^{\mathrm{MWF}} \bigcup \mathcal{K}^{\mathrm{LCMV}} \bigcup \mathcal{K}^{\mathrm{DOA}}$.

- Each node $k \in \mathcal{K}^{\mathrm{MWF}}$ estimates the mixture of S source signals in $\check{\mathbf{s}}$ as they are locally observed at its sensors, i.e., it estimates the M_k-channel signal $\mathbf{s}_k = \mathbf{A}_k\check{\mathbf{s}}$, in the LMMSE sense (this will be formalized in Section 6.3.1). Hence each node $k \in \mathcal{K}^{\mathrm{MWF}}$ treats all S sources as desired sources and the signal estimation procedure in each node $k \in \mathcal{K}^{\mathrm{MWF}}$ preserves the node-specific spatial information in $\mathbf{s}_k$ while reducing the noise $\mathbf{n}_k$.

  Although the node-specific desired signals $\mathbf{s}_k$ are drawn from the same S-channel latent signal $\check{\mathbf{s}}$, it is noted that since $\check{\mathbf{s}}$ and $\mathbf{A}_k$ are both assumed to be unknown, the nodes are not aware of the relationship between their node-specific desired signals $\mathbf{s}_k$.

- Each node $k \in \mathcal{K}^{\mathrm{LCMV}}$ estimates D_k signals, corresponding to D_k desired sources, while the remaining $I_k = S - D_k$ sources are considered to be

Section 6.5.2 will explain how the cases when S is wrongly estimated affect the algorithm in practice (see Remark 3 in Section 6.5.2).

interferers. Each node $k \in \mathcal{K}^{\text{LCMV}}$ applies a spatial filter that minimizes the total output variance at its output, under a set of node-specific linear constraints such that the signals coming from the directions of the D_k desired sources are passed through without distortion, while signals coming from the directions of the I_k interfering sources are blocked [161].

- Each node $k \in \mathcal{K}^{\text{DOA}}$ estimates the node-specific DOAs for all S sources, i.e., it estimates $\theta_{k_1} \dots \theta_{k_S}$ [118]. It is assumed that each node only knows its own local sensor array geometry, but that the relative geometry with respect to the other nodes is unknown, i.e., the positions of the nodes and their orientations are unknown.

6.3 Centralized estimation algorithms

In this section, we first review the centralized estimation algorithm for each task, where each node $k \in \mathcal{K}$ would transmit its M_k-channel sensor signal $\mathbf{y}_k$ to all other nodes (or to a fusion center). As a result, each node can perform its node-specific task based on the network-wide M-channel sensor signal $\mathbf{y}$ defined in (6.2). In the sequel, the hat notation ($\hat{.}$) is always used to refer to centralized estimation.

6.3.1 Network-wide MWF

The goal of each node $k \in \mathcal{K}^{\text{MWF}}$ is to denoise its M_k-channel sensor signal, i.e., estimate $\mathbf{s}_k$ based on the network-wide M-channel sensor signal $\mathbf{y}$. To achieve this, node $k \in \mathcal{K}^{\text{MWF}}$ uses an $M \times M_k$ linear estimator $\hat{\mathbf{W}}_k$, which can be viewed as a network-wide spatial filter, to estimate $\mathbf{s}_k$ as $\hat{\mathbf{s}}_k = \hat{\mathbf{W}}_k^H \mathbf{y}$, where superscript H denotes the conjugate transpose operator. The MWF [112] computes $\hat{\mathbf{W}}_k$ based on the LMMSE criterion, i.e.,

$$\hat{\mathbf{W}}_k = \underset{\mathbf{W}_k}{\operatorname{argmin}} \, E\Big\{ \big\| \mathbf{s}_k - \mathbf{W}_k^H \mathbf{y} \big\|^2 \Big\} \tag{6.3}$$

where $E\{.\}$ is the expected value operator. Assuming $\mathbf{R}_{yy} = E\{\mathbf{y}\mathbf{y}^H\}$ has full rank, the unique solution of (6.3) is [112]:

$$\hat{\mathbf{W}}_k = \mathbf{R}_{yy}^{-1} \mathbf{R}_{ss} \mathbf{E}_k \tag{6.4}$$

where $\mathbf{R}_{ss} = E\{\mathbf{s}\mathbf{s}^H\}$, and where $\mathbf{E}_k = [\mathbf{0} \; \mathbf{I}_{M_k} \; \mathbf{0}]^T$ is an $M \times M_k$ matrix that selects the M_k columns of $\mathbf{R}_{ss}$ corresponding to the channels of $\mathbf{s}$ that are included in $\mathbf{s}_k$.

Assuming $\mathbf{s}$ and $\mathbf{n}$ are uncorrelated, based on (6.2) we can further write:

$$\mathbf{R}_{ss} = \mathbf{R}_{yy} - \mathbf{R}_{nn} = \mathbf{A}\mathbf{\Phi}\mathbf{A}^H \tag{6.5}$$

where $\mathbf{R}_{nn} = E\{\mathbf{n}\mathbf{n}^H\}$, and where $\mathbf{\Phi} = \text{diag}\{\phi_1, \ldots, \phi_S\}$ is an $S \times S$ diagonal matrix, where $\phi_s = E\{|\check{s}_s|^2\}$, with $\check{s}_s$ the s-th channel of $\check{\mathbf{s}}$. It is noted that $\mathbf{R}_{ss}$ is a rank-S matrix.

In practice, the sensor signal correlation matrix $\mathbf{R}_{yy}$ is generally estimated via sample averaging. The noise correlation matrix $\mathbf{R}_{nn}$ is known *a-priori* in some applications, otherwise can be estimated from the *'noise-only'* signal segments in scenarios when the desired sources have an ON-OFF behavior [118], [26]. In speech enhancement application for instance, a Voice Activity Detection (VAD) can be applied to distinguish between '*speech-and-noise*' and '*noise-only*' signal segments, from which $\mathbf{R}_{yy}$ and $\mathbf{R}_{nn}$ are estimated, respectively.

A straightforward method to estimate $\mathbf{R}_{ss}$ would be based on the subtraction $\mathbf{R}_{yy} - \mathbf{R}_{nn}$. The resulting estimate, however, generally has a rank greater than S and may not even be positive semi-definite in the presence of second order statistics errors, e.g., due to non-stationarity of the noise or VAD errors. Incorporating such poorly-estimated $\mathbf{R}_{ss}$ in the MWF solution (6.4) often results in a poor denoising performance of the MWF [103]. A rank-S representation of $\mathbf{R}_{ss}$ based on a GEVD of the estimates of $\mathbf{R}_{yy}$ and $\mathbf{R}_{nn}$ can be alternatively incorporated in the MWF solution (6.4) to deliver a superior performance [103]. The resulting MWF, which is referred to as GEVD-based MWF, will be explained in the rest of this section.

The GEVD of the ordered matrix pair $(\mathbf{R}_{yy}, \mathbf{R}_{nn})$ defines the generalized eigenvectors (GEVCs) $\hat{\mathbf{x}}_m$ $(m = 1 \ldots M)$ and corresponding generalized eigenvalues (GEVLs) $\hat{\lambda}_m$ as [122]

$$\mathbf{R}_{yy}\hat{\mathbf{X}} = \mathbf{R}_{nn}\hat{\mathbf{X}}\hat{\mathbf{L}} \quad \text{s.t.} \quad \hat{\mathbf{X}}^H\mathbf{R}_{nn}\hat{\mathbf{X}} = \mathbf{I}_M \tag{6.6}$$

where $\hat{\mathbf{X}} = [\hat{\mathbf{x}}_1 \ \ldots \ \hat{\mathbf{x}}_M]$ and $\hat{\mathbf{L}} = \text{diag}\{\hat{\lambda}_1 \ldots \hat{\lambda}_M\}$, and where $\mathbf{I}_M$ denotes the $M \times M$ identity matrix. In (6.6) it is assumed, without loss of generality (w.l.o.g.), that the GEVLs are sorted in descending order, with $\hat{\lambda}_1$ being the largest, and that the GEVCs are scaled such that their $\mathbf{R}_{nn}$-weighted norm is 1 (as expressed in right-hand-side of (6.6)). It can be shown that the GEVD in (6.6) extracts the directions with maximal SNR, similar to how principal component analysis extracts the directions with maximal variance [1, 122]. Assuming that $\mathbf{R}_{nn}$ is invertible, the GEVD problem (6.6) is equivalent to a joint diagonalization of $\mathbf{R}_{yy}$ and $\mathbf{R}_{nn}$ which can be written as

$$\mathbf{R}_{yy} = \hat{\mathbf{Q}}\hat{\mathbf{L}}\hat{\mathbf{Q}}^H, \ \mathbf{R}_{nn} = \hat{\mathbf{Q}}\hat{\mathbf{Q}}^H \tag{6.7}$$

where $\hat{\mathbf{Q}} = \hat{\mathbf{X}}^{-H}$ is a full-rank $M \times M$ matrix (not necessarily orthogonal). Based on (6.5) and (6.7), we obtain $\mathbf{R}_{ss} = \mathbf{R}_{yy} - \mathbf{R}_{nn} = \hat{\mathbf{Q}}(\hat{\mathbf{L}} - \mathbf{I}_M)\hat{\mathbf{Q}}^H$. Comparing this with (6.5), the GEVD-based rank-S representation of $\mathbf{R}_{ss}$ is given as

$$\mathbf{R}_{ss} = \hat{\mathbb{Q}}(\hat{\mathbb{L}} - \mathbf{I}_S)\hat{\mathbb{Q}}^H \tag{6.8}$$

where $\hat{\mathbb{L}}$ is an $S \times S$ diagonal matrix containing the first S diagonal elements of $\hat{\mathbf{L}}$ and where $\hat{\mathbb{Q}}$ is an $M \times S$ matrix containing the first S columns of $\hat{\mathbf{Q}}$, i.e.,

$$\hat{\mathbb{L}} = [\mathbf{I}_S \ \mathbf{0}]\,\hat{\mathbf{L}}\,[\mathbf{I}_S \ \mathbf{0}]^T \quad , \quad \hat{\mathbb{Q}} = \hat{\mathbf{Q}}\,[\mathbf{I}_S \ \mathbf{0}]^T . \tag{6.9}$$

By substituting (6.7) and (6.8) in (6.4), the network-wide GEVD-based estimate of the node-specific M_k-channel signal $\mathbf{s}_k$ at node $k \in \mathcal{K}^{\mathrm{MWF}}$ is obtained as $\hat{\mathbf{s}}_k = \hat{\mathbf{W}}_k^H \mathbf{y}$, with

$$\hat{\mathbf{W}}_k = \mathbf{R}_{yy}^{-1}\hat{\mathbb{Q}}(\hat{\mathbb{L}} - \mathbf{I}_S)\hat{\mathbb{Q}}^H\mathbf{E}_k. \tag{6.10}$$

6.3.2 Network-wide LCMV beamforming

At each node $k \in \mathcal{K}^{\mathrm{LCMV}}$, let ξ_k and τ_k denote the set of indices from $s \in \{1, \ldots, S\}$ corresponding to the D_k and I_k desired sources and interferers, respectively. As we assume that the steering matrix $\mathbf{A}$ is unknown, and since its estimation is difficult if the source signals are not known (even with a centralized algorithm), we will rely on the LCMV approach of [161], which relaxes the estimation of $\mathbf{A}$ to a more practical estimation problem. To this end, we define $\hat{\mathbb{Q}}_k^{\mathrm{D}}$ and $\hat{\mathbb{Q}}_k^{\mathrm{I}}$, where $\hat{\mathbb{Q}}_k^{\mathrm{D}}$ is an $M \times D_k$ matrix whose columns define a unitary basis for the desired sources subspace spanned by the columns of $\mathbf{A}$ with indices ξ_k, and where $\hat{\mathbb{Q}}_k^{\mathrm{I}}$ is an $M \times I_k$ matrix whose columns define a unitary basis for the interferers subspace spanned by the columns of $\mathbf{A}$ with indices τ_k. The resulting network-wide LCMV problem for each node $k \in \mathcal{K}^{\mathrm{LCMV}}$ is then defined as [69], [161]

$$\hat{\mathbf{w}}_k = \min_{\mathbf{w}_k} E\{|\mathbf{w}_k^H\mathbf{y}|^2\} \tag{6.11}$$

$$\text{s.t. } \hat{\mathbf{P}}_k^H\mathbf{w}_k = \hat{\mathbf{v}}_k \tag{6.12}$$

where $\hat{\mathbf{P}}_k \triangleq [\hat{\mathbb{Q}}_k^{\mathrm{D}} \ \ \hat{\mathbb{Q}}_k^{\mathrm{I}}]$ and where $\hat{\mathbf{v}}_k$ is the desired response vector, which defines the required response towards the desired sources (captured by $\hat{\mathbb{Q}}_k^{\mathrm{D}}$) and towards the interfering sources (captured by $\hat{\mathbb{Q}}_k^{\mathrm{I}}$). As $\hat{\mathbf{P}}_k$ only contains unitary bases for the two subspaces (rather than the columns of the actual steering matrix $\mathbf{A}$), the vector $\hat{\mathbf{v}}_k$ should be designed according to a specific strategy [161].

Defining $\hat{\mathbf{q}}_{k1}^{\mathrm{D}}$ as the column of $(\hat{\mathbb{Q}}_k^{\mathrm{D}})^H$ corresponding to the first (w.l.o.g.) sensor of node k, the choice of $\hat{\mathbf{v}}_k = [(\hat{\mathbf{q}}_{k1}^{\mathrm{D}})^T \ \mathbf{0}]^T$ leads to an LCMV beamformer $\hat{\mathbf{w}}_k$ in (6.11)-(6.12) that extracts the mixture of the desired source signals in ξ_k as they are observed at the first sensor of node k, while completely blocking the interfering signals in τ_k (a proof can be found in [134]). In some practical cases, however, it is favorable to let part of the interfering signals pass through as well without distortion (e.g., to preserve some directional characteristics of the interfering sources in hearing aid applications [162]). Hence we here consider the general case of [69, 162], defined as

$$\hat{\mathbf{v}}_k = \begin{bmatrix} \alpha\, \hat{\mathbf{q}}_{k1}^{\mathrm{D}} \\ \epsilon\, \hat{\mathbf{q}}_{k1}^{\mathrm{I}} \end{bmatrix} \tag{6.13}$$

where the user-defined gains $0 < \alpha \leq 1$ and $0 \leq \epsilon < 1$ control the output of the resulting LCMV beamformer, and where $\hat{\mathbf{q}}_{k1}^{\mathrm{I}}$ denotes the first (w.l.o.g.) column of $(\hat{\mathbb{Q}}_k^{\mathrm{I}})^H$ at node k. Note that the value of α and ϵ in (6.13) can be chosen differently at each node $k \in \mathcal{K}^{\mathrm{LCMV}}$, but their node index k is dropped for the sake of an easier notation.

The solution of the LCMV problem (6.11)-(6.13) is then given by [69], [134]

$$\hat{\mathbf{w}}_k = \mathbf{R}_{yy}^{-1}\hat{\mathbf{P}}_k\Big(\hat{\mathbf{P}}_k^H\mathbf{R}_{yy}^{-1}\hat{\mathbf{P}}_k\Big)^{-1}\hat{\mathbf{v}}_k. \tag{6.14}$$

The single-channel output at node $k \in \mathcal{K}^{\mathrm{LCMV}}$ is then given as $\hat{d}_k = \hat{\mathbf{w}}_k^H\mathbf{y}$.

In practice, to compute $\hat{\mathbb{Q}}_k^{\mathrm{D}}$ and $\hat{\mathbb{Q}}_k^{\mathrm{I}}$ from the network-wide M-channel sensor signal $\mathbf{y}$, the network-wide source-activity-based correlation matrices, denoted as $\mathbf{R}_{yy}^{\mathrm{D}_k}$ and $\mathbf{R}_{yy}^{\mathrm{I}_k}$, are first estimated via sample averaging. To achieve this, *'desired-sources-only'* samples, obtained from signal segments during which one or more of the D_k desired sources are active, are used to estimate $\mathbf{R}_{yy}^{\mathrm{D}_k}$. Likewise, *'interferers-only'* samples, obtained from signals segments during which one or more of the I_k interferers are active are used to estimate $\mathbf{R}_{yy}^{\mathrm{I}_k}$. Note that we made the implicit assumption here that the S sources have an ON-OFF behavior and that we know the activity of each source individually, possibly based on a source activity detection algorithm. This is a valid assumption for speech signals in WASNs (see Section 6.6), where speaker activity can be detected using multi-speaker VAD algorithms [163], [153]. Similar to (6.6), from a GEVD of $(\mathbf{R}_{yy}^{\mathrm{D}_k}, \mathbf{R}_{nn})$ and $(\mathbf{R}_{yy}^{\mathrm{I}_k}, \mathbf{R}_{nn})$ we have

$$\mathbf{R}_{yy}^{\mathrm{D}_k}\hat{\mathbf{X}}_k^{\mathrm{D}} = \mathbf{R}_{nn}\hat{\mathbf{X}}_k^{\mathrm{D}}\hat{\mathbf{L}}_k^{\mathrm{D}} \text{ s.t. } (\hat{\mathbf{X}}_k^{\mathrm{D}})^H\mathbf{R}_{nn}\hat{\mathbf{X}}_k^{\mathrm{D}} = \mathbf{I}_M \tag{6.15}$$

$$\mathbf{R}_{yy}^{\mathrm{I}_k}\hat{\mathbf{X}}_k^{\mathrm{I}} = \mathbf{R}_{nn}\hat{\mathbf{X}}_k^{\mathrm{I}}\hat{\mathbf{L}}_k^{\mathrm{I}} \text{ s.t. } (\hat{\mathbf{X}}_k^{\mathrm{I}})^H\mathbf{R}_{nn}\hat{\mathbf{X}}_k^{\mathrm{I}} = \mathbf{I}_M \tag{6.16}$$

where $\hat{\mathbf{X}}_k^{(.)}$ and $\hat{\mathbf{L}}_k^{(.)}$ are $M \times M$ matrices containing the GEVCs and GEVLs (in descending order as in (6.6)) of the matrix pair $(\mathbf{R}_{yy}^{(.)_k}, \mathbf{R}_{nn})$, respectively. By defining $\hat{\mathbf{Q}}_k^{\mathrm{D}} = (\hat{\mathbf{X}}_k^{\mathrm{D}})^{-H}$ and $\hat{\mathbf{Q}}_k^{\mathrm{I}} = (\hat{\mathbf{X}}_k^{\mathrm{I}})^{-H}$, the $\hat{\mathbb{Q}}_k^{\mathrm{D}}$ and $\hat{\mathbb{Q}}_k^{\mathrm{I}}$ are obtained from the first D_k and I_k columns of $\hat{\mathbf{Q}}_k^{\mathrm{D}}$ and $\hat{\mathbf{Q}}_k^{\mathrm{I}}$, respectively.

Note that, in theory the matrix $\hat{\mathbf{P}}_k = [\hat{\mathbb{Q}}_k^{\mathrm{D}}\ \hat{\mathbb{Q}}_k^{\mathrm{I}}]$ and the matrix $\hat{\mathbb{Q}}$ in (6.9) should have the same column space. In practice, however, because of discrepancies between the signal segments based on which the correlation matrices $\mathbf{R}_{yy}$, $\mathbf{R}_{yy}^{\mathrm{D}_k}$ and $\mathbf{R}_{yy}^{\mathrm{I}_k}$ are estimated, this generally does not hold true. The mismatch may in some scenarios become much more severe, especially when insufficient *'desired-sources-only'* and *'interferers-only'* samples are available to accurately estimate $\hat{\mathbb{Q}}_k^{\mathrm{D}}$ and $\hat{\mathbb{Q}}_k^{\mathrm{I}}$, respectively. In such scenarios, the LCMV solution (6.14) may often result in an inadequate beamforming output [116], [160]. Therefore, we apply the subspace projection-based approach of [160] such that the discarded samples associated with signal segments during which the desired sources and interferers are simultaneously active can also be exploited to enhance the estimation performance. To do so, $\hat{\mathbb{Q}}_k^{\mathrm{D}}$ and $\hat{\mathbb{Q}}_k^{\mathrm{I}}$ are projected onto the joint (larger) subspace spanned by the columns of $\hat{\mathbb{Q}}$, i.e., [160]

$$\hat{\mathbb{Q}}_k^{\mathcal{D}} \triangleq \mathrm{proj}_{\hat{\mathbb{Q}}}(\hat{\mathbb{Q}}_k^{\mathrm{D}}) = \hat{\mathbb{Q}}(\hat{\mathbb{Q}}^T\hat{\mathbb{Q}})^{-1}\hat{\mathbb{Q}}^T\hat{\mathbb{Q}}_k^{\mathrm{D}} \tag{6.17}$$

$$\hat{\mathbb{Q}}_k^{\mathcal{I}} \triangleq \mathrm{proj}_{\hat{\mathbb{Q}}}(\hat{\mathbb{Q}}_k^{\mathrm{I}}) = \hat{\mathbb{Q}}(\hat{\mathbb{Q}}^T\hat{\mathbb{Q}})^{-1}\hat{\mathbb{Q}}^T\hat{\mathbb{Q}}_k^{\mathrm{I}}. \tag{6.18}$$

By replacing $\hat{\mathbf{P}}_k$ in (6.14) with $\hat{\boldsymbol{\mathcal{P}}}_k \triangleq [\hat{\mathbb{Q}}_k^{\mathcal{D}}\ \hat{\mathbb{Q}}_k^{\mathcal{I}}]$, we obtain the following projection-based LCMV estimator

$$\hat{\mathbf{w}}_k = \mathbf{R}_{yy}^{-1}\hat{\boldsymbol{\mathcal{P}}}_k\big(\hat{\boldsymbol{\mathcal{P}}}_k^H\mathbf{R}_{yy}^{-1}\hat{\boldsymbol{\mathcal{P}}}_k\big)^{-1}\hat{\boldsymbol{v}}_k \tag{6.19}$$

where $\hat{\boldsymbol{v}}_k$ is the same as $\hat{\mathbf{v}}_k$ in (6.13), except that $\hat{\mathbf{q}}_{k1}^{\mathrm{D}}$ and $\hat{\mathbf{q}}_{k1}^{\mathrm{I}}$ are now drawn from $(\hat{\mathbb{Q}}_k^{\mathcal{D}})^H$ and $(\hat{\mathbb{Q}}_k^{\mathcal{I}})^H$, respectively.

It is emphasized that (6.17)-(6.18) ensures that $\hat{\boldsymbol{\mathcal{P}}}_k$ and $\hat{\mathbb{Q}}$ always have the same column space (assuming $\hat{\boldsymbol{\mathcal{P}}}_k$ is rank-S). This correction step has been demonstrated to substantially improve the performance of the LCMV beamformer [160].

6.3.3 Network-wide DOA estimation

The goal for each node $k \in \mathcal{K}^{\mathrm{DOA}}$ is to estimate all the S node-specific DOAs $\theta_{k_1} \ldots \theta_{k_S}$ from the network-wide M-channel sensor signal $\mathbf{y}$. It is assumed that each node only knows its own local sensor array geometry, but that the relative geometry with respect to the other nodes is unknown. Nevertheless, the

spatial coherence between the nodes can still be (partially) exploited [118]. To achieve this, first recall $\hat{\mathbb{Q}}$ as defined in (6.9), which is equal to the network-wide steering matrix $\mathbf{A}$ up to a column transformation (see (6.5), (6.8)). Now based on the partitioning of $\hat{\mathbb{Q}}$ as

$$\hat{\mathbb{Q}} \triangleq \begin{bmatrix} \hat{\mathbb{Q}}_1 \\ \vdots \\ \hat{\mathbb{Q}}_K \end{bmatrix} \tag{6.20}$$

where $\hat{\mathbb{Q}}_k$ is the $M_k \times S$ submatrix of $\hat{\mathbb{Q}}$ corresponding to M_k sensors at node k, each node $k \in \mathcal{K}^{\text{DOA}}$ can compute the node-specific DOAs by feeding $\hat{\mathbb{Q}}_k$ into a subspace-based DOA estimation method, e.g., MUSIC [114], or ESPRIT [115].

It is noted that only a part of $\hat{\mathbb{Q}}$ is used for DOA estimation at node k because the relative geometry between the nodes is indeed unknown. In practice, however, as $\hat{\mathbb{Q}}_k$ is extracted from the network-wide $\hat{\mathbb{Q}}$, the inherent spatial coherence between the nodes is exploited to improve the estimation compared to a purely local estimation [118].

The obtained node-specific DOA estimates at node $k \in \mathcal{K}^{\text{DOA}}$ based on $\hat{\mathbb{Q}}_k$ are denoted as $\hat{\theta}_{k_1} \dots \hat{\theta}_{k_S}$.

6.4 Distributed MDMT algorithm

In section 6.3, it has been assumed that each node $k \in \mathcal{K}$ performs its own node-specific estimation task based on the network-wide M-channel sensor signal $\mathbf{y}$. We now aim at designing an algorithm that lets each node $k \in \mathcal{K}$ obtain the same solution and performance of its corresponding centralized estimation algorithm in a distributed fashion over a fully-connected WSN. The computational burden of the nodes' individual tasks is then shared among the different nodes. Furthermore, each node $k \in \mathcal{K}$ only broadcasts an S-channel compressed signal to the other nodes, rather than its full M_k-channel sensor signal $\mathbf{y}_k$ (assuming $S < M_k, \forall k \in \mathcal{K}$)[4]. In the considered heterogeneous WSN, the nodes do not know each other's SP tasks, and hence perform the same operations as they would perform in a hypothetical homogeneous WSN where all the other nodes contribute to the same distributed algorithm to solve node-specific estimation problems corresponding to the same SP task (distributed MWF-based signal enhancement [123], distributed LCMV beamforming [69] or

[4]For the sake of an easier exposition, we will assume from now on that $S < M_k, \forall k \in \mathcal{K}$. If there exists a node k for which $S \geq M_k$, no compression is done at node k, and instead node k should simply broadcast its uncompressed sensor signal $\mathbf{y}_k$ to other nodes.

distributed DOA estimation [118]). Remarkably, despite the fact that each node $k \in \mathcal{K}$ solves a different SP task and is not aware of the SP tasks of other nodes, it will be shown in Section 6.5.2 that all the local estimates converge to their corresponding centralized solutions. Since the algorithm description below only includes the necessary ingredients of the underlying algorithms, for more details and intuitions on distributed GEVD-based MWF, LCMV and DOA estimation in a homogeneous WSN we refer to [123], [69] and [118], respectively.

In the proposed distributed MDMT algorithm, each node $k \in \mathcal{K}$ first compresses its M_k-channel sensor signal $\mathbf{y}_k$ into an S-channel compressed signal $\mathbf{z}_k^i = \mathbf{F}_k^{iH}\mathbf{y}_k$ with an $M_k \times S$ compression matrix $\mathbf{F}_k^i$ (which will be defined later, see (6.45)), where the superscript i is the iteration index. The compressed signal $\mathbf{z}_k^i$ is then broadcast to the other nodes, rather than $\mathbf{y}_k$, and hence the required per-node communication bandwidth is reduced by a factor of $\max\{(M_k/S), 1\}$. We will later explain how these compression matrices at the different nodes will be updated from iteration to iteration in a data-driven fashion. As the compression matrices change over time, also the statistics of the $\mathbf{z}_k^i$ signals will change over time, which means that all nodes will have to continuously track or re-compute the second-order statistics of the received data from other nodes. Eventually, we will show that the strategy to update the compression matrices will ensure that all of them converge to a stable setting, in which each node in the network obtains an optimal (i.e., network-wide) performance for its node-specific task as if it would have access to the raw uncompressed data of all the nodes. In particular, without accessing the network-wide sensor signal $\mathbf{y}$, each node $k \in \mathcal{K}^{\text{MWF}}$ obtains $\hat{\mathbf{s}}_k = \hat{\mathbf{W}}_k^H\mathbf{y}$ with $\hat{\mathbf{W}}_k$ given in (6.10), each node $k \in \mathcal{K}^{\text{LCMV}}$ obtains $\hat{d}_k = \hat{\mathbf{w}}_k^H\mathbf{y}$ with $\hat{\mathbf{w}}_k$ given in (6.19), and each node $k \in \mathcal{K}^{\text{DOA}}$ obtains the DOA estimates $\hat{\theta}_{k_1} \dots \hat{\theta}_{k_S}$.

Considering a KS-channel signal $\mathbf{z}^i = [\mathbf{z}_1^{iT} \dots \mathbf{z}_K^{iT}]^T$, let $\mathbf{z}_{-k}^i$ denote $\mathbf{z}^i$ with $\mathbf{z}_k^i$ removed. Assuming a fully-connected WSN, each node k then has access to a P_k-channel signal $\widetilde{\mathbf{y}}_k$ which is defined as $\widetilde{\mathbf{y}}_k^i = [\mathbf{y}_k^T\ \mathbf{z}_{-k}^{iT}]^T$, with $P_k = M_k + S(K-1)$. In the sequel, we use the tilde notation ($\widetilde{\cdot}$) for quantities that are computed based on the signal $\widetilde{\mathbf{y}}_k^i = \widetilde{\mathbf{s}}_k^i + \widetilde{\mathbf{n}}_k^i$. Moreover, the corresponding P_k-dimensional local correlation matrices at each node $k \in \mathcal{K}$ are denoted as $\mathbf{R}_{\tilde{y}_k\tilde{y}_k}^i$, $\mathbf{R}_{\tilde{s}_k\tilde{s}_k}^i$ and $\mathbf{R}_{\tilde{n}_k\tilde{n}_k}^i$. In practice, since these correlation matrices will change over time (due to changes in the scenario and due to updates of the compressor matrices), each node k estimates these local correlation matrices based on a block of N signal samples. In the next iteration, a new block of N samples (over a different time window) is used, which means that the iterations are spread out over time in a block-adaptive fashion, and that the compressed signal $\mathbf{z}_k^i = \mathbf{F}_k^{iH}\mathbf{y}_k$ is computed only once for each set of samples.

At iteration i of the distributed MDMT algorithm, node q is the only updating

node, where it re-computes its local compression matrix $\mathbf{F}_q^i$ based on the local correlation matrices $\mathbf{R}_{\tilde{y}_q\tilde{y}_q}$ and $\mathbf{R}_{\tilde{n}_q\tilde{n}_q}$. In the next iteration the updating node is changed. For conciseness, iteration index i will be mostly dropped in the sequel, unless when we want to explicitly refer to a specific iteration.

Similar to (6.6)-(6.7), each node $k \in \mathcal{K}$ then computes a local GEVD on the *reduced-dimension* matrix pencil $(\mathbf{R}_{\tilde{y}_k\tilde{y}_k}, \mathbf{R}_{\tilde{n}_k\tilde{n}_k})$ as

$$\mathbf{R}_{\tilde{y}_k\tilde{y}_k}\widetilde{\mathbf{X}}_k = \mathbf{R}_{\tilde{n}_k\tilde{n}_k}\widetilde{\mathbf{X}}_k\widetilde{\mathbf{L}}_k \quad \text{s.t.} \quad \widetilde{\mathbf{X}}_k^H\mathbf{R}_{\tilde{n}_k\tilde{n}_k}\widetilde{\mathbf{X}}_k = \mathbf{I}_{P_k} \tag{6.21}$$

where $\widetilde{\mathbf{X}}_k$, $\widetilde{\mathbf{L}}_k$ are P_k-dimensional matrices containing the local GEVCs and GEVLs (in descending order as in (6.6)), respectively. Here, we also define $\widetilde{\mathbb{X}}_k$ and $\widetilde{\mathbb{Q}}_k$ as the first S columns of $\widetilde{\mathbf{X}}_k$ and $\widetilde{\mathbf{Q}}_k$, respectively, where $\widetilde{\mathbf{Q}}_k = \widetilde{\mathbf{X}}_k^{-H}$. In addition, $\widetilde{\mathbb{L}}_k$ is defined as the $S \times S$ diagonal matrix containing the first S (largest) diagonal entries of $\widetilde{\mathbf{L}}_k$.

Each node $k \in \mathcal{K}$ subsequently completes its node-specific estimation depending on the SP task it is assigned with. In the next Subsection, we first explain these algorithms for the "homogeneous case" where all nodes have the same SP task.

6.4.1 Prelude: the homogeneous case

Distributed GEVD-based MWF

If all nodes were MWF nodes, i.e., if $\mathcal{K} = \mathcal{K}^{\text{MWF}}$, the GEVD-based Distributed Adaptive Node-specific Signal Estimation (DANSE) algorithm [123] could be used. At iteration i, all nodes $k \in \mathcal{K}$ then solve the following local LMMSE problem (compare to (6.3)):

$$\widetilde{\mathbf{W}}_k = \underset{\mathbf{W}_k}{\text{argmin}}\, E\Big\{\big\|\mathbf{s}_k - \mathbf{W}_k^H\widetilde{\mathbf{y}}_k\big\|^2\Big\} \tag{6.25}$$

where the solution is (compare to (6.4))

$$\widetilde{\mathbf{W}}_k = \mathbf{R}_{\tilde{y}_k\tilde{y}_k}^{-1}\mathbf{R}_{\tilde{s}_k\tilde{s}_k}\mathbf{E}_{M_k} \tag{6.26}$$

where $\mathbf{R}_{\tilde{y}_k\tilde{y}_k}$, $\mathbf{R}_{\tilde{n}_k\tilde{n}_k}$ and $\mathbf{R}_{\tilde{s}_k\tilde{s}_k}$ are the P_k-dimensional correlation matrix corresponding respectively to $\widetilde{\mathbf{y}}_k$, $\widetilde{\mathbf{n}}_k$ and $\widetilde{\mathbf{s}}_k$, and where $\mathbf{E}_{M_k}$ is a $P_k \times M_k$ matrix that selects the first M_k columns of $\mathbf{R}_{\tilde{s}_k\tilde{s}_k}$. Similar to (6.8), the GEVD-based rank-S representation of $\mathbf{R}_{\tilde{s}_k\tilde{s}_k}$ can be written as

$$\mathbf{R}_{\tilde{s}_k\tilde{s}_k} = \widetilde{\mathbb{Q}}_k(\widetilde{\mathbb{L}}_k - \mathbf{I}_S)\widetilde{\mathbb{Q}}_k^H \tag{6.27}$$

Table 6.1: Distributed MDMT algorithm

1. Set $i \leftarrow 0$, $q \leftarrow 1$, and initialize all $\mathbf{F}_k^0$ and $\widetilde{\mathbf{W}}_k^0$, $\forall\, k \in \mathcal{K}$, with random entries.

2. Each node $k \in \mathcal{K}$ broadcasts N new samples of its S-channel compressed signal $\mathbf{z}_k^i$:

$$\mathbf{z}_k^i[iN+j] = \mathbf{F}_k^{i\,H}\mathbf{y}_k^i[iN+j], \qquad j = 1 \dots N \tag{6.22}$$

   where the notation [.] denotes a sample index.

3. Each node $k \in \mathcal{K}$ first updates $\mathbf{R}^i_{\tilde{y}_k\tilde{y}_k}$ and $\mathbf{R}^i_{\tilde{n}_k\tilde{n}_k}$ via sample averaging using the samples at times $iN{+}1$ up to $(i{+}1)N$ and then computes the GEVD of $(\mathbf{R}^i_{\tilde{y}_k\tilde{y}_k}, \mathbf{R}^i_{\tilde{n}_k\tilde{n}_k})$ from which $\widetilde{\mathbb{X}}_k^i$, $\widetilde{\mathbb{L}}_k^i$ and $\widetilde{\mathbb{Q}}_k^i$ are obtained. Then:

   - if $k \in \mathcal{K}^{\mathrm{MWF}}$: compute $\mathbf{R}^i_{\tilde{s}_k\tilde{s}_k} = \widetilde{\mathbb{Q}}_k^i(\widetilde{\mathbb{L}}_k^i - \mathbf{I}_S)\widetilde{\mathbb{Q}}_k^{i\,H}$ and then use this to compute the local node-specific MWF estimator as

$$\widetilde{\mathbf{W}}_k^{i+1} = (\mathbf{R}^i_{\tilde{y}_k\tilde{y}_k})^{-1}\mathbf{R}^i_{\tilde{s}_k\tilde{s}_k}\mathbf{E}_{M_k} \tag{6.23}$$

   - if $k \in \mathcal{K}^{\mathrm{LCMV}}$: update $\mathbf{R}^{i\,\mathrm{D}}_{\tilde{y}_k\tilde{y}_k}$ and $\mathbf{R}^{i\,\mathrm{I}}_{\tilde{y}_k\tilde{y}_k}$ and then compute $\widetilde{\mathbb{Q}}_k^{i\,\mathcal{D}}$ and $\widetilde{\mathbb{Q}}_k^{i\,\mathcal{I}}$ based on (6.34)-(6.35). With $\widetilde{\boldsymbol{\mathcal{P}}}_k^i = [\widetilde{\mathbb{Q}}_k^{i\,\mathcal{D}}\ \ \widetilde{\mathbb{Q}}_k^{i\,\mathcal{I}}]$, the local node-specific LCMV estimator is then computed as

$$\widetilde{\mathbf{W}}_k^{i+1} = (\mathbf{R}^i_{\tilde{y}_k\tilde{y}_k})^{-1}\widetilde{\boldsymbol{\mathcal{P}}}_k^i\left((\widetilde{\boldsymbol{\mathcal{P}}}_k^i)^H(\mathbf{R}^i_{\tilde{y}_k\tilde{y}_k})^{-1}\widetilde{\boldsymbol{\mathcal{P}}}_k^i\right)^{-1}\widetilde{\boldsymbol{\mathcal{V}}}_k^i \tag{6.24}$$

   - if $k \in \mathcal{K}^{\mathrm{DOA}}$: use $\mathbb{Q}_k^i = \begin{bmatrix}\mathbf{I}_{M_k} & \mathbf{0}\end{bmatrix}\widetilde{\mathbb{Q}}_k^i$ and estimate the node-specific DOAs $\widetilde{\theta}^i_{k_1} \dots \widetilde{\theta}^i_{k_S}$, e.g., via ESPRIT [115] or MUSIC [114] and update $\widetilde{\mathbf{W}}_k^{i+1} = \widetilde{\mathbb{X}}_k^i$.

4. **Updating node** q: updates its compression matrix as $\mathbf{F}_q^{i+1} = \begin{bmatrix}\mathbf{I}_{M_q} & \mathbf{0}\end{bmatrix}\widetilde{\mathbf{W}}_q^{i+1}\begin{bmatrix}\mathbf{I}_S \\ \mathbf{0}\end{bmatrix}$.

5. Each node $k \in \mathcal{K}^{\mathrm{MWF}}$ estimates the next N samples of its M_k-channel output signal as $\widetilde{\mathbf{s}}_k^i[iN+j] = (\widetilde{\mathbf{W}}_k^{i+1})^H\widetilde{\mathbf{y}}_k^i[iN+j]$.

   Each node $k \in \mathcal{K}^{\mathrm{LCMV}}$ estimates the next N samples of its single-channel output signal as $\widetilde{d}_k^i[iN+j] = (\widetilde{\mathbf{w}}_k^{i+1})^H\widetilde{\mathbf{y}}_k^i[iN+j]$.

   Each node $k \in \mathcal{K}^{\mathrm{DOA}} \setminus q$ keeps its latest estimates $\widetilde{\theta}^{i_x}_{k_1} \dots \widetilde{\theta}^{i_x}_{k_S}$, with $i_x < i$.

6. $i \leftarrow i+1$ and $q \leftarrow (q \mod K) + 1$ and return to step 2.

where plugging (6.27) into the local MWF solution (6.26) gives

$$\widetilde{\mathbf{W}}_k = \mathbf{R}_{\tilde{y}_k\tilde{y}_k}^{-1}\widetilde{\mathbb{Q}}_k(\widetilde{\mathbb{L}}_k - \mathbf{I}_S)\widetilde{\mathbb{Q}}_k^H\mathbf{E}_{M_k}. \tag{6.28}$$

The estimate of $\mathbf{s}_k$ for each node $k \in \mathcal{K}$ at iteration i is then computed as $\widetilde{\mathbf{s}}_k = \widetilde{\mathbf{W}}_k^H\widetilde{\mathbf{y}}_k$.

Assuming node q is the updating node at iteration i, it updates its compression matrix $\mathbf{F}_q$ from the $M_q \times S$ upper-left submatrix of $\widetilde{\mathbf{W}}_q$, i.e.,

$$\mathbf{F}_q = [\mathbf{I}_{M_q}\ \mathbf{0}]\widetilde{\mathbf{W}}_q\begin{bmatrix}\mathbf{I}_S\\ \mathbf{0}\end{bmatrix}. \tag{6.29}$$

Distributed GEVD-based LCMV

If all nodes were LCMV nodes, i.e., if $\mathcal{K} = \mathcal{K}^{\text{LCMV}}$, the Linearly Constrained (LC)-DANSE algorithm [69] could be used. At iteration i, all nodes $k \in \mathcal{K}$ first update their source-activity-based P_k-dimensional correlation matrices $\mathbf{R}_{\tilde{y}_k\tilde{y}_k}^{\text{D}}$ and $\mathbf{R}_{\tilde{y}_k\tilde{y}_k}^{\text{I}}$ corresponding to $\widetilde{\mathbf{y}}_k$, and then compute $\widetilde{\mathbb{Q}}_k^{\text{D}}$ and $\widetilde{\mathbb{Q}}_k^{\text{I}}$ via a GEVD of $(\mathbf{R}_{\tilde{y}_k\tilde{y}_k}^{\text{D}}, \mathbf{R}_{\tilde{n}_k\tilde{n}_k})$ and $(\mathbf{R}_{\tilde{y}_k\tilde{y}_k}^{\text{I}}, \mathbf{R}_{\tilde{n}_k\tilde{n}_k})$, respectively (similar to (6.15)-(6.16)). With $\widetilde{\mathbf{P}}_k \triangleq [\widetilde{\mathbb{Q}}_k^{\text{D}}\ \ \widetilde{\mathbb{Q}}_k^{\text{I}}]$, each node $k \in \mathcal{K}$ subsequently solves the local LCMV problem

$$\widetilde{\mathbf{W}}_k = \min_{\mathbf{W}_k} E\{\|\mathbf{W}_k^H\widetilde{\mathbf{y}}_k\|^2\} \tag{6.30}$$

$$\text{s.t. } \widetilde{\mathbf{P}}_k^H\mathbf{W}_k = \widetilde{\mathbf{V}}_k \tag{6.31}$$

where $\widetilde{\mathbf{V}}_k$ is defined as [69]

$$\widetilde{\mathbf{V}}_k = \begin{bmatrix}\alpha\widetilde{\mathbf{q}}_{k1}^{\text{D}} & \eta_1\widetilde{\mathbf{q}}_{k2}^{\text{D}} & \dots & \eta_{(S-1)}\widetilde{\mathbf{q}}_{kS}^{\text{D}}\\ \epsilon\widetilde{\mathbf{q}}_{k1}^{\text{I}} & \beta_1\widetilde{\mathbf{q}}_{k2}^{\text{I}} & \dots & \beta_{(S-1)}\widetilde{\mathbf{q}}_{kS}^{\text{I}}\end{bmatrix} \tag{6.32}$$

where $\widetilde{\mathbf{q}}_{kj}^{\text{D}}$ and $\widetilde{\mathbf{q}}_{kj}^{\text{I}}$ denote the j-th column of $(\widetilde{\mathbb{Q}}_k^{\text{D}})^H$ and $(\widetilde{\mathbb{Q}}_k^{\text{I}})^H$, respectively, and where α and ϵ are the same as in (6.13), and where $\eta_j \in \mathbb{C}$ and $\beta_j \in \mathbb{C}$ can be chosen randomly[5], as long as the resulting $\widetilde{\mathbf{V}}_k$ is full rank [69]. Note that (6.30)-(6.32) extends the LCMV beamformer with $S-1$ auxiliary LCMV beamformers, which will be used to create a compression matrix $\mathbf{F}_q$ with S columns (see (6.37)). In [69], it is explained that this is necessary in order to embed the full S-dimensional signal subspace in the compressed signal $\mathbf{z}_q$, which facilitates the convergence of the LC-DANSE algorithm to the centralized

[5]The blackboard notation $\mathbb{C}$ here denotes the set of all complex numbers.

LCMV solution (details omitted). The solution to the local LCMV problem (6.30)-(6.32) is then given by

$$\widetilde{\mathbf{W}}_k = \mathbf{R}_{\tilde{y}_k\tilde{y}_k}^{-1}\widetilde{\mathbf{P}}_k\Big(\widetilde{\mathbf{P}}_k^H\mathbf{R}_{\tilde{y}_k\tilde{y}_k}^{-1}\widetilde{\mathbf{P}}_k\Big)^{-1}\widetilde{\mathbf{V}}_k. \tag{6.33}$$

Similar to (6.17)-(6.18), in order to improve the estimation performance in practice, we define the following projection-based matrices

$$\widetilde{\mathbb{Q}}_k^{\mathcal{D}} \triangleq \text{proj}_{\widetilde{\mathbb{Q}}_k}(\widetilde{\mathbb{Q}}_k^{\text{D}}) = \widetilde{\mathbb{Q}}_k(\widetilde{\mathbb{Q}}_k^T\widetilde{\mathbb{Q}}_k)^{-1}\widetilde{\mathbb{Q}}_k^T\widetilde{\mathbb{Q}}_k^{\text{D}} \tag{6.34}$$

$$\widetilde{\mathbb{Q}}_k^{\mathcal{I}} \triangleq \text{proj}_{\widetilde{\mathbb{Q}}_k}(\widetilde{\mathbb{Q}}_k^{\text{I}}) = \widetilde{\mathbb{Q}}_k(\widetilde{\mathbb{Q}}_k^T\widetilde{\mathbb{Q}}_k)^{-1}\widetilde{\mathbb{Q}}_k^T\widetilde{\mathbb{Q}}_k^{\text{I}}. \tag{6.35}$$

Replacing $\widetilde{\mathbf{P}}_k$ in (6.33) with $\widetilde{\boldsymbol{\mathcal{P}}}_k \triangleq [\widetilde{\mathbb{Q}}_k^{\mathcal{D}}\ \ \widetilde{\mathbb{Q}}_k^{\mathcal{I}}]$, and using the first S columns of $(\widetilde{\mathbb{Q}}_k^{\mathcal{D}})^H$ and $(\widetilde{\mathbb{Q}}_k^{\mathcal{I}})^H$ in (6.32), we obtain the following projection-based LCMV estimator

$$\widetilde{\mathbf{W}}_k = \mathbf{R}_{\tilde{y}_k\tilde{y}_k}^{-1}\widetilde{\boldsymbol{\mathcal{P}}}_k\Big(\widetilde{\boldsymbol{\mathcal{P}}}_k^H\mathbf{R}_{\tilde{y}_k\tilde{y}_k}^{-1}\widetilde{\boldsymbol{\mathcal{P}}}_k\Big)^{-1}\widetilde{\boldsymbol{\mathcal{V}}}_k. \tag{6.36}$$

The output signal at each node $k \in \mathcal{K}$ is then computed as $\widetilde{d}_k = \widetilde{\mathbf{w}}_k^H\widetilde{\mathbf{y}}_k$, with $\widetilde{\mathbf{w}}_k$ denoting the first column of $\widetilde{\mathbf{W}}_k$.

Assuming node q is the updating node at iteration i, it updates its compression matrix $\mathbf{F}_q$ as

$$\mathbf{F}_q = [\mathbf{I}_{M_q}\ \mathbf{0}]\widetilde{\mathbf{W}}_q. \tag{6.37}$$

It has been shown in [69] that the LC-DANSE algorithm using (6.36) converges to an equilibrium point where each node k computes the S-channel output of the extended network-wide LCMV beamformer

$$\hat{\mathbf{W}}_k = \mathbf{R}_{yy}^{-1}\hat{\boldsymbol{\mathcal{P}}}_k\big(\hat{\boldsymbol{\mathcal{P}}}_k^H\mathbf{R}_{yy}^{-1}\hat{\boldsymbol{\mathcal{P}}}_k\big)^{-1}\hat{\boldsymbol{\mathcal{V}}}_k \tag{6.38}$$

where the $S \times S$ matrix of desired responses $\hat{\boldsymbol{\mathcal{V}}}_k$ is defined as

$$\hat{\boldsymbol{\mathcal{V}}}_k = \begin{bmatrix} \alpha\hat{\mathbf{q}}_{k1}^{\mathcal{D}} & \eta_1\hat{\mathbf{q}}_{k2}^{\mathcal{D}} & \dots & \eta_{(S-1)}\hat{\mathbf{q}}_{kS}^{\mathcal{D}} \\ \epsilon\hat{\mathbf{q}}_{k1}^{\mathcal{I}} & \beta_1\hat{\mathbf{q}}_{k2}^{\mathcal{I}} & \dots & \beta_{(S-1)}\hat{\mathbf{q}}_{kS}^{\mathcal{I}} \end{bmatrix} \tag{6.39}$$

where $\hat{\mathbf{q}}_{kj}^{\mathcal{D}}$ and $\hat{\mathbf{q}}_{kj}^{\mathcal{I}}$ denote the column of $(\hat{\mathbb{Q}}_k^{\mathcal{D}})^H$ and $(\hat{\mathbb{Q}}_k^{\mathcal{I}})^H$ corresponding to the j-th sensor of node k, respectively. Note that this again extends the original centralized network-wide LCMV beamformer (6.14) with $S-1$ auxiliary LCMV beamformers. Nevertheless, since the first column of $\hat{\boldsymbol{\mathcal{V}}}_k$ is the same as the

desired response vector $\hat{\boldsymbol{v}}_k$ defined earlier in (6.19) for the centralized case, the first column of the extended $\hat{\mathbf{W}}_k$ in (6.38) is equal to the centralized LCMV beamformer defined in (6.13).

At each node $k \in \mathcal{K}^{\mathrm{LCMV}}$, note that the projections (6.17)-(6.18) ensure that

$$\hat{\boldsymbol{\mathcal{P}}}_k \triangleq [\hat{\mathbb{Q}}_k^{\mathcal{D}} \;\; \hat{\mathbb{Q}}_k^{\mathcal{I}}] = \hat{\mathbb{Q}}\hat{\boldsymbol{\Delta}}_k \tag{6.40}$$

with $\hat{\boldsymbol{\Delta}}_k$ an $S \times S$ column transformation matrix (its definition follows from comparing (6.40) with (6.17)-(6.18)). This allows to rewrite (6.39) as

$$\hat{\boldsymbol{\mathcal{V}}}_k = \begin{bmatrix} \big(\hat{\mathbb{Q}}_k^{\mathcal{D}}\big)^H [\boldsymbol{\alpha}\; \mathbf{0}]^T \\ \big(\hat{\mathbb{Q}}_k^{\mathcal{I}}\big)^H [\boldsymbol{\epsilon}\; \mathbf{0}]^T \end{bmatrix} = \hat{\boldsymbol{\Delta}}_k^H \begin{bmatrix} [\mathbf{I}_{\mathrm{D}_k}\; \mathbf{0}] \hat{\mathbb{Q}}_k^H [\boldsymbol{\alpha}\; \mathbf{0}]^T \\ [\mathbf{0}\; \mathbf{I}_{\mathrm{I}_k}] \hat{\mathbb{Q}}_k^H [\boldsymbol{\epsilon}\; \mathbf{0}]^T \end{bmatrix} \tag{6.41}$$

where the $S \times S$ diagonal matrices $\boldsymbol{\alpha}$ and $\boldsymbol{\epsilon}$ are defined as $\boldsymbol{\alpha} \triangleq \mathrm{diag}\{\alpha, \eta_1, \ldots, \eta_{(S-1)}\}$ and $\boldsymbol{\epsilon} \triangleq \mathrm{diag}\{\epsilon, \beta_1, \ldots, \beta_{(S-1)}\}$, and where $\mathbf{I}_{\mathrm{D}_k}$ and $\mathbf{I}_{\mathrm{I}_k}$ are identity matrices of dimensions D_k and I_k, respectively.

Distributed GEVD-based DOA estimation

If all nodes were DOA nodes, i.e., if $\mathcal{K} = \mathcal{K}^{\mathrm{DOA}}$, the distributed signal subspace algorithm of [120] could be used to iteratively estimate $\mathbb{Q}_k$ at each node k, where $\mathbb{Q}_k$ is defined as the $M_k \times S$ upper-left submatrix of $\widetilde{\mathbf{Q}}_k = (\widetilde{\mathbf{X}}_k)^{-H}$, i.e.,

$$\mathbb{Q}_k \triangleq [\mathbf{I}_{M_k}\; \mathbf{0}]\, \widetilde{\mathbf{Q}}_k\, [\mathbf{I}_S\; \mathbf{0}]^T. \tag{6.42}$$

The estimated $\mathbb{Q}_k$ is then fed into a subspace-based DOA estimation algorithm such as MUSIC [114] or ESPRIT [115]. The resulting DOA estimates at iteration i of the distributed algorithm are denoted as $\widetilde{\theta}_{k_1} \ldots \widetilde{\theta}_{k_S}$. In this case, the underlying distributed algorithm (described in [120]) essentially requires each node k to locally update an auxiliary estimator defined as (details omitted)

$$\widetilde{\mathbf{W}}_k = \widetilde{\mathbb{X}}_k \tag{6.43}$$

where the columns of $\widetilde{\mathbb{X}}_k$ were earlier defined as the S principal GEVCs of the matrix pencil $(\mathbf{R}_{\tilde{y}_k \tilde{y}_k}, \mathbf{R}_{\tilde{n}_k \tilde{n}_k})$. Finally, the compression matrix $\mathbf{F}_q$ at the updating node q is updated as

$$\mathbf{F}_q = [\mathbf{I}_{M_q}\; \mathbf{0}]\, \widetilde{\mathbf{W}}_q. \tag{6.44}$$

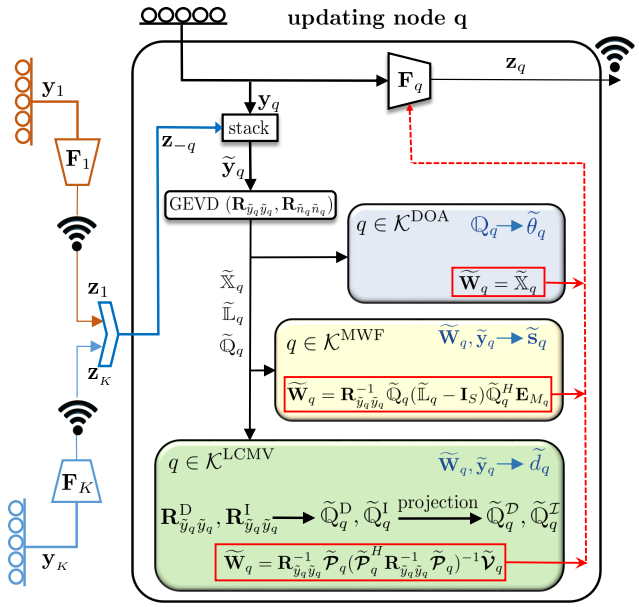


Figure 6.2: Block scheme for an updating node q in the distributed MDMT algorithm.

6.4.2 The heterogeneous case

In the three "homogeneous case" algorithms outlined in Subsection 6.4.1, when the updating node q has solved its own (three times different) node-specific estimation problem, its compression matrix $\mathbf{F}_q$ is updated using (three times) the same strategy. In particular, $\mathbf{F}_q$ in the GEVD-based DANSE [123], LC-DANSE [69] and distributed DOA estimation algorithm [118] is always chosen as the first M_q rows and S columns of the respective $\widetilde{\mathbf{W}}_q$ (see (6.29),(6.37),(6.44)), i.e.,

$$\mathbf{F}_q = [\mathbf{I}_{M_q}\ \mathbf{0}]\,\widetilde{\mathbf{W}}_q \begin{bmatrix}\mathbf{I}_S \\ \mathbf{0}\end{bmatrix}. \tag{6.45}$$

It is emphasized, however, that the corresponding estimators $\widetilde{\mathbf{W}}_q$ expressed in (6.28), (6.36), (6.43) are different, hence the compression matrices as well as the compressed signals $\mathbf{z}_q$ are different in the three cases.

We can now propose the distributed MDMT algorithm in which each node $k \in \mathcal{K}$ exploits the compressed signals $\mathbf{z}_n$ of the other nodes $n \in \mathcal{K} \setminus k$ for its own SP task. This will be done independent of how the compression matrices $\mathbf{F}_k$ have been generated. The distributed MDMT algorithm is described in

Table 6.1. In addition, Figure 6.2 provides a block scheme for an updating node q in the algorithm. When comparing Table 6.1 with the three "homogeneous case" algorithms described in Subsection 6.4.1, it is noted that each node $k \in \mathcal{K}$ performs the same operations as in a hypothetical homogeneous network where all the other nodes also have the same SP task as node k. Remarkably, it will be shown in Section 6.5.2 that the algorithm lets all the nodes achieve the network-wide centralized solution of their node-specific estimation problem as if all nodes would have access to the network-wide sensor signal $\mathbf{y}$.

In the final step in each iteration of the distributed MDMT algorithm, nodes $k \in \mathcal{K}^{\mathrm{MWF}}$ estimate their node-specific M_k-channel output as $\widetilde{\mathbf{s}}_k = \widetilde{\mathbf{W}}_k^H \widetilde{\mathbf{y}}_k$ in each iteration, nodes $k \in \mathcal{K}^{\mathrm{LCMV}}$ estimate their node-specific single-channel output as $\widetilde{d}_k = \widetilde{\mathbf{w}}_k^H \widetilde{\mathbf{y}}_k$, while the nodes $k \in \mathcal{K}^{\mathrm{DOA}}$ keep their latest node-specific DOAs until their next updating turn.

Remark 1: Note that each node in each iteration i of the distributed MDMT algorithm fulfills its task using a set of operations *identical* to those of its corresponding centralized realization describe in Section III, except that in the former the operations are done on the local *reduced-dimension* matrices. In particular, each node k has a per-node computational complexity of $\mathcal{O}((M_k + S(K-1))^3)$ per update, compared to its centralized complexity of $\mathcal{O}(M^3)$. It is emphasized that this is in fact achieved through the communication bandwidth reduction with a factor M_k/S due to transmitting and receiving the compressed signals $\mathbf{z}_k$ (rather than $\mathbf{y}_k$ in the centralized realization). These reductions in terms of the per-node communication cost and computational complexity come at the cost of having a slower tracking and adaptation performance. As will be discussed in the next section, after convergence of the compression matrices at each node, all the nodes achieve the network-wide centralized solution of their node-specific estimation problem.

6.5 Convergence and optimality

In this section, we first provide some fundamental details about the parameterization and the solution space of the distributed MDMT algorithm. In particular, each node $k \in \mathcal{K}$, defines its own network-wide solution of its own task, i.e., by (6.10), (6.38) or (6.20) if $k \in \mathcal{K}^{\mathrm{MWF}}$, $k \in \mathcal{K}^{\mathrm{LCMV}}$ or $k \in \mathcal{K}^{\mathrm{DOA}}$, respectively. It will be shown in Subsection 6.5.1 that the network-wide solutions of all the different nodes all lie in the parameterized solution space as defined by the fusion structure of the distributed algorithm (see (6.48)). The results of Subsection 6.5.1 will then be used in Subsection 6.5.2 to prove the convergence and optimality of the proposed algorithm.

6.5.1 Parameterization of the solution space

We define $\mathbf{H}_k$ and $\mathbf{G}_{k-k}$ as the first M_k rows and the last $S(K-1)$ rows of the the local estimator $\widetilde{\mathbf{W}}_k$ at each node $k \in \mathcal{K}$, respectively, i.e.,

$$\mathbf{H}_k \triangleq [\mathbf{I}_{M_k} \ \mathbf{0}]\, \widetilde{\mathbf{W}}_k \tag{6.46}$$

$$\begin{aligned}\mathbf{G}_{k-k} &\triangleq \left[\mathbf{0} \ \mathbf{I}_{S(K-1)}\right] \widetilde{\mathbf{W}}_k \\ &= \left[\mathbf{G}_{k1}^T \dots \mathbf{G}_{k(k-1)}^T \ \mathbf{G}_{k(k+1)}^T \dots \mathbf{G}_{kK}^T\right]^T\end{aligned} \tag{6.47}$$

where submatrix $\mathbf{G}_{kn}$ is an $S \times N_k$ transformation matrix that node k applies to the broadcast signal $\mathbf{z}_n$ received from node $n \in \mathcal{K}\backslash k$, where $N_k = M_k$ if $k \in \mathcal{K}^{\mathrm{MWF}}$ and $N_k = S$ if $k \in \mathcal{K}^{\mathrm{LCMV}} \bigcup \mathcal{K}^{\mathrm{DOA}}$. After applying such $\mathbf{G}_{kn}$ transformations, and considering the definition of the compression matrices $\mathbf{F}_k$ in (6.45), it follows that an equivalent network-wide filter at each node $k \in \mathcal{K}$ is given as

$$\mathbf{W}_k = \begin{bmatrix} \mathbf{F}_1\mathbf{G}_{k1} \\ \vdots \\ \mathbf{H}_k \\ \vdots \\ \mathbf{F}_K\mathbf{G}_{kK} \end{bmatrix}, \forall k \in \mathcal{K} \tag{6.48}$$

such that $\widetilde{\mathbf{W}}_k^H \widetilde{\mathbf{y}}_k = \mathbf{W}_k^H \mathbf{y}$. Here, $\mathbf{H}_k$ is inserted instead of $\mathbf{F}_k\mathbf{G}_{kk}$ (note that $\mathbf{G}_{kk}$ is not defined in the partitioning (6.47)). It will be shown later in this section that the network-wide centralized solution of each node-specific estimation problem of each node $k \in \mathcal{K}$ is in the solution space defined by (6.48).

In the case of the MWF, when $\mathbf{R}_{yy}$ is replaced with (6.7), (6.10) can be rewritten as

$$\hat{\mathbf{W}}_k = \hat{\mathbf{Q}}^{-H}\hat{\mathbf{L}}^{-1}\hat{\mathbf{Q}}^{-1}\hat{\mathbb{Q}}(\hat{\mathbb{L}} - \mathbf{I}_S)\hat{\mathbb{Q}}^H\hat{\mathbf{E}}_k. \tag{6.49}$$

Note that since $\hat{\mathbf{Q}}^{-1}\hat{\mathbb{Q}} = [\mathbf{I}_S \ \mathbf{0}]^T$, the term $\hat{\mathbf{L}}^{-1}\hat{\mathbf{Q}}^{-1}\hat{\mathbb{Q}}$ in (6.49) can be alternatively written as

$$\hat{\mathbf{L}}^{-1}\hat{\mathbf{Q}}^{-1}\hat{\mathbb{Q}} = [\mathbf{I}_S \ \mathbf{0}]^T\hat{\mathbb{L}}^{-1}. \tag{6.50}$$

With this, and since $\hat{\mathbb{Q}}^H\hat{\mathbf{E}}_k = \hat{\mathbb{Q}}_k^H$ (see (6.20)), equation (6.49) can be simplified as

$$\hat{\mathbf{W}}_k = \hat{\mathbf{X}}\,[\mathbf{I}_S \ \mathbf{0}]^T\,\hat{\mathbb{L}}^{-1}(\hat{\mathbb{L}} - \mathbf{I}_S)\hat{\mathbb{Q}}_k^H \tag{6.51}$$

$$= \hat{\mathbb{X}}\hat{\mathbf{\Psi}}_k\ , \ \ \forall k \in \mathcal{K}^{\mathrm{MWF}} \tag{6.52}$$

where $\hat{\mathbf{\Psi}}_k$ is $S \times M_k$ transformation matrix at each node $k \in \mathcal{K}^{\mathrm{MWF}}$ defined as

$$\hat{\mathbf{\Psi}}_k = (\mathbf{I}_S - \hat{\mathbb{L}}^{-1})\hat{\mathbb{Q}}_k^H. \tag{6.53}$$

Similar to (6.49)-(6.53), at each node $k \in \mathcal{K}^{\mathrm{MWF}}$ the local MWF solution (6.28) can be expressed as

$$\widetilde{\mathbf{W}}_k = \widetilde{\mathbf{Q}}_k^{-H}[\mathbf{I}_S\ \mathbf{0}]^T\widetilde{\mathbb{L}}_k^{-1}(\widetilde{\mathbb{L}}_k - \mathbf{I}_S)\widetilde{\mathbb{Q}}_k^H\mathbf{E}_{M_k} \tag{6.54}$$

$$= \widetilde{\mathbb{X}}_k\widetilde{\mathbf{\Psi}}_k\ ,\ \ \forall k \in \mathcal{K}^{\mathrm{MWF}} \tag{6.55}$$

with the $S \times M_k$ transformation matrix $\widetilde{\mathbf{\Psi}}_k$ defined as (substitution $\widetilde{\mathbb{Q}}_k^H\mathbf{E}_{M_k} = \mathbb{Q}_k^H$ is verified by (6.42))

$$\widetilde{\mathbf{\Psi}}_k = (\mathbf{I}_S - \widetilde{\mathbb{L}}_k^{-1})\mathbb{Q}_k^H. \tag{6.56}$$

In the case of the LCMV, when $\mathbf{R}_{yy}$ is replaced with (6.7), and using (6.40), (6.50) and (6.41) the extended network-wide LCMV beamformer (6.38) can be expressed as

$$\hat{\mathbf{W}}_k = \hat{\mathbb{X}}\hat{\mathbf{\Delta}}_k^{-H}\hat{\boldsymbol{\mathcal{Y}}}_k = \hat{\mathbb{X}}\hat{\mathbf{\Theta}}_k\ ,\ \ \forall k \in \mathcal{K}^{\mathrm{LCMV}} \tag{6.57}$$

where, from (6.41), it follows that the $S \times S$ transformation matrix $\hat{\mathbf{\Theta}}_k$ is

$$\hat{\mathbf{\Theta}}_k = \begin{bmatrix} [\mathbf{I}_{\mathrm{D}_k}\ \mathbf{0}]\hat{\mathbb{Q}}_k^H[\boldsymbol{\alpha}\ \mathbf{0}]^T \\ [\mathbf{0}\ \mathbf{I}_{\mathrm{I}_k}]\hat{\mathbb{Q}}_k^H[\boldsymbol{\epsilon}\ \mathbf{0}]^T \end{bmatrix}. \tag{6.58}$$

Similar to (6.40)-(6.58), the local LCMV solution (6.36) can be expressed as

$$\widetilde{\mathbf{W}}_k = \widetilde{\mathbb{X}}_k\widetilde{\mathbf{\Theta}}_k\ ,\ \ \forall k \in \mathcal{K}^{\mathrm{LCMV}} \tag{6.59}$$

where the $S \times S$ transformation matrix $\widetilde{\mathbf{\Theta}}_k$ can be defined similarly to (6.58) as

$$\widetilde{\mathbf{\Theta}}_k = \begin{bmatrix} [\mathbf{I}_{\mathrm{D}_k}\ \mathbf{0}]\widetilde{\mathbb{Q}}_k^H[\boldsymbol{\alpha}\ \mathbf{0}]^T \\ [\mathbf{0}\ \mathbf{I}_{\mathrm{I}_k}]\widetilde{\mathbb{Q}}_k^H[\boldsymbol{\epsilon}\ \mathbf{0}]^T \end{bmatrix}. \tag{6.60}$$

Based on the descriptions of the local MWF estimator in (6.55) at nodes $k \in \mathcal{K}^{\mathrm{MWF}}$, the local LCMV estimator in (6.59) at nodes $k \in \mathcal{K}^{\mathrm{LCMV}}$ and the local DOA estimator in (6.43) at nodes $k \in \mathcal{K}^{\mathrm{DOA}}$, the compression matrices $\mathbf{F}_k$ of the distributed MDMT algorithm can be summarized as (see (6.45))

$$\mathbf{F}_k = \begin{cases} \mathbb{X}_k\widetilde{\mathbf{\Phi}}_k & \forall k \in \mathcal{K}^{\mathrm{MWF}} \quad (6.61a) \\ \mathbb{X}_k\widetilde{\mathbf{\Theta}}_k & \forall k \in \mathcal{K}^{\mathrm{LCMV}} \quad (6.61b) \\ \mathbb{X}_k & \forall k \in \mathcal{K}^{\mathrm{DOA}} \quad (6.61c) \end{cases}$$

where the $S \times S$ transformation matrix $\widetilde{\mathbf{\Phi}}_k$ at nodes $k \in \mathcal{K}^{\mathrm{MWF}}$ is defined as the matrix containing the first S columns of $\widetilde{\mathbf{\Psi}}_k$, i.e., $\widetilde{\mathbf{\Phi}}_k \triangleq \widetilde{\mathbf{\Psi}}_k \, [\mathbf{I}_S \, \mathbf{0}]^T$. From this

Table 6.2: Solution space of the distributed MDMT algorithm in (6.48)

at node k	$\forall n \in \mathcal{K}^{\mathrm{MWF}} \setminus k$	$\forall n \in \mathcal{K}^{\mathrm{LCMV}} \setminus k$	$\forall n \in \mathcal{K}^{\mathrm{DOA}} \setminus k$
$k \in \mathcal{K}^{\mathrm{MWF}}$ $\mathbf{H}_k = \hat{\mathbb{X}}_k \hat{\mathbf{\Psi}}_k$	$\mathbf{F}_n = \hat{\mathbb{X}}_n \hat{\mathbf{\Phi}}_n$ $\mathbf{G}_{kn} = (\hat{\mathbf{\Phi}}_n)^{-1} \hat{\mathbf{\Psi}}_k$	$\mathbf{F}_n = \hat{\mathbb{X}}_n \hat{\mathbf{\Theta}}_n$ $\mathbf{G}_{kn} = (\hat{\mathbf{\Theta}}_n)^{-1} \hat{\mathbf{\Psi}}_k$	$\mathbf{F}_n = \hat{\mathbb{X}}_n$ $\mathbf{G}_{kn} = \hat{\mathbf{\Psi}}_k$
$k \in \mathcal{K}^{\mathrm{LCMV}}$ $\mathbf{H}_k = \hat{\mathbb{X}}_k \hat{\mathbf{\Theta}}_k$	$\mathbf{F}_n = \hat{\mathbb{X}}_n \hat{\mathbf{\Phi}}_n$ $\mathbf{G}_{kn} = (\hat{\mathbf{\Phi}}_n)^{-1} \hat{\mathbf{\Theta}}_k$	$\mathbf{F}_n = \hat{\mathbb{X}}_n \hat{\mathbf{\Theta}}_n$ $\mathbf{G}_{kn} = (\hat{\mathbf{\Theta}}_n)^{-1} \hat{\mathbf{\Theta}}_k$	$\mathbf{F}_n = \hat{\mathbb{X}}_n$ $\mathbf{G}_{kn} = \hat{\mathbf{\Theta}}_k$
$k \in \mathcal{K}^{\mathrm{DOA}}$ $\mathbf{H}_k = \hat{\mathbb{X}}_k$	$\mathbf{F}_n = \hat{\mathbb{X}}_n \hat{\mathbf{\Phi}}_n$ $\mathbf{G}_{kn} = (\hat{\mathbf{\Phi}}_n)^{-1}$	$\mathbf{F}_n = \hat{\mathbb{X}}_n \hat{\mathbf{\Theta}}_n$ $\mathbf{G}_{kn} = (\hat{\mathbf{\Theta}}_n)^{-1}$	$\mathbf{F}_n = \hat{\mathbb{X}}_n$ $\mathbf{G}_{kn} = \mathbf{I}_S$

result, it can be shown that the solution space defined by the parameterization (6.48) contains the network-wide centralized solution for each node $k \in \mathcal{K}$, namely (see (6.52), (6.57))

$$\hat{\mathbf{W}}_k = \begin{cases} \hat{\mathbb{X}} \hat{\mathbf{\Psi}}_k & \forall k \in \mathcal{K}^{\mathrm{MWF}} \quad (6.62a) \\ \hat{\mathbb{X}} \hat{\mathbf{\Theta}}_k & \forall k \in \mathcal{K}^{\mathrm{LCMV}} \quad (6.62b) \\ \hat{\mathbb{X}} & \forall k \in \mathcal{K}^{\mathrm{DOA}} \quad (6.62c) \end{cases}$$

Now considering the task assigned to each node $k \in \mathcal{K}$, setting $\mathbf{F}_k$ and $\mathbf{G}_{kn}$ in (6.48) according to Table 6.2 verifies that the parameterization of the distributed MDMT algorithm allows to generate the network-wide centralized solutions (6.62a)-(6.62c) as a special case.

6.5.2 Proof of convergence and optimality

In this section, it is shown that the distributed MDMT algorithm converges, i.e.,

$$\lim_{i \to \infty} \mathbf{W}_k^i = \hat{\mathbf{W}}_k, k \in \mathcal{K}. \quad (6.63)$$

Remark 2: To make the convergence analysis tractable, it is assumed that all correlation matrices can be perfectly estimated using infinite observation windows (similar to [67, 68, 71, 123]). When finite observation windows are used to estimate correlations, this analysis should be considered as an asymptotic case, where larger observation windows increase the approximation accuracy.

Theorem I: *If $\mathbf{R}_{yy}$ is full rank, then the estimates obtained from the distributed MDMT algorithm converge for any initialization of the compression matrices*

$\mathbf{F}_k^0, \forall k \in \mathcal{K}$ *to the corresponding estimates obtained with the centralized estimation algorithms, i.e., when* $i \to \infty$, $\forall k \in \mathcal{K}^{MWF}$, $\widetilde{\mathbf{s}}_k^i = \hat{\mathbf{s}}_k$, *and* $\forall k \in \mathcal{K}^{LCMV}$, $\widetilde{d}_k^i = \hat{d}_k$, *and* $\forall k \in \mathcal{K}^{DOA}$, $\widetilde{\theta}_{k_s}^i = \hat{\theta}_{k_s}$ *(or equivalently based on the parameterization (6.48),* $\lim_{i\to\infty} \mathbf{W}_k^i = \hat{\mathbf{W}}_k, k \in \mathcal{K}$*).*

Proof: Similar to the strategies used in [123], it can be shown that Theorem I eventually follows from the convergence properties of a particular distributed algorithm that computes the network-wide GEVD (this distributed GEVD algorithm is explained in Appendix A). However, because of the key difference between [123] and the proposed distributed MDMT algorithm, the proof of Theorem I becomes non-trivial and needs to be formalized. It is reiterated that this key difference originates from the fact that in [123] the compression matrices at different nodes $k \in \mathcal{K}$ are drawn from the estimators $\widetilde{\mathbf{W}}_k$ of the same task (MWF), whereas in the distributed MDMT algorithm they are drawn from the estimators $\widetilde{\mathbf{W}}_k$ of different tasks, depending on whether node k is an MWF, LCMV or DOA node. As a result, the convergence of different nodes $k \in \mathcal{K}^{\mathrm{MWF}}$, $k \in \mathcal{K}^{\mathrm{LCMV}}$ and $k \in \mathcal{K}^{\mathrm{DOA}}$ is required to be investigated individually. A formal proof of Theorem I will hence be provided in Appendix B.

Remark 3: So far we have assumed that all nodes $k \in \mathcal{K}$ know the exact number of sources S, and that all LCMV nodes $k \in \mathcal{K}^{\mathrm{LCMV}}$ know their node-specific values of D_k and $I_k = S - D_k$. In practice, however, S usually has to be estimated on-the-fly. With $\bar{S}$ denoting the resulting estimate, the following two cases can then happen when $\bar{S} \neq S$:

- $\bar{S} > S$: The centralized solutions (asymptotically) converge to their corresponding optimal solutions. In particular, it has been shown that the $M \times \bar{S}$ matrix $\mathbb{X}$ converges to the $M \times \bar{S}$ matrix $\hat{\mathbb{X}}$ [1] and the $M \times \bar{S}$ matrix $\mathbb{Q}$ converges to the $M \times \bar{S}$ matrix $\hat{\mathbb{Q}}$ [120]. Consequently, when these rank-$\bar{S}$ estimates are incorporated, the resulting centralized MWF solution (6.10) (asymptotically) converges to the optimal LMMSE solution of (6.3), and the LCMV solution (6.14) (asymptotically) converges to its optimal LCMV solution. Note that in this case, a drawback is that the compression matrices $\mathbf{F}_k$ will become $M_k \times \bar{S}$, and hence each node k will require a *larger* communication bandwidth comparing to the case when $\bar{S} = S$, which would provide the same performance.
- $\bar{S} < S$: Based on similar arguments as [123], the rank-$\bar{S}$ approximation effectively redefines (imposes) a common latent signal subspace of dimension $\bar{S}$ for the underlying data model, while the actual data (6.1)-(6.2) actually has a common latent signal subspace of dimension S (see (6.5)). Nevertheless, since the GEVD provides the subspace $\mathbb{X}$ with the maximal SNR, the resulting estimators may still provide a relevant (but not optimal) solution

(this was analyzed in [123] for the case where all nodes are MWF node). Note that in this case, each node k will require a *lower* communication bandwidth comparing to the case when $\bar{S} = S$ at the cost of a reduced estimation performance compared to the centralized case. However, the convergence proof of Theorem I remains valid in this case.

For the case of LCMV, the estimate $\bar{S}$ also has an influence on the values D_k and $I_k = \bar{S} - D_k$ in nodes $k \in \mathcal{K}^{\text{LCMV}}$. Since these are node-specific quantities, each LCMV node can make an independent choice and errors in the determination of these local values of D_k and I_k will not affect the convergence of the algorithm (although the local output computed by node k will be wrong due to an incorrect definition of its node-specific D_k and I_k). If $D_k + I_k = S$, the algorithm also remains optimal for all (other) nodes. If $D_k + I_k = \bar{S}$ with $\bar{S} \neq S$, then one of the situations explained above will apply.

6.6 Simulation results

To evaluate the performance of the proposed distributed MDMT algorithm and to further investigate its convergence in a realistic environment, a multi-source acoustic scenario is simulated using the image method [152]. It should be mentioned that we do not aim at implementing a fully practical speech enhancement scenario and the goal here is only to show the convergence and efficiency of the proposed MDMT algorithm in a realistic enclosure. To achieve this goal, a WASN with 4 nodes ($K = 4$) is considered inside a square room $6\,\text{m} \times 6\,\text{m} \times 6\,\text{m}$ with reflection coefficients of 0.2 at all surfaces, where the acoustic Room Impluse Responses (RIRs) are simulated using the RIR-generator in [164]. Each acoustic node of the WASN is equipped with a uniform linear array with 3 omni-directional microphones ($M_k = 3, \forall k \in \mathcal{K}$), with an inter-microphone distance of 10 cm. The acoustic scenario is depicted in Figure 6.3. Two speech sources ($S = 2$) are located at the positions $[x = 2\,\text{m}, y = 3\,\text{m}]$ and $[x = 3\,\text{m}, y = 2.5\,\text{m}]$ (source 2 at the broadside direction of node 2) and produce different speech signals (English sentences with silence period in between). In addition, five noise sources are located in the room, each producing a multi-talker babble noise (mutually uncorrelated). In Figure 6.3, the input *signal to noise ratio* (iSNR) in decibel (dB) at the first microphone of each node is given.

In this MDMT WASN, node 1 computes an MWF and estimates the speech signals coming from both of the speech sources (depicted as S1 and S2 in Figure 6.3), node 2 computes an LCMV beamformer and estimates source 1 while suppressing source 2, node 3 estimates an LCMV beamformer and estimates source 2 while suppressing source 1, and node 4 aims to compute the DOAs

for both of the speech sources (true DOAs at node 4 are 72° and 116°). For both of the LCMV nodes, $\alpha = 0.9$ and $\epsilon = 0.1$ are chosen. Since speech signals are broadband signals, the algorithms operate in the short-time Fourier transform domain, where the estimation problems are solved for each frequency bin separately. We use a sampling frequency of 16 kHz, a Hann-windowed DFT with window size 256 and with 50% overlap. We assume a perfect multi-speaker VAD to exclude the effect of errors. In addition to the noise captured from the localized noise source, uncorrelated white Gaussian noise is added to each microphone signal to model the microphone's self-noise and other possible isotropic noise contributions. It is noted that these simulations are carried out in batch mode, which means that the signal statistics are estimated over the full signal length in each iteration. For the DOA estimation at node 4, we use a wideband ESPRIT algorithm [115] to estimate the DOAs from its $\hat{\mathbb{Q}}_4$ and $\mathbb{Q}_4^i$.

As a performance measure at the MWF node 1, we use the output SNR, which is particularly defined in iteration i as

$$\text{oSNR}^i \triangleq 10\log_{10}\frac{E\{\|\mathbf{W}_1^{i\,H}\mathbf{s}(1)\|^2\}}{E\{\|\mathbf{W}_1^{i\,H}\mathbf{n}(1)\|^2\}} \tag{6.64}$$

where (1) refers to the fact that the output signals at the first microphone is considered. In addition, at this MWF node, the MSE between the estimated signal $\widetilde{\mathbf{s}}_1^i$ and that of the centralized MWF $\hat{\mathbf{s}}_1$ is applied. For the LCMV nodes 2 and 3, we use output *signal-to-interference-and-noise ratio* as the measure, which is defined as the ratio between the output power of the desired signal and the output power of both the noise and interfering speech source. In these LCMV nodes, the MSE between the $\tilde{d}_k^i$ and its corresponding centralized estimate $\tilde{d}_k$ is applied to further investigate the convergence. For the DOA node 4, the convergence and the absolute error (in degrees) between the MDMT estimates and those of the centralized case are provided. Figures 6.4 and 6.5 illustrate both the convergence and the performance of the proposed distributed MDMT algorithm at all nodes. Cases where nodes estimates their node-specific tasks on their own, called as 'isolated', are added in Figure 6.4 to also show the effectiveness of the algorithm. Note that in theory the MSE values given in Figure 6.5 will go to zero. However, since in this audio scenario convolutive mixtures are solved in the short-time Fourier transform domain, the data model (6.1) is only appropriately satisfied and hence the convergence of MSEs to zero cannot be achieved. Finally, to further evaluate the output performance of the MDMT algorithm at MWF and LCMV nodes, we define MSE2 as the MSE between the estimated and true values of the desired signals for the isolated, network-wide and the distributed MDMT output solutions. The MSE2 results are shown in Figure 6.6. All these results clearly show that all the estimates obtained from the proposed distributed MDMT algorithm converges to the

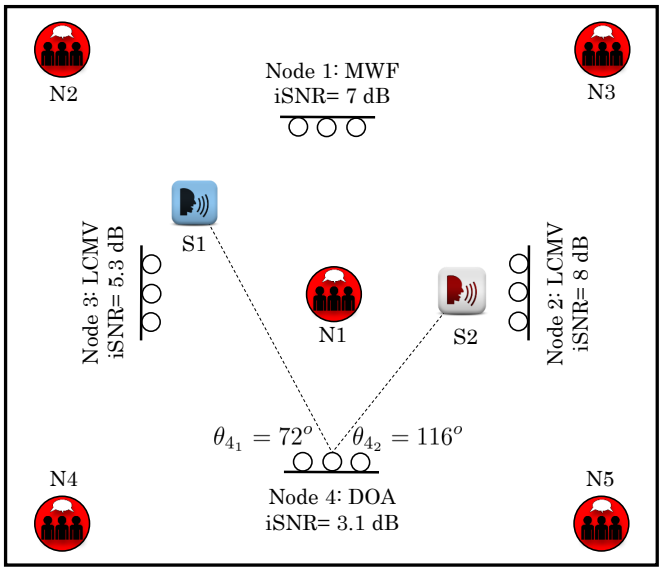
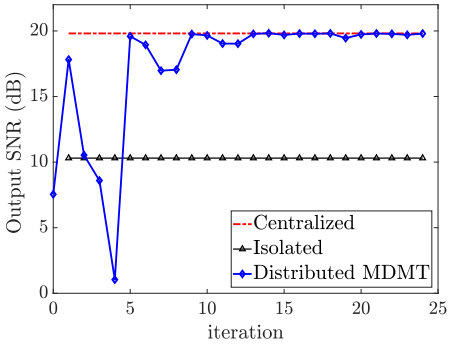


Figure 6.3: Multi-source acoustic scenario for the MDMT WASN. Speech sources are S1 and S2. Five multi-talker babble noise sources are N1 to N5.

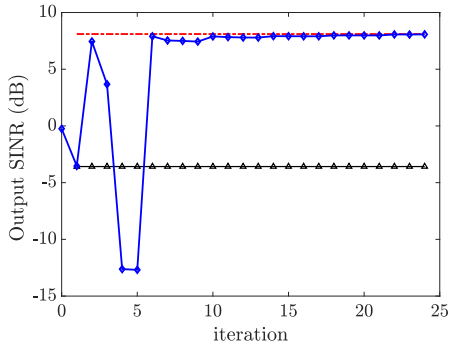
corresponding centralized estimates obtained in the centralized case, which in fact deliver a significantly better performance compared to the isolated case.

6.7 Conclusions

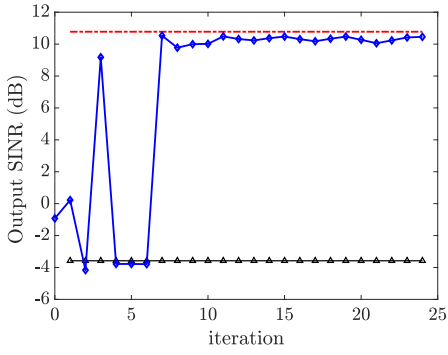
We have studied a particular multi-task problem in an MDMT WSN formed by three different groups of nodes. In the first group, each node aims at applying an MWF to denoise its sensor signals. In the second group, each node aims to extract specific desired source signals, while suppressing others by implementing LCMV beamformer. In the third group, each node is interested in estimating the node-specific DOAs of a set of desired sources. For this setting, we have derived a distributed MDMT algorithm under which all the nodes can cooperate to solve their different signal processing tasks. Theoretical results show that the algorithm allows each node to attain the network-wide centralized solution of its estimation problem with reduced communication resources, even without being aware of the SP tasks solved by the other nodes. To do so, the proposed algorithm relies on a low-rank approximation of the desired signals correlation matrix based on the GEVD. Finally, simulations have validated the theoretical results and have shown the efficiency of the proposed algorithm.



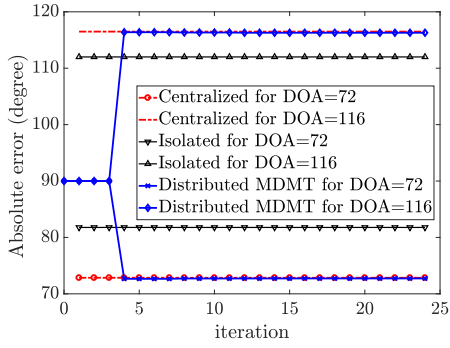
(a) Node 1 **MWF**: S1 target S2 target.



(b) Node 2 **LCMV**: S1 target S2 interferer.

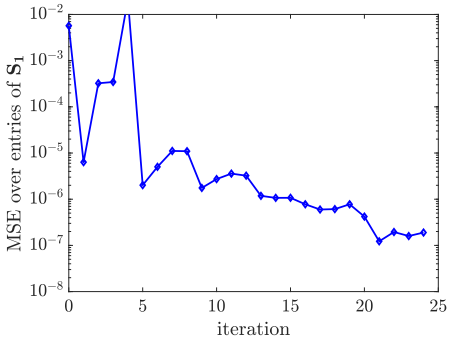


(c) Node 3 **LCMV**: S1 interferer S2 target.

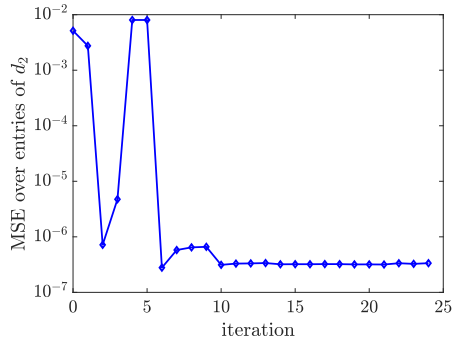


(d) Node 4 **DOA**: S1 target S2 target.

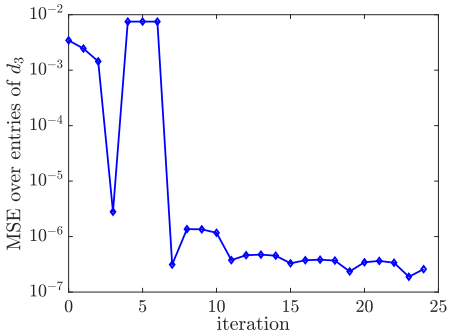
Figure 6.4: The oSNR for the local output signal at the node 1 (MWF), the oSINRs for the local output signals at the nodes 2 and 3 (LCMVs), and absolute error for the local DOA estimates at the node 4 (DOA) within the WASN described in Figure 6.3. The plots show the convergence of all the nodes to their centralized (optimal) cases in terms of the output performance, versus the iteration index of the distributed MDMT algorithm.



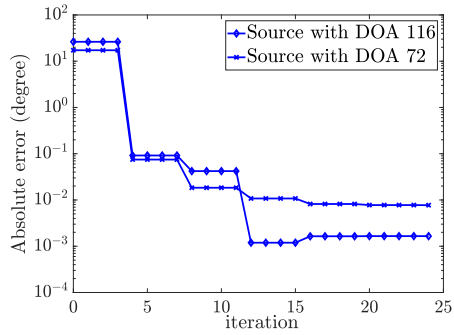
(a) Node 1 **MWF**: S1 target S2 target.



(b) Node 2 **LCMV**: S1 target S2 interferer.

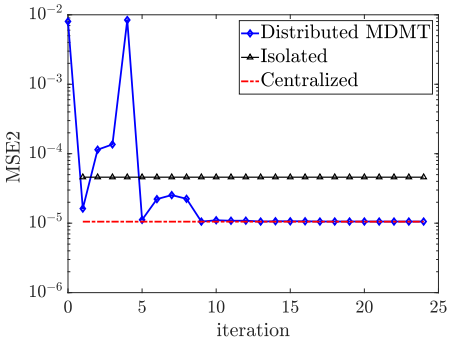


(c) Node 3 **LCMV**: S1 interferer S2 target.

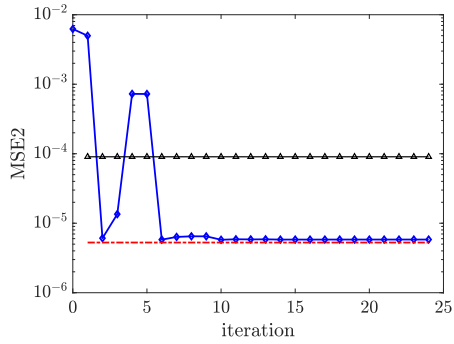


(d) Node 4 **DOA**: S1 target S2 target.

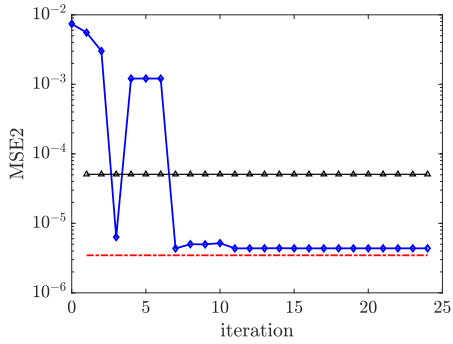
Figure 6.5: The MSE at the node 1 (MWF) and the nodes 2 and 3 (LCMVs) between the local output signals and those signals obtained from their centralized (optimal) cases within the WASN described in Figure 6.3. Likewise, at the node 4 (DOA), individual absolute errors between each of the local DOA estimates and those obtained from the centralized (optimal) cases are shown. The plots show the convergence of all the local estimates to their corresponding centralized (optimal) estimates, versus the iteration index of the distributed MDMT algorithm.



(a) Node 1 **MWF**: S1 target S2 target.



(b) Node 2 **LCMV**: S1 target S2 interferer.



(c) Node 3 **LCMV**: S1 interferer S2 target.

Figure 6.6: The MSE2 shows the MSE at the node 1 (MWF) and the nodes 2 and 3 (LCMVs) between the estimated signals (including both centralized and distributed MDMT estimates) and *true* values of the desired signals, versus the iteration index of the distributed MDMT algorithm within the WASN described in Figure 6.3.

Part III

Real-time experimental validation

Chapter 7

Distributed GEVD-based Speech Enhancement with Collaborating Microphone Arrays: Real-Time Implementation and Instrumental Evaluation

7.1 Introduction

In Chapter 3, the generalized eigenvalue decomposition (GEVD)-based distributed adaptive node-specific signal estimation (DANSE) algorithm was presented and its convergence properties were analyzed in detail. As discussed, the GEVD-based DANSE algorithm essentially provides a distributed realization of the centralized GEVD-based MWFs of individual nodes of a sensor network and allows the nodes to cooperate by exchanging pre-filtered and compressed signals. As a result, the GEVD-based DANSE algorithm reduces per-node computational power and communication bandwidth. It has been shown in Chapter 3 that the GEVD-based DANSE algorithm eventually converges to the centralized GEVD-based MWFs as if each node would have access to all

the sensor signals of the entire network. Unlike the original version of DANSE in [67, 68] in which the required speech correlation matrices are estimated using a straightforward subtraction-based method, this algorithm incorporates a GEVD-based low-rank approximation of the signal correlation matrix in each local MWF. This GEVD-based MWF has been shown to deliver a superior performance in the presence of second-order statistics errors (e.g., due to low-SNR conditions, highly non-stationary noise or erroneous voice activity detections (VADs) [103]).

Monte-Carlo simulation results were provided in Chapter 3 to demonstrate the convergence and performance behavior of the GEVD-based DANSE algorithm in different conditions. However, since mostly batch-mode simulations were performed for stationary and narrowband signals, the true potential of the GEVD-based DANSE algorithm as an adaptive distributed noise reduction algorithm has not yet been demonstrated. Therefore, this chapter aims to further asses the merits of this algorithm in a fully adaptive and realistic scenario. To achieve this, a realistic speech enhancement scenario is created using an acoustic sensor network with three collaborating microphone arrays. The GEVD-based DANSE algorithm is then run on a real-time test platform to let different microphone arrays collaborate to enhance their captured noisy speech signals. Therefore, this evaluation needs to additionally deal with practical real-time implementation aspects. It is noted that in this evaluation setup the communication links are wired, and all the microphone arrays are connected to the a central signal processing platform, in which the distributed algorithms are operated, emulating a real-time acoustic sensor network (see Section 7.3).

Speech enhancement algorithms in essence aim at increasing speech *intelligibility* and *quality* for better speech perception in presence of ambient noise, while improvement in the former is typically found to be more important [165]. Therefore, when it comes to evaluation of such algorithms, it is important to consider both of these perceptual aspects separately, as speech enhancement techniques do not necessarily improve both at the same time, e.g., an attempt to increase the speech intelligibility (noise reduction performance) can decrease the speech quality (speech distortion level) [113, 166, 167].

This chapter evaluates the GEVD-based DANSE algorithm using instrumental measures for both speech intelligibility and speech quality. For **speech intelligibility**, the SI-weighted signal-to-noise ratio (SNR) measure [168, 169] and the short-time objective intelligibility (STOI) measure [170] will be used. For **speech quality**, the SI-weighted spectral distortion (SD) measure [105] and the perceptual evaluation of speech quality (PESQ) measure [171] will be used. These instrumental measures are widely used to evaluate the capabilities of speech enhancement algorithms in terms of speech intelligibility enhancement

and quality improvement [165, 172, 173].

This chapter is organized as follows. In Section 7.2, the speech enhancement algorithms under evaluation will be outlined. In Section 7.3, the hardware platform employed for real-time signal processing will be introduced. In Section 7.4, the acoustic test scenario in which the speech enhancement algorithms are assessed will be described. In Section 7.5, the instrumental measures utilized in this evaluation setup will be briefly explained. In Section 7.6, the graphical user interface (GUI) designed for this evaluation setup will be introduced. Finally, in Section 7.7, evaluation results will be provided and conclusions will be drawn.

7.2 Algorithms under evaluation

In order to effectively evaluate the performance of the GEVD-based DANSE algorithm, the following (upper-bound or lower-bound) benchmark algorithms are also provided:

- **Isolated GEVD-based MWF**: the *non-cooperative* case where each node has only access to its own microphone signals (lower-bound benchmark)
- **Centralized GEVD-based MWF**: the *full-bandwidth cooperative* case where each node has access to all microphone signals in the acoustic sensor network (upper-bound benchmark)
- **GEVD-based DANSE**: the *cooperative distributed* case where nodes only exchange optimally-compressed S-channel signals (S denotes the number of target speech sources)
- **Basic cooperation with single microphone exchange**: this requires the same computational power and communication bandwidth as DANSE, but without using the optimal compression.

All above-mentioned speech enhancement algorithms have been implemented on a central real-time signal processing platform and evaluated on the same acoustic test scenario. To apply the instrumental measures, all multi-channel input and output signals as well as algorithms' parameters (e.g., filter coefficients, VADs) need to be logged while the systems is running in real-time. The fact that all of these algorithms run in real-time, makes this an ideal platform for future subjective listening tests. To facilitate that, a GUI is designed to let the experimenters interact with the system, change the parameter settings, and

switch between the above-mentioned speech enhancement strategies in real-time (see Section 7.6 for more details).

Acoustic test scenarios which are particularly challenging for an MWF-based approach have been chosen in this evaluation, including low-SNR conditions and non-stationary noise fields. Under such conditions, we have compared the performance of the above-mentioned speech enhancement strategies with different instrumental measures. As a result, the benefits obtainable by the GEVD-based DANSE algorithm in a realistic speech enhancement scenario have been investigated.

A video footage providing an example of the output signals of the above-mentioned speech enhancement strategies while the setup is running in real-time has been presented in [174]. This indeed reveals how the GEVD-based DANSE algorithm enhances the speech signal in the presence of highly non-stationary construction and multi-talker noise and obtains the same performance as the centralized MWF.

7.3 Real-time signal processing platform

For the real-time signal processing platform, a so-called high performance, multi-core *xPC target* machine (Speedgoat, Switzerland) is employed. This x86-based target machine allows for hardware-in-the-loop (HIL) prototyping and testing of signal processing algorithms in real-time and in realistic environments, while avoiding the implementation using any low-level programming language. For this, the signal processing algorithms under test are first modeled using Simulink® on the so-called *host PC* (see Figure 7.1). This host PC is additionally used for live monitoring and tuning model parameters and configuring the entire system. Afterwards, a software environment containing SIMULINK Coder®, Matlab Coder®, C/C++ compiler and Simulink Real-time® (formerly known as xPC Target® toolbox), translates the SIMULINK model into an executable C-code and builds a target application, which is then transferred to the xPC target machine using an Ethernet connection. The embedded kernel of the xPC target machine then executes the codes in real-time independently from the host PC, i.e., the Ethernet connection can be terminated once the xPC starts to execute the codes. However, to exploit the favorable feature of the system on monitoring and live parameter tuning during the experiment (at runtime), the Ethernet connection needs to be pertained to let the experimenter interact with the xPC target via a GUI specifically designed for algorithms under test. This feature for interactive control of user-specific parameters in real-time is indeed a great advantage of this real-time signal processing platform. Besides,

the xPC target machine allows to log the data at intermediate processing stages in real-time, making it possible to analyze the data offline at any later time.

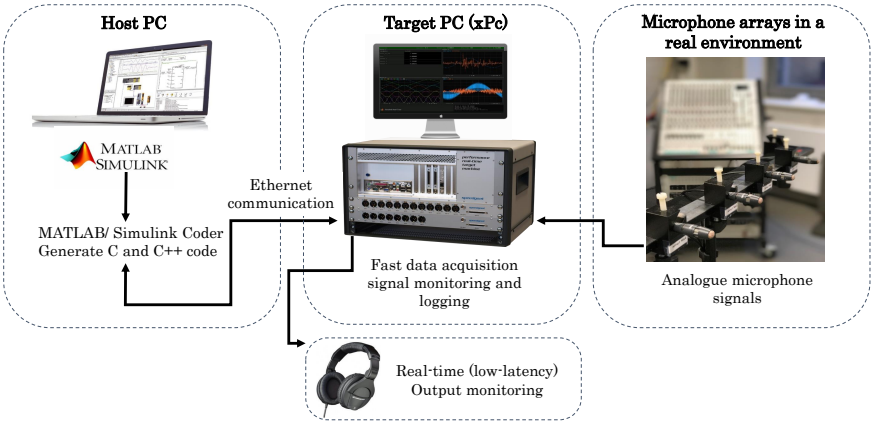


Figure 7.1: Block diagram of the complete real-time signal processing system.

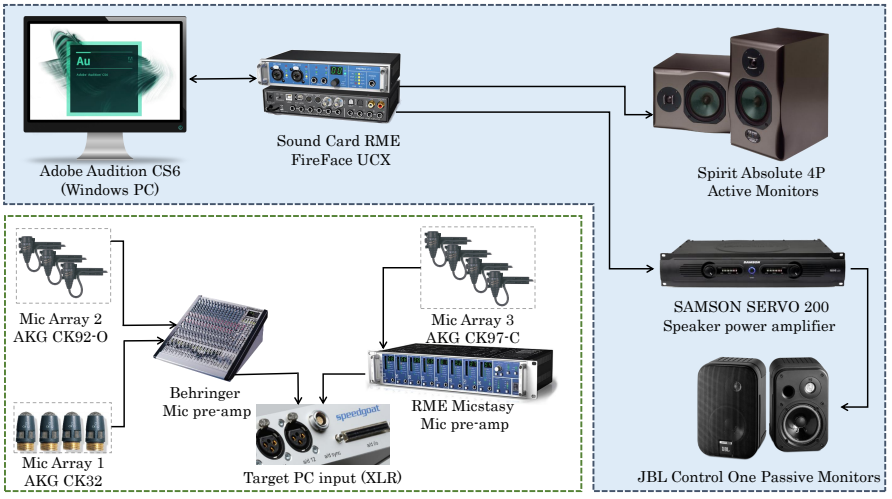


Figure 7.2: Block diagram of the complete audio setup. The shaded and non-shaded parts show equipment used for the audio playback and recording tasks, respectively.

Another interesting feature of this real-time signal processing platform is its ability to split the entire computational load of the signal processing problem into smaller parts and process them in parallel on different CPU cores. This can significantly reduce the risk of CPU overruns during the execution of relatively complex algorithms. In order for this to happen, the model needs to be explicitly partitioned and manually mapped to different CPU cores, preferably in such a way that the computational load is evenly shared between the available cores. To facilitate the latter, the task and function execution profiling tool offered by Simulink Real-time® can be leveraged to properly tune the performance of the real-time algorithms under test.

To realize the real-time HIL evaluation setup, low-latency input/output (I/O) boards are interfaced between the C-coded algorithms and the microphone signals captured in the acoustic environment. A block diagram of the complete real-time signal processing system is illustrated in Figure 7.1.

The xPC target machine used in this evaluation setup has the following hardware specifications. It includes a total of 12 high-resolution 24-bit sigma-delta differential analog input channels[1] and 8 16-bit differential analog output channels[2] which are all accessed through balanced XLR audio connectors. Note that the microphones/ loudspeakers signals are required to be pre-amplified before/after they are routed to these input/ output XLR panels. Other specifications are as follows: processor intel Core i7 3.5 GHz (four cores), 4GB DDR3 RAM, and 250 GB solid state drive for data logging.

7.4 Test setup description

7.4.1 Audio equipment

The audio devices are used to perform two main tasks: audio *playback* and audio *recording*. The equipment used in each task is depicted in Figure 7.2.

On the **playback** side, all audio stimuli are controlled by Adobe Audition® CS6 and played back at a sampling frequency of 16kHz. An audio stream input/output (ASIO) driver protocol is used to interface with the RME FireFace UCX (RME Audio, Germany) sound card. The analogue output channels are then fed into the different loudspeakers available in the environment. Two

[1] General Standards 24DSI12, simultaneous sampling, dedicated ADC per channel, and synchronized channels

[2] General Standards 16AO16, simultaneous sampling, dedicated DAC per channel, and synchronized channels

pairs of monitor loudspeakers are used, of which the first pair consists of 2 JBL Control One passive monitor loudspeakers (LS1 and LS2 in Figure 7.3) powered by a Samsung Servo 200 power amplifier, and the second pair consists of Soundcraft Spirit Absolute 4P active monitor loudspeakers (LS3 and LS4 in Figure 7.3). These channels are controlled independently through 4 different (mono) analogue output ports. In addition, a pair of stereo PC loudspeakers (Harman Kardon HK 206) is employed (LS5 in Figure 7.3). The entire audio playback setup is used to produce degraded speech signals by various noise types and with different SNR levels.

On the **recording** side, three different microphone arrays are used to pick up the degraded speech signals produced by the loudspeakers (see Figure 7.2). All microphones are condenser microphones, hence each requires a power (often called a *phantom power*). These powers are provided by a Behringer (for array 1 and 2) and an RME Micstasy (for array 3) pre-amplifiers. In particular, the first array consists of four AKG CK32 omnidirectional microphones (MA1 in Figure 7.3), the second array consists of three AKG-CK92 omnidirectional microphones (MA2 in Figure 7.3), and the third one consists of four AKG CK97-C cardioid microphones (MA3 in Figure 7.3). The microphone arrays have a uniform linear configuration with an inter-microphone spacing of 10 cm, 15 cm and 5 cm, respectively, and use separate stands. These microphones are employed without any prior gain or phase calibration and are placed in random locations in a standard office-style room (the acoustic test scenario will be described in Section 7.4.2).

All the microphone signals are finally fed to XLR input ports of the xPC target machine. Hence, these microphone signals occupy 11 input ports (out of 12). At port 12, a copy of the clean speech signal is provided to enable the xPC target machine to internally construct a perfect VAD. For cases when the experimenter needs to listen to the outputs in real-time (e.g., in a demonstration session), the processed output signals are routed to output ports and are presented through monitor headphones Beyerdynamic DT 770 PRO.

7.4.2 Acoustic test scenario

To reflect a daily-life acoustic situation, an acoustically-untreated room with approximate dimensions of 5.3 m × 3.5 m × 2.5 m is chosen for this evaluation setup. All 5 loudspeakers and 3 microphone arrays described in Section 7.4.1 are employed to generate the test scenario. A schematic illustration of this scenario including the positions of audio equipment is depicted in Figure 7.3. Furthermore, photographs taken inside the test room are shown in Figure 7.4. Two test scenarios have been considered. The first scenario considers a case

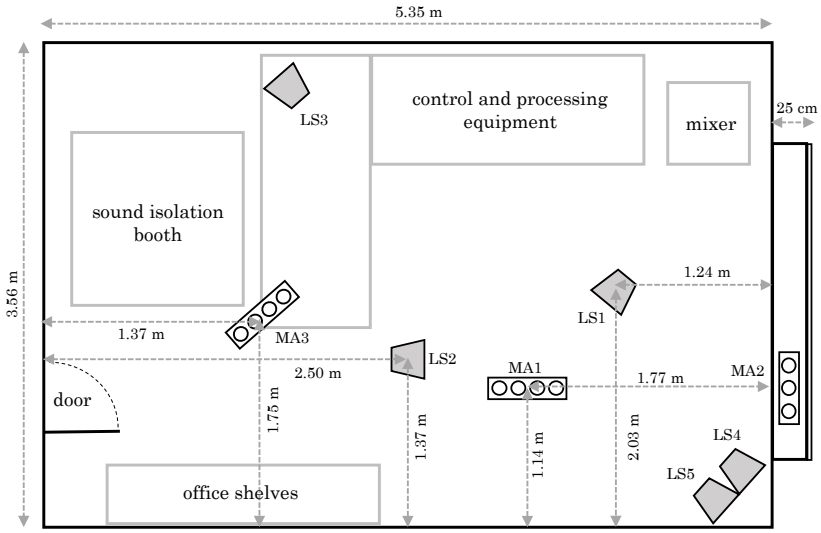


Figure 7.3: Schematic illustration of the test setup including the positions of loudspeakers (LS1-LS5) and microphone arrays (MA1-MA3).

where only one target speech source is present. In this case, LS1 has been used as target speech source, and LS2-LS5 are used as noise sources (see Section 7.7.2). The second scenario considers a multi-speaker case where two target speech sources are present. In this case, LS1 and LS2 have been used as target speech sources, and LS3-LS5 are used as noise sources (see Section 7.7.3).

Regarding the elevation level of the audio equipment, all the microphone arrays MA1-MA3 are placed at 1.10 m above the ground, and the loudspeakers L1-L5 are placed at the height of $[1.10m\ 1.10m\ 1m\ 0.7m\ 0.7m]$, respectively. Note that LS3-LS5 loudspeakers are intentionally not pointed toward the microphone arrays, but are pointed toward the walls. This is to reduce the direct-to-reverberant energy ratio (DRR) as an attempt toward introducing a better Pseudo-diffuse noise field.

Dry clean speech signals are taken from the hearing in noise test (HINT) database [175]. The speech signals are corrupted with stationary and non-stationary noise components. To create a stationary yet spatial noise field, a noise signal whose long-term average spectrum is similar to that of speech signals, known as speech-shaped noise (SSN), is drawn from NOISEX-92 database [176]. As for generting non-stationary noise, a multi-talker babble noise is drawn from Auditec database [177], where different (uncorrelated) noise segments of the same original noise file have been chosen by delaying the starting point.



Figure 7.4: photos of the evaluation setup

7.5 Instrumental measures

In order to assess predicted speech intelligibility and predicted speech quality, the algorithms under evaluation will be compared by means of four different instrumental measures. These measures are briefly explained in the following.

7.5.1 SI-weighted SNR (SNR_{SI})

Speech enhancement algorithms can be evaluated in terms of noise reduction performance by means of the SI-weighted SNR (SNR_{SI}) measure [168], in which the perceptual speech intelligibility is taken into account when calcualting the SNR values. For this, the speech and noise components of the microphone signals are first required to be processed separately by the filters computed based on the speech enhancement algorithm under evaluation. The resulting output speech and noise signals are then filtered by one-third octave bandpass filters (containing 18 center frequencies f_i ranging from 160 Hz to 8 kHz), and the long-term SNR is obtained per band. These SNR values are then weighted by band-importance function provided by speech intelligibility index (SII) standard [169], then averaged to obtain the overall SNR_{SI}. The SNR_{SI} improvement (in dB) is then defined as [105, 168]:

$$\Delta\mathrm{SNR}_{SI} = \sum_i I_i \, (\mathrm{SNR}^{\mathrm{out}}_{SI,i} - \mathrm{SNR}^{\mathrm{in}}_{SI,i}) \tag{7.1}$$

where I_i is the importance weight corresponding to i-th one-third octave band with center frequency f_i for intelligibility[3] (values of f_i's and I_i's are defined in [169]), and where $\mathrm{SNR}^{\mathrm{out}}_{SI,i}$ and $\mathrm{SNR}^{\mathrm{in}}_{SI,i}$ denote the output SNR and the input SNR[4] (in dB) calculated for the i-th one-third octave band, respectively.

7.5.2 SI-weighted spectral distortion (SD_{SI})

This measure is essentially used to evaluate possible speech distortions that the filters of a speech enhancement algorithm impose on the speech signals, taking into account the band-importance for the perceptual speech intelligibility. Similar to the SNR_{SI} measure introduced in the previous section, the SD_{SI}

[3] For instance, this weighting takes into account that that frequencies around 1 kHz are most important while frequencies above 5 kHz are less important for speech intelligibility, when percieved by normal hearing subjects.

[4] In this evaluation setup, the $\mathrm{SNR}^{\mathrm{in}}_{SI,i}$ has always been calculated at the first microphone of each node.

measure is defined as

$$\mathrm{SD}_{SI} = \sum_i I_i \ \mathrm{SD}_i \tag{7.2}$$

where SD_i is the average spectral distortion in the i-th one-third octave band with center frequency f_i, defined as [105]

$$\mathrm{SD}_i = \frac{1}{(2^{1/6} - 2^{-1/6}) f_i} \int_{2^{-1/6} f_i}^{2^{1/6} f_i} |10 \log_{10} G(f)| df$$

where $G(f)$ is the speech power transfer function calculated as

$$G(f) = \frac{P_s^{\mathrm{in}}(f)}{P_s^{\mathrm{out}}(f)}$$

where $P_s^{\mathrm{in}}(f)$ and $P_s^{\mathrm{out}}(f)$ denote the power, for the frequency f, of the speech component in the input reference microphone signal and the output processed signal, respectively. When using SD_{SI}, a distortion value of 0 dB corresponds to an undistorted output speech signal, while larger SD_{SI} values correspond to more speech distortion introduced by a speech enhancement algorithm.

7.5.3 Short-time objective intelligibility (STOI)

The short-time objective intelligibility (STOI) measure [170] calculates the correlation between a clean reference speech signal and a (noisy) processed speech signal delivered by a speech enhancement algorithm. To obtain this, both clean reference and noisy speech signals are divided into short time-frequency segments, filtered by one-third octave bandpass filters (containing 15 center frequencies f_i ranging from 150 Hz to 4.3 kHz), and then per-segment correlations are computed. Afterwards, these correlations are averaged across all time frames and all frequency bands to obtain one global STOI score, which ranges between 0-100 percent. The STOI measure has been shown to have a high correlation with subjective speech intelligibility test for normal hearing subjects and hence can be used to predict output speech intelligibility of speech enhancement and speech separation algortihms [178, 179].

7.5.4 Perceptual evaluation of speech quality (PESQ)

The perceptual evaluation of speech quality (PESQ) measure [171] has been originally developed to assess the speech quality of telephone networks. PESQ compares a clean reference speech signal with a processed speech signal by means of a perceptual model and produces an output which is mapped to a mean

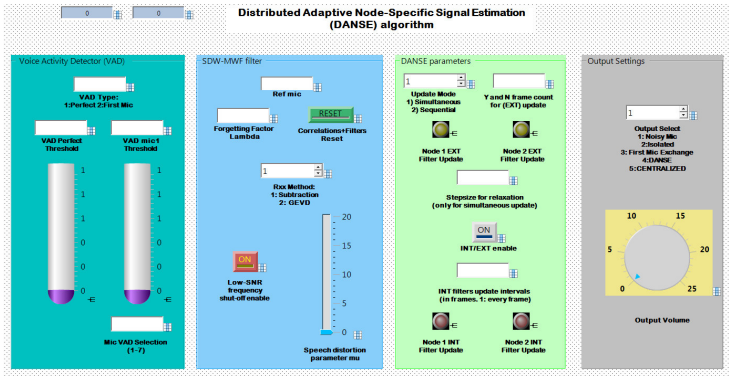


Figure 7.5: Screenshot of the designed GUI.

opinion score (MOS), covering a range from -0.5 (highly degraded processed speech signals) to 4.5 (no difference between clean reference and processed speech signals). Therefore, a higher PESQ score means a better speech quality. The PESQ measure has found to be well correlated with subjective quality measure for normal hearing subjects [180]. Note that when the PESQ measure is used to asses the quality of a speech signal processed by speech enhancement algorithms, residual noise is treated as a quality impairment.

7.6 Graphical user interface (GUI)

A GUI, running on the Host PC, is designed via Simulink Real-time Explorer® and provided such that the experimenter can easily select the desired speech enhancement algorithm and change the corresponding parameters in real-time. In particular this GUI has 4 main control panels, as follows:

- **VAD settings**: through which the experimenter can choose the VAD type[5]
- **MWF settings**: through which the experimenter can select the reference microphone and tune the speech distortion parameter [103]. Moreover, either the subtraction-based or GEVD-based method can be selected for the estimation of the speech correlation matrix

[5]Here a perfect VAD is used for all the experiments. However, the GUI allows to further choose a realistic VAD which works based on the energy of the frames.

- **DANSE settings**: through which the experimenter can choose the update mode (sequential [67] vs. simultaneous [68]), tune the node update frequency (as a function of the number of collected speech-and-noise and noise-only frames) and step size for relaxation (when the simultaneous update mode is selected [68])
- **Output settings**: where the experimenter can switch between the four algorithms and control the volume of the output monitor port.

A screenshot of this GUI is depicted in Figure 7.5.

7.7 Results and discussion

The centralized GEVD-based MWF and its distributed realization via the GEVD-based DANSE algorithm were both thoroughly explained in Chapter 3. Therefore, in this section only ingredients and modifications that are necessary to present the results for this evaluation setup are provided, and the reader is referred to Chapter 3 for more details and discussions about these algorithms.

7.7.1 General specifications

The acoustic sensor network consists of 3 nodes, where the data model at each node $k \in \mathcal{K} = \{1, 2, 3\}$ is described as (ω is the frequency-domain variable)

$$\mathbf{y}_k(\omega) = \underbrace{\mathbf{A}_k(\omega)\check{\mathbf{s}}(\omega)}_{\mathbf{s}_k(\omega)} + \mathbf{n}_k(\omega) \tag{7.3}$$

where $\mathbf{A}_k(\omega)$ is the steering matrix (containing *unknown* transfer functions between microphones and target speech sources), and where $\mathbf{y}_k(\omega)$, $\mathbf{s}_k(\omega)$ and $\mathbf{n}_k(\omega)$ denote the multi-channel microphone signals, speech signals and additive noise signals received by M_k microphones of node k. As discussed in Section 7.4.1, we have $M_1 = 4$, $M_2 = 3$ and $M_3 = 4$. The noise signals at each node contain both spatially correlated and uncorrelated components, i.e., in addition to signals from the localized noise sources, all microphone signals have an uncorrelated noise component which consist of Gaussian white noise with power 10^{-8} (in time-domain). In (7.3), $\check{\mathbf{s}}(\omega)$ contains the signals of S target speech sources.

The microphone signals have been sampled at f_s=16 kHz, and all the algorithms under evaluation (outlined in Section 7.2) have been implemented in the short-time Fourier transform (STFT) domain, with the DFT size equal to $L = 512$ (L

corresponds to the length of FIR filters in time-domain). A weighted overlap-add (WOLA) filterbank [181, 182] is employed with 50% overlap between the STFT frames. This introduces an algorithmic input/output (I/O) delay of $32ms$.

Similar to [32], to implement the algorithm in an adaptive fashion, the required second-order statistics are estimated in each frequency bin as (ω will be dropped for conciseness)

$$\textbf{if VAD=1}: \qquad \mathbf{R}_{yy,k}(t) = \lambda \mathbf{R}_{yy,k}(t-1) + (1-\lambda)\mathbf{y}_k(t)\mathbf{y}_k(t)^H \tag{7.4}$$

$$\textbf{if VAD=0}: \qquad \mathbf{R}_{nn,k}(t) = \lambda \mathbf{R}_{nn,k}(t-1) + (1-\lambda)\mathbf{y}_k(t)\mathbf{y}_k(t)^H \tag{7.5}$$

where $0 \leq \lambda < 1$ is a long-term exponential forgetting factor, and where t is the STFT frame index. In this setup, $\lambda = 0.99$ is chosen, which means that the correlation matrices are averaged over the past $1/(1-0.99) = 100$ frames.

A variant of the MWF which includes a trade-off between speech distortion and noise reduction, referred to as the speech-distortion-weighted MWF (SDW-MWF) [113], is employed in all the algorithms under evaluation. The SDW-MWF is computed as

$$\mathbf{w}_k = (\mathbf{R}_{ss} + \mu \mathbf{R}_{nn})^{-1} \mathbf{R}_{ss} \mathbf{e}_{\text{ref}} \tag{7.6}$$

where the speech distortion parameter has always been set to $\mu = 1$, and where $\mathbf{e}_{\text{ref}}$ is a column selector vector which selects the column of $\mathbf{R}_{ss}$ corresponding to the reference microphone. In this setup, always the first microphone of each node is considered as the reference, i.e., the microphone signals connected to the ports 1, 5, 8 of xPC are selected as the reference microphones of the nodes $k = 1, 2, 3$, respectively.

The GEVD-based DANSE algorithm assumes a fully-connected acoustic sensor network. In order to deliver a faster and smoother tracking performance, the nodes update simultaneously every time they have collected at least 100 new speech-and-noise and at least 100 new noise-only frames. The later is because after each update the statistics of $\mathbf{z}_k$ change, and therefore nodes should first collect a sufficiently large number of observations of $\mathbf{z}_k$ to be able to reliably estimate the new correlation matrices [10]. It is emphasized that the updates of the GEVD-based DANSE are spread over time in a time-recursive fashion, i.e., subsequent update iterations are performed on different signal frames. Note that to guarantee convergence when nodes update simultaneously, a relaxed update must be performed as follows [10, 68]:

$$\mathbf{w}_k^{\text{new}} = (1-\alpha)\mathbf{w}_k^{\text{old}} + \alpha(\mathbf{R}_{ss} + \mu \mathbf{R}_{nn})^{-1} \mathbf{R}_{ss} \mathbf{e}_{\text{ref}} \tag{7.7}$$

where $0 < \alpha \leq 1$ is the relaxation factor. In this evaluation, $\alpha = 0.5$ has been chosen, as this value was reported to eliminate limit cycles in every evaluated setup in [32, 68].

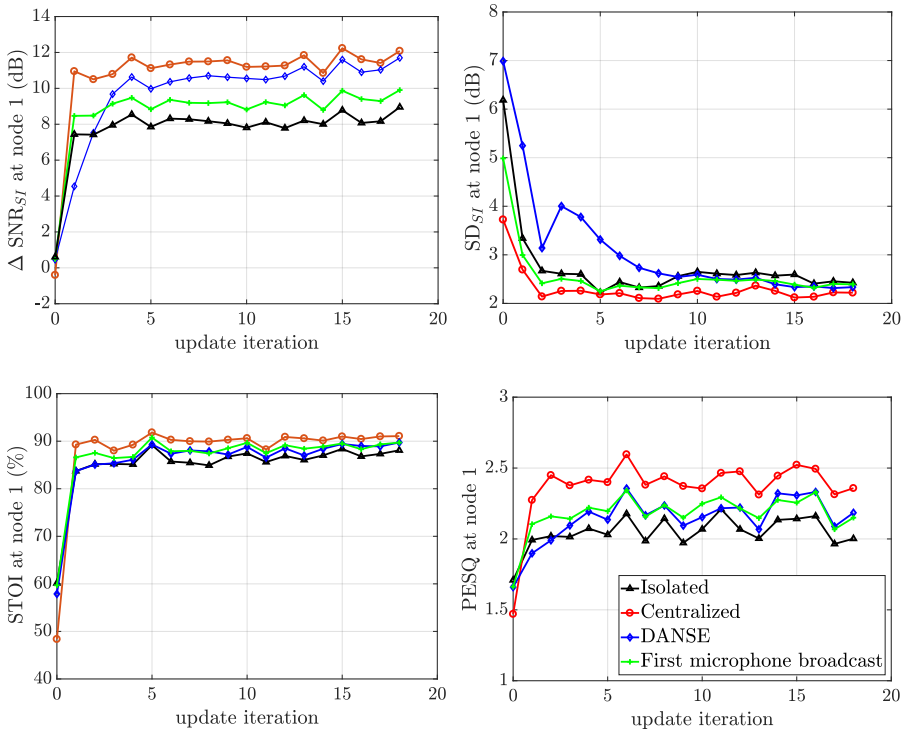


Figure 7.6: Comparison of speech intelligibility (left-column measures ΔSNR_{SI} and STOI) and quality (right-column measures SD_{SI} and PESQ) improvements at node 1 in the single target speaker scenario, averaged input $\text{SNR}_{SI} = 2.3\,\text{dB}$.

Per-node $\text{SNR}_{SI}^{\text{in}}$ values have been calculated from the reverberant speech and noise components at the first microphone of each node. In order to calculate the $\text{SNR}_{SI}^{\text{out}}$ values, the per-node filter coefficients (in the frequency-domain) computed by the speech enhancement algorithms have been first logged inside the xPC target machine in real-time when all the target speech and noise sources were present. Afterwards, these filter coefficients have been applied to the speech and noise signals which were recorded separately in exactly the same setup. The resulting filtered speech and noise signals have been used to calculate the $\text{SNR}_{SI}^{\text{out}}$ values. To facilitate the signal alignment procedure in this setting, a sine wave pulse with the frequency of 1 kHz was played at the end of each recording session. The speech signal of the reference microphone at each node was also used as the clean reference speech signal to compute the STOI and PESQ measures.

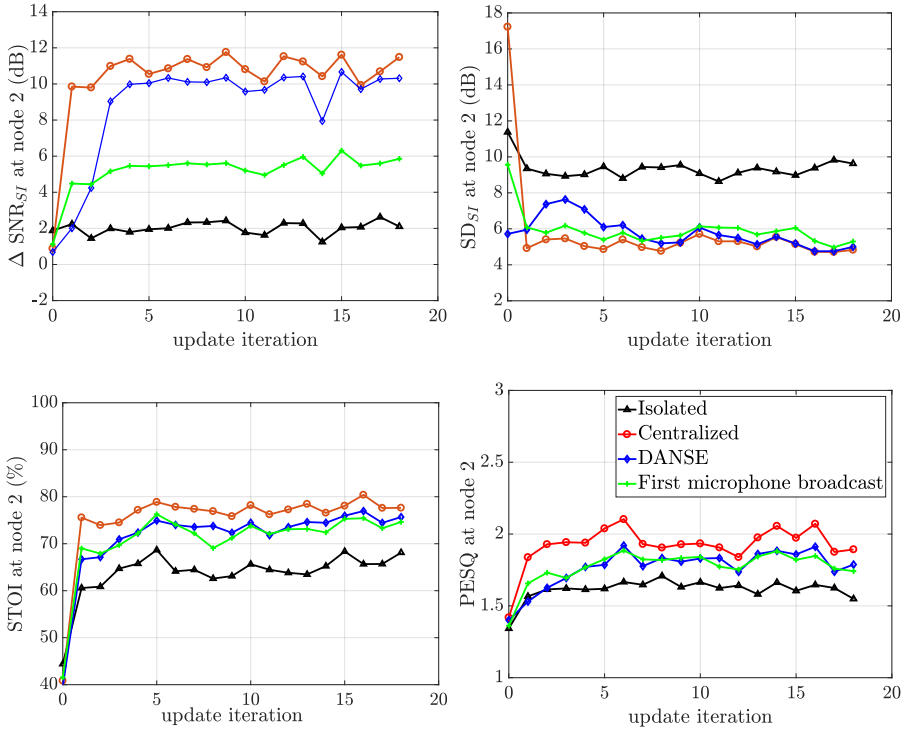


Figure 7.7: Comparison of speech intelligibility (left-column measures ΔSNR_{SI} and STOI) and quality (right-column measures SD_{SI} and PESQ) improvements at node 2 in the single target speaker scenario, averaged input $\text{SNR}_{SI} = 0.8\,\text{dB}$.

7.7.2 Single target speaker scenario

The performance evaluation has been firstly carried out in a scenario in which a single target speaker (speech source) is present, i.e., $S = 1$. This leads to speech correlation matrices $\mathbf{R}_{ss}$ which are approximately rank 1. To generate such a test scenario (see Figure 7.3), the loudspeaker LS1 produces 20 speech sentences (taken from HINT database), with 2 seconds of silence between every two consecutive sentences. A Pseudo-diffuse non-stationary noise field is generated by producing three multi-talker babble noise signals (mutually uncorrelated) using the loudspeakers LS2-LS4. To further introduce a stationary (yet spatial) noise to the scenario, the loudspeaker LS5 produces SSN noise (NOISEX-92). The power of multi-talker babble noise loudspeakers LS2-LS4 and that of loudspeaker LS5 are normalized so that the clean speech and noise

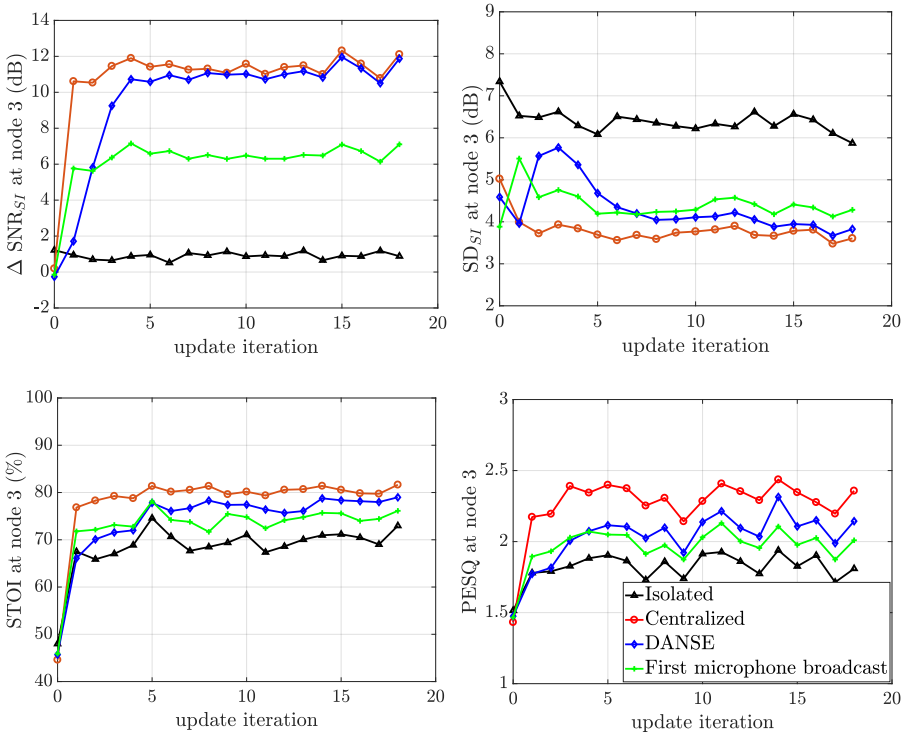


Figure 7.8: Comparison of speech intelligibility (left-column measures ΔSNR_{SI} and STOI) and quality (right-column measures SD_{SI} and PESQ) improvements at node 3 in the single target speaker scenario, averaged input $\text{SNR}_{SI} = 1.7\,\text{dB}$.

signals are mixed at long-term broadband $\text{SNR}_{SI}^{\text{in}}$ of $[2.3\,\text{dB}, 0.8\,\text{dB}, 1.7\,\text{dB}]$ at the first microphone of microphone arrays MA1-MA3, respectively. The ratios between the power of received multi-talker babble noise and the power of SSN noise at the first microphone of MA1-MA3 are calculated as [0.02 0.04 0.02], respectively.

To properly assess the algorithms and obtain smoother figures, the results have been averaged across 6 independent experiments where each used a different set of speech sentences and different noise segments.

The evaluation results using instrumental measures for speech intelligibility and quality for all the nodes are shown in Figures 7.6-7.8, where all the algorithms under evaluation have been assessed in the same conditions. In these figures, the

horizontal axes show the update iteration index at each node, where the updates happen simultaneously at all the nodes once they collect 100 speech-and-noise and 100 noise-only frames. This ensures that the nodes update once for each sentence, providing sufficient speech-and-noise and noise-only samples between two consecutive updates and further guarantee a valid output generated by the applied instrumental measures.

To be able to calculate the instrumental measures at each update iteration, the filter coefficients obtained from all four algorithms have been logged in real-time inside the xPC target machine each time the nodes update. Afterwards, the logged filter coefficients at iteration i have been applied to all the subsequent noise and speech segments[6] up until the time nodes update at iteration $i+1$. For instance the initialization filter $\mathbf{w}_k^0$ applies to the first noise period and the first speech sentence, then the first filter update $\mathbf{w}_k^1$ happens roughly at the end of sentence 1, and is applied to the second noise period and the second speech sentence, and so on. Note that therefore, the measures cannot be calculated for the last update after speech sentence 20.

The results in Figures 7.6-7.8 compare the performance of the GEVD-based DANSE with the isolated case (non-collaborative, lower-bound benchmark), centralized case (full-bandwidth collaborative, upper-bound benchmark) and the case where nodes only share their first microphone between each other (basic collaborative, same per-node cost as the GEVD-based DANSE algorithm).

In this evaluation, the speech enhancement performance delivered by these algorithms has been assessed in terms of speech intelligibility improvements using ΔSNR_{SI} and STOI (left-column plots), and in terms of speech quality improvements using SD_{SI} and PESQ (right-column plots).

The discussions about the points revealed by the provided results are given in the following:

- These results clearly show that the collaborative schemes (centralized, DANSE, first microphone broadcast) significantly outperform the non-collaborative scheme (isolated). The results also confirm that the benefits of using such collaborative schemes are more substantial at the low-SNR nodes. For instance, the ΔSNR_{SI} between the centralized and isolated cases at node 1 (high-SNR node) is about 3 dB, whereas it is about 10 dB at node 3 (low-SNR node). The same trend can be observed through the speech quality measures.

[6] As discussed in Section 7.7.1, for $\text{SNR}_{SI}^{\text{out}}$ values the logged filter coefficients have been applied to the speech and noise signals which were recorded separately in exactly the same setup.

- The GEVD-based DANSE algorithm outperforms the case where nodes merely share their first microphone signals between each other (basic cooperation). It is reiterated that these two algorithms demand the same computational power and communication bandwidth, but the GEVD-based DANSE employs an optimal filter-and-sum compressors at each node to generate the single-channel exchange signals. Although the difference between these two cases may seem less distinct when it comes to speech quality improvements at higher-SNR nodes (e.g., see right-column plots of Figure 7.6), it has been observed that the GEVD-based DANSE algorithm often delivers a slightly better performance after its convergence.

- The GEVD-based DANSE algorithm has slower convergence properties compared to the other cases, which is due to its time-recursive, iterative nature.

- As discussed in Chapter 3, the GEVD-based low-rank approximation effectively selects the 'modes' corresponding to the highest SNR. This can be verified by observing the ΔSNR_{SI} measures at all the nodes. Despite the fact that the node-specific speech signals estimated at the nodes have a maximal SNR level, they indeed have different quality and intelligibility characteristics.

- The results reveal a discrepancy between the centralized case and the GEVD-based DANSE algorithm (after its convergence). Nevertheless, the theoretical convergence analysis provided in Chapter 3 claimed that the GEVD-based DANSE algorithm perfectly converges to the centralized solutions. This was also verified through Simulation results. Note that the theoretical convergence analysis provided in Chapter 3 relied on the assumptions that (1) the signals are stationary, (2) the data model is perfectly satisfied, and (3) the required correlation matrices are error-free, i.e., they can be perfectly estimated. Nevertheless, none of these assumptions are ideally met in this realistic speech enhancement scenario. This is due to the fact that (1) here the algorithms are applied to enhance a speech signal, which is indeed a highly non-stationary signal, (2) to satisfy the instantaneous mixture model, the algorithms are implemented in the STFT domain since the microphone signals consist of convolutive mixtures of multiple signals. This requires the GEVD-based DANSE algorithm to be applied to each frequency bin separately. These per-bin problems, however, only approximately satisfy the data model (7.3) due to the inevitable 'spectral leakage' effect due to the finite-windowing in the time-domain as part of the STFT analysis, (3) the algorithms are implemented in an adaptive fashion where subsequent filter updates use estimated signal statistics based on relatively short signal segments. This generates larger estimation errors on the correlation matrices. Under such

conditions, therefore, the convergence cannot be perfectly achieved. It has been shown in [10] that choosing a shorter STFT block size L can reduce this discrepancy. Also, it is noted that even the centralized MWF does not completely converge due to similar variations in the estimated signal statistics.

- It is observed that the PESQ scores presented in Figures 7.6-7.8 are relatively low and only cover a range up to 2.6. As discussed in section 7.5, however, the full scale of PESQ measure ranges from -0.5 to 4.5. Note that the PESQ measure was originally developed to compare a clean reference speech signal with its processed (distorted yet still clean) speech signal. To assess the speech enhancement algorithms, however, here PESQ has been used to compare the quality of a *noisy* output speech signal with a clean reference speech signal. In such conditions, residual noise is regarded as a quality impairment and hence lowers the PESQ score. This justifies the relatively low PESQ scores that are shown in Figures 7.6-7.8.

7.7.3 Two target speakers scenario

The second acoustic test scenario evaluates the performance of the GEVD-based DANSE algorithm when two target speakers (speech sources) are present, i.e., when $S = 2$. This leads to speech correlation matrices $\mathbf{R}_{ss}$ which are approximately rank 2.

In Chapter 3, it was shown theoretically and through batch-mode stationary simulation results, that if the number of sources is underestimated as $\bar{S}$, the GEVD-based DANSE algorithm will still be able to converge to the centralized GEVD-based MWF which as well applies a rank-$\bar{S}$ approximation. This is in a way favorable since when $\bar{S} < S$, the GEVD-based DANSE requires less per-node communication bandwidth and computational power. However, despite its convergence properties, the performance of the GEVD-based DANSE algorithm as an adaptive distributed noise reduction algorithm for cases when $\bar{S} < S$ should be further evaluated. Therefore, this evaluation aims to assess the performance of this algorithm in a test scenario containing two target speakers, including the case where the number of sources is underestimated as $\bar{S} = 1$. This evaluation will be carried out using the ΔSNR_{SI} and SD_{SI} measures. Note that the isolated case and the case of basic cooperation will not be included in the evaluation results of this section, since Section 7.7.2 already revealed that the centralized and the GEVD-based DANSE algorithms outperforms these cases.

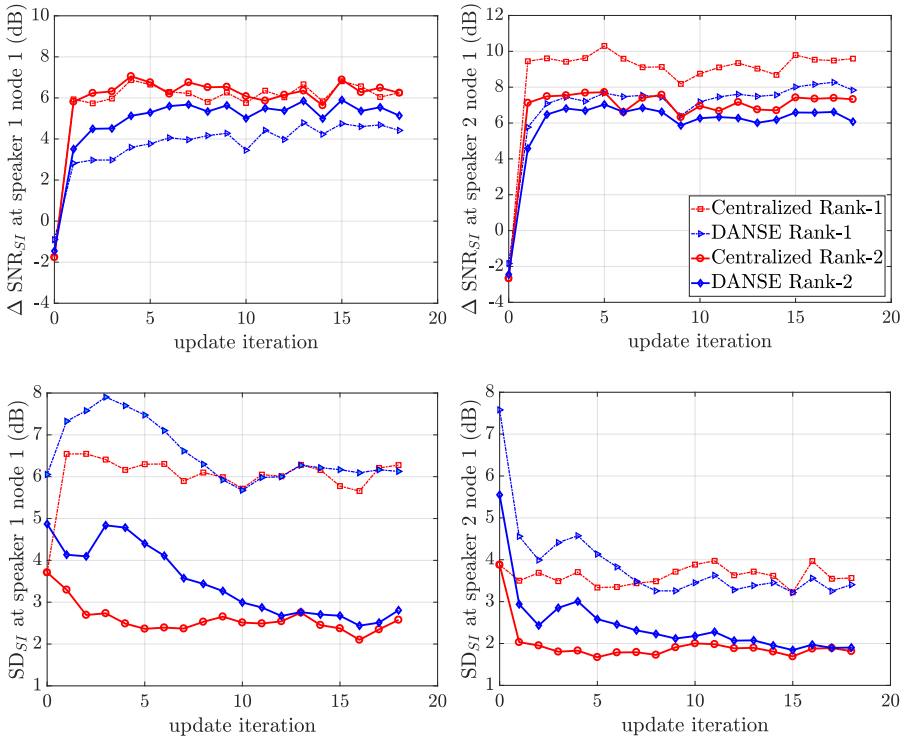


Figure 7.9: Comparison of speech intelligibility and quality improvements at node 1 in the two target speakers scenario, left-column for speaker one (averaged $\text{SNR}_{SI}^{\text{in}} = 3.2\,\text{dB}$), right-column for speaker two (averaged $\text{SNR}_{SI}^{\text{in}} = 3.6\,\text{dB}$).

To generate the acoustic test scenario for this evaluation, the loudspeakers LS1 and LS2 (see Figure 7.3) simultaneously produce 20 speech sentences (taken from HINT database), with 2 seconds of silence between every two consecutive sentences. The speech sentences played back from these loudspeakers are different, yet trimmed to have the same duration. In this scenario, a non-stationary noise field is generated by introducing two multi-talker babble noise signals (mutually uncorrelated) using the loudspeakers LS3 and LS4. To further introduce a stationary noise, an identical SSN signal (NOISEX-92) is played back from both channels of LS5 (with one piece placed closer to MA2). The power of the noise sources LS3-LS5 are normalized so that the resulting noise signals and the clean speech of the first target speech source (LS1) and the second target speech (LS2) are mixed individually at long-term broadband $\text{SNR}_{SI}^{\text{in}}$ of $[3.2\,\text{dB}, 0.1dB\,\text{dB}, 0.7\,\text{dB}]$ and $[3.6\,\text{dB}, 0.8\,\text{dB}, 1.2\,\text{dB}]$ at the first microphone

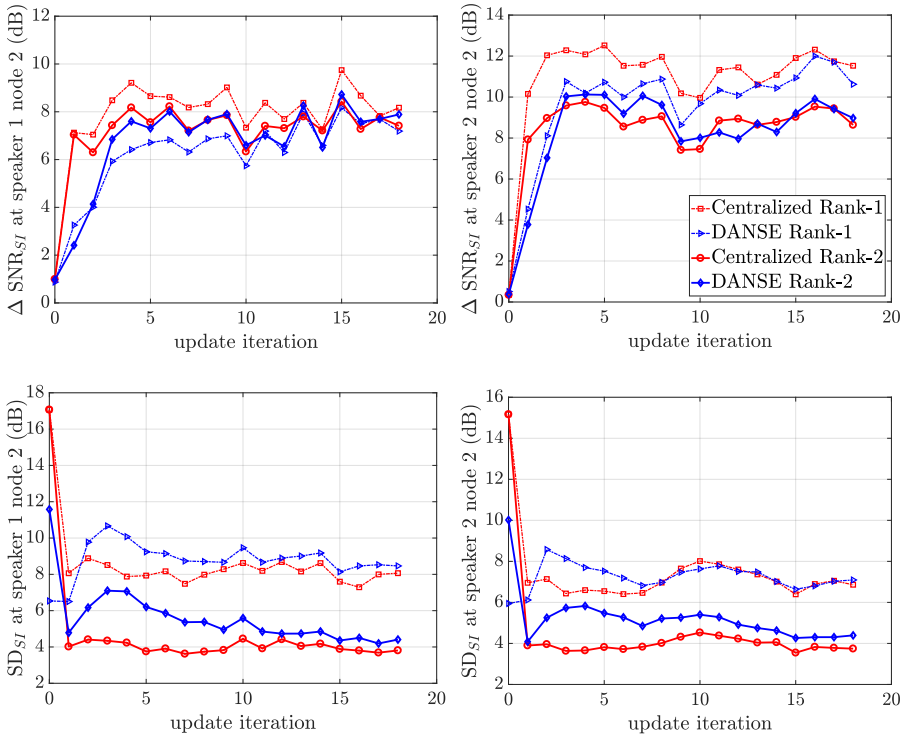


Figure 7.10: Comparison of speech intelligibility and quality improvements at node 2 in the two target speakers scenario, left-column for speaker one (averaged $\text{SNR}_{SI}^{\text{in}} = 0.1\,\text{dB}$), right-column for speaker two (averaged $\text{SNR}_{SI}^{\text{in}} = 0.8\,\text{dB}$).

of microphone arrays MA1-MA3, respectively. Note that all these SNR_{SI} values have been averaged across 6 experiments. Moreover, the SSN to multi-talker babble noise power ratios at the first microphone of MA1-MA3 are [0.08 0.05 0.09], respectively.

The evaluation results (averaged across 6 experiments) using instrumental measures for speech intelligibility (ΔSNR_{SI}) and quality (SD_{SI}) for all the nodes are shown in Figures 7.9-7.11. The discussions about the points revealed by the provided results are given in the following:

- It is observed that the centralized rank-1 MWFs often provides higher SNR_{SI} improvements than the centralized rank-2 MWFs, i.e., they deliver a more aggressive noise reduction. Moreover, the ΔSNR_{SI} differences

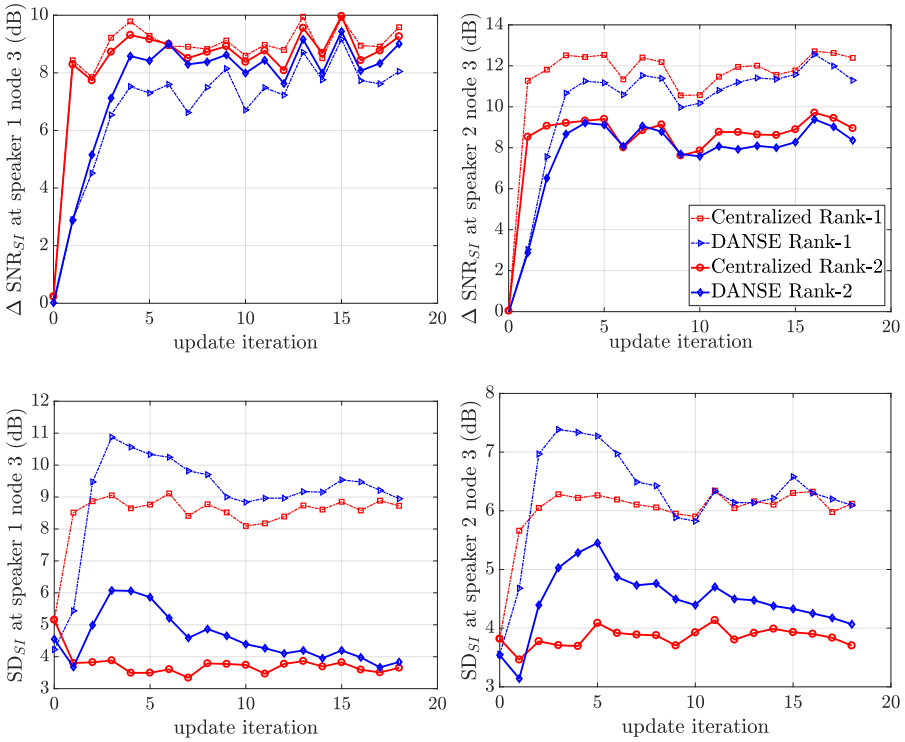


Figure 7.11: Comparison of speech intelligibility and quality improvements at node 3 in the two target speakers scenario, left-column for speaker one (averaged $\text{SNR}_{SI}^{\text{in}} = 0.7\,\text{dB}$), right-column for speaker two (averaged $\text{SNR}_{SI}^{\text{in}} = 1.2\,\text{dB}$).

between the centralized rank-1 and rank-2 MWFs appear to be more for one of the target speakers (in this case the one that has a higher SNR at each node, compare the top plots in Figures 7.9-7.11 and note that the second speaker (right-plots) has higher $\text{SNR}_{SI}^{\text{in}}$ at all the nodes). This may be due to the fact that since the centralized rank-1 MWFs try to maximize the SNR of a mixture of the two speech signals, it tends to do so by putting more effort into enhancing the higher SNR speech signal. As a result, although a maximal SNR is obtained, Figures 7.9-7.11 clearly show that this effort inevitably introduces a significant distortion (measured by the SD_{SI} measures) on both of the speech signals and hence impairs the quality of their mixture. However, these results show that a better trade-off can be achieved by applying the centralized rank-2 MWFs, with which the actual data model is indeed better satisfied. The latter

significantly improves the quality of both of the speech signals, while delivering a less aggressive noise reduction.

- It is observed that both of the rank-1 and rank-2 GEVD-based DANSE algorithms follow their corresponding centralized MWFs. The convergence of these cases is more evident when the SD_{SI} measures are used. Note that however when the ΔSNR_{SI} measures are used, as the different sentences used for filter updates have different $\text{SNR}_{SI}^{\text{in}}$ values, the figures are producing more fluctuations. As a result, even the centralized MWFs may not seem to be perfectly converging (as in top plots of Figure 7.10).

Chapter 8

Conclusion

8.1 Summary and conclusions

The thesis has introduced novel distributed algorithms for several homogeneous and heterogeneous multi-task WSNs in which nodes apply a compressive filter-and-sum scheme to significantly reduce the required communication bandwidth. As a result, nodes of these multi-task WSNs have only access to a subset of the full sensor signal observations. Nevertheless, the ultimate goal of all the developed algorithms is to provide a framework under which the nodes can still obtain the same estimation performance as in a centralized case, i.e., the case where all the sensor signal observations are available at once (at each node or at a fusion center). Furthermore, the design procedure in this thesis has always sought to effectively address the challenging design aspects discussed in the introduction, such as dealing with possible discrepancies between nodes, efficient use of communication bandwidth and computational resources, robustness to adverse environmental and algorithmic impacts, efficient use of exchange signals for integrated problems and blindness. In the remainder of this chapter, the benefits offered by the distributed algorithms developed in this thesis will be highlighted.

Communication bandwidth and per-iteration computational complexity reduction

The primary goal of all the derived distributed algorithms in this thesis was always to improve the performance of the per-node estimates by letting each

node access more spatially-distributed sensor signals via a cooperative scheme, i.e., by letting each node obtain its estimates based on the union of its own senor signal observations and those received from other nodes. However, as argued in the introduction, the most power-hungry operation at each node is known to be the wireless transmission of the sensor signals. Therefore, the distributed algorithms of this thesis all relied on compressive filter-and-sum operations to efficiently compress the exchanged data between nodes, leading to a significant reduction in the required communication bandwidth. Furthermore, as the compressor at each node was derived from the node's local spatial filters, such compressive scheme was indeed application-dependent and hence node-specific. As a result of this communication bandwidth reduction, computational complexity at each node was also reduced at each iteration since the operations were to be carried out on reduced-dimension matrices. Both of these reductions therefore directly extend the WSN's lifetime, since the energy required by each node to transmit and process the signals was reduced. Remarkably, despite the fact that the nodes only had access to a subset of the full sensor signals, under all the developed distributed algorithms all nodes achieved the corresponding centralized performance of their corresponding SP task, as if they had access to raw (uncompressed) signals of the entire WSN.

Improvements in per-node subspace estimation through cooperation

All subspace-based SP techniques employed in this thesis (MWF, MVDR, LCMV and DOA estimation) required the estimation of target and/or noise subspaces. In the WSNs envisaged in this thesis, the per-node signal subspace could be estimated locally without any signal exchange between nodes. However, chapter 2 revealed that these per-node estimates deliver better performance when obtained based on the centralized signal subspace, since more correlation structure could be exploited. These performance improvements in fact motivated the cooperative subspace estimation method which was implicitly integrated in all the distributed algorithms developed in the subsequent chapters.

Distributed estimation in homogeneous multi-task WSNs

As argued in the introduction, all nodes of a homogeneous multi-task WSN use the same SP technique(s) and therefore the same per-node SP routines to solve their different node-specific tasks. The first part of this thesis consisting of Chapters 2-4 has developed distributed algorithms for such homogenous multi-task WSNs, where the per-node routines were always designed following

a top-down strategy. As a result, nodes were fully aware of each other's tasks and an upper-layer coordination was applied to effectively tailor the per-node routines prior to operation of the corresponding WSNs. Under such conditions, the ultimate goal of each node was to obtain the same centralized estimation performance in a distributed fashion. In particular, Part I of this thesis offered the following distributed algorithms for such homogeneous WSNs:

Chapter 2 dealt with a homogeneous multi-task WSN in which nodes were tasked with estimating their own signal subspace based on a GEVD and in a distributed fashion. To achieve this, two distributed algorithms were presented where both of them allowed each node to obtain its corresponding part of the centralized signal subspace estimate with reduced per-node communication and computational cost. The algorithm which adopted the so-called inversion-based method performed the task based on the inversion of the matrices containing the GEVCs of a pair of reduced-dimension sensor signal covariance matrices at each node. This method exhibited a faster convergence speed compared to the so-called correlation-based method. It was also argued that the benefits of the former method come at the cost of more complex per-node computations.

Chapter 3 dealt with a homogeneous multi-task WSN in which nodes were tasked with estimating their own node-specific desired signals (desired source signals as received by their reference sensors) from noisy sensor signal observations in a distributed fashion. To achieve this, each node applied an MWF in which a GEVD-based low-rank approximation of signal correlation matrices was incorporated in the per-node local filter computations (and hence also in the per-node filter-and-sum compression rules). The resulting algorithm, referred to as GEVD-based DANSE, exhibited a better per-node estimation performance compared to the original DANSE algorithm, especially under conditions introducing higher estimation errors in signal correlation matrices, which may be due to non-stationary noise in low-SNR conditions or to erroneous VADs. Moreover, the conditions under which the GEVD-based DANSE algorithm converged to the centralized solutions were less strict than those of the original DANSE algorithm, in that the node-specific desired signals of the different nodes no longer needed to span the same low-dimensional latent signal subspace. This was because the low-rank structure of the data model was inherently imposed by the GEVD-based DASNE algorithm itself, by means of the GEVD. The GEVD-based DANSE algorithm also allowed further compression of the exchanged sensor signal observations compared to the compression rate of the original DANSE algorithm, with possibly reducing the estimation performance but still guaranteeing convergence. It has been shown theoretically and via simulations that the GEVD-based DANSE algorithm converges to the centralized GEVD-based MWF as if each node would have access to all the sensor signal observations.

Chapter 4 dealt with a homogeneous multi-task WASN in which nodes were tasked with estimating their node-specific DOAs with respect to a set of target speech sources, within an environment where these tasks were being hindered by the presence of localized noise sources (such as interfering talkers) or diffuse noise (such as multi-talker babble noise). To achieve robust node-specific DOA estimates at each node, the GEVD-based DANSE algorithm developed in Chapter 3 was first applied as a preprocessing step to denoise all the microphone signals of the WASN. Each node therefore attained two different tasks, i.e., noise reduction using the GEVD-based DANSE, and the GEVD-based DOA estimation. In this setup, it was shown that the exchanged compressed signals of the first task, i.e., of the GEVD-based DANSE algorithm, could also significantly increase the performance of the second task, i.e., the node-specific DOA estimates. These per-node improvements are indeed remarkable, since they were achieved even though (1) different nodes had different DOAs depending on their position, orientation and positions of target speech sources, (2) the position of the nodes, as well as their relative geometry and hence the between-nodes signal coherence models were unknown, (3) the compressed signals of the GEVD-based DANSE had actually been optimized to improve the performance of the noise reduction task and not the DOA estimation task. By re-using these compressed signals, the proposed distributed algorithm provided a cooperative integrated noise reduction and DOA estimation scheme, where the noise reduction (GEVD-based DANSE's final filtering stage) could actually be shortcut to either let the nodes estimate both their node-specific desired signals and their node-specific DOAs simultaneously, or to reduce the per-node computational cost when only the DOA estimation task had to be performed on the denoised signals.

Remarkably, all the above-mentioned distributed algorithms let each node of these homogeneous multi-task WSNs converge to their corresponding centralized estimates as if they all had access to all uncompressed sensor signal observations, even though the incorporated GEVD-based low-rank approximations were only applied locally at each node on the reduced-dimension signal correlation matrices.

Distributed estimation in heterogeneous multi-task WSNs

As argued in the introduction, nodes of a heterogeneous multi-task WSN use different SP techniques and therefore different per-node SP routines to solve their node-specific tasks. The second part of this thesis consisting of Chapters 5-6 has developed distributed algorithms for two particular heterogeneous multi-task WSNs which were formed by three different groups of nodes. It was assumed that the nodes had no prior knowledge about the SP techniques that other nodes

use. Nevertheless, the developed algorithms allowed nodes to immediately start exchanging their compressed signals with each other even though their per-node filter-and-sum compression rules were in these circumstances (1) extracted from different SP techniques (2) optimized at each node as if the other nodes were as well using the same SP technique (as in homogeneous multi-task WSNs presented in Part I). For these WSNs a bottom-up strategy could be followed, i.e., without upper-layer coordination to pre-define the per-node SP routines. In particular, Part II has developed the distributed algorithms for the following heterogeneous multi-task WSNs:

Chapter 5 presented a distributed algorithm for a particular heterogenous multi-task WASN in which some of the nodes aimed at estimating their node-specific DOAs with respect to a single desired speech source, while other nodes aimed at implementing either an MWF or an MVDR beamformer in order to estimate their node-specific signals of the desired source as it impinges on their local reference microphones. The developed distributed algorithm then lets these nodes exchange single-channel signals to improve the estimation performance of their own node-specific task.

Chapter 6 considered a more general heterogeneous multi-task WSN in which $S \geq 2$ sources were present. In this setting, some of the nodes applied an LCMV beamformer to extract the node-specific signals of their desired sources and to cancel out those of their interfering sources. In this case, each node could have a different set of node-specific constraints to define its desired/interfering sources. The other nodes then applied either MWF or DOA estimation to obtain the estimates corresponding to all the S sources. Moreover, a projection-based method was employed at the LCMV nodes to achieve reliable estimates even when limited relevant sensor data was available to estimate the required subspaces. A distributed algorithm was then developed for this particular heterogeneous multi-task WSN to let all the nodes exchange S-channel signals to improve the estimation performance of their own node-specific task.

In both of the above-mentioned heterogeneous multi-task WSNs, it was shown via theoretical results as well as multi-source simulations that the developed distributed algorithms allowed each node to attain the centralized solution of its own node-specific task with reduced communication resources. Remarkably, this was achieved even though each node was solving different signal processing task and did not know the tasks solved by the other nodes, and hence performed the same SP routines as it would have performed in a hypothetical homogeneous multi-task WSN where all the other nodes had used the same SP technique. The compatibility between the SP techniques of the different nodes was facilitated by incorporating a GEVD-based subspace estimation in the heterogeneous multi-task WSNs described above.

Efficient use of exchange signals for integrated noise reduction and DOA estimation

In Chapter 4, the developed distributed algorithm allowed the exchanged signals of the GEVD-based DANSE, which was primarily used for denoising the microphone signals of the WASN, to also be included in the per-node signal subspace estimation procedure to further improve the node-specific DOA estimates. This further led to an integrated distributed noise reduction and DOA estimation scheme, where the noise reduction (i.e., GEVD-based DANSE's final filtering stage) could actually be shortcut to reduce the computational cost at each node.

Robust subspace estimation in adverse conditions

All the SP techniques used in this thesis relied on the second-order statistics of the sensor signal observations to estimate the underlying signal subspaces. The estimation of such subspaces, however, can severely degrade in adverse conditions, e.g., when the WSN is exposed to highly non-stationary noise components or when a VAD provides an erroneous decisions due to low-SNR regimes. To better cope with such adverse conditions, a GEVD-based method was leveraged in this thesis to provide more robust estimates. Moreover, a subspace projection-based approach was proposed in Appendix A to improve the robustness of the LCMV beamformer in adverse conditions when only a limited number of relevant samples are available to accurately estimate the required subspaces.

Blindness

All the proposed distributed algorithms in this thesis were blind in the sense that they did not require any prior knowledge about the position of the sources and nodes. Hence, it was always assumed that the impulse responses (or equivalently the transfer functions) from the sources to the sensors are unknown. Whenever a node was tasked with a DOA estimation (Chapters 4-6), it was sufficient that only its *local* geometry was known, while their relative geometry and hence the between-nodes signal coherence model was always considered to be unknown.

8.2 Suggestions for future research

In addition to the challenging aspects that have been discussed and addressed in this thesis, further studies are necessary to investigate the following design aspects:

In Chapter 4 of this thesis, a distributed algorithm has been developed for a particular homogeneous multi-task WASN in which each node obtained an integrated noise reduction and DOA estimation scheme. As a result, the performance of both of these tasks were simultaneously enhanced using the same exchange signals optimized for the distributed noise reduction task. In general, such schemes that allow multiple tasks to effectively enhance their performance using the same amount of communication bandwidth are highly favorable. Therefore, it would be very relevant to explore the possibility of designing distributed algorithms that let multi-task WSNs integrate other tasks such as echo cancellation, feedback cancellation, active noise control, or dereverberation, without the need to exchange a dedicated signal for each individual task.

In this thesis, an optimal (centralized) solution has individually been defined for each and every task of the envisaged multi-task WSNs. The developed distributed algorithms then let all the nodes simultaneously converge to the optimal solutions of their own SP tasks. However, this convergence to optimal node-specific estimates cannot indeed be guaranteed in many other situations, e.g., when nodes have different interests or they do not have sufficient overlap in the set of target sources [129, 183], when sufficient bandwidth resources are not available to optimally fuse the exchanged signals, or when the different SP techniques used to fulfill the tasks are not well related. The latter may for instance happen in a heterogeneous multi-task WASN where nodes use different SP techniques to perform tasks such as echo cancellation, feedback cancellation, active noise control, or dereverberation. For these cases, alternatively a single global cost function can be defined as the *sum* of all local node-specific cost functions of different node-specific SP tasks. In this setting, however, there often exists no optimal solution that can simultaneously let all nodes obtain the optimal solutions of their own SP tasks. This essentially happens because any attempt to obtain an optimal solution for a particular local node-specific cost function leads to sub-optimal solutions for all the other local node-specific cost functions (e.g., due to bandwidth limitations). In other words, the performance of a particular task cannot improve unless the performance of other tasks worsen. In such conditions, the global cost function can define the solutions in a more general concept of optimality, referred to as *Pareto-optimality* [184]. Several algorithms for distributed parameter estimation have been proposed to enable the nodes of such WSNs to obtain the Pareto-optimal solutions, using e.g., incremental-based strategies [61, 185, 186], diffusion-based strategies [187, 188]

and consensus-based strategies [185, 189]. However, all of these methods rely on an iterative re-estimation process, which may not be suitable for real-time estimation problems. Therefore, future research efforts should be focused on the design of distributed algorithms that let the nodes of such multi-task WSNs obtain these Pareto-optimal solutions, while adopting a more efficient communication scheme and while offering a low-latency regime for real-time signal and parameter estimation problems (e.g., by letting the iterations spread over time).

The algorithms designed in this thesis assumed that all the nodes observe the full subspace spanned by all the source signals (including the desired and interfering sources). This means that all the nodes of the envisaged WASNs were assumed to be able to 'hear' all speech sources present in the acoustic enclosure. However, it may happen that this assumption is not perfectly satisfied, e.g., when a node of a WASN is placed somewhere far away from a speech source, or when its microphones are highly polluted by a nearby noise source. In such conditions, the nodes may not be able to attain their corresponding centralized solutions anymore, and hence the solutions may become suboptimal [10] or a stronger compression may be possible [130]. As a result, extra studies need to be undertaken, including an extended theoretical analysis for the cases where the full underlying subspace is not necessarily observable by all nodes of a WSN.

In a cooperative scheme, it is possible that for privacy reasons some nodes of a WASN do not wish to share their (compressed) signals containing their conversation content. In these circumstances, some privacy constraints need to be placed in conjunction with the primary design constraints (e.g., those for maximizing performance or minimizing the communication bandwidth). Such privacy preservation aspects in general become more crucial in WSNs where different devices with different tasks are willing to cooperate while preserving their privacy. To achieve this, an encryption-based method can be applied to the exchange signals [190, 191]. However, these methods may strongly increase the per-node computational complexity and may make the solution of the primary task suboptimal. Therefore, it will be important to further study the feasibility of designing distributed algorithms that could possibly let the privacy preservation and optimal solutions with respect to the primary constraints be simultaneously achieved, while still relying on efficient operations in terms of the computational complexity and communication bandwidth.

To exclude potential imperfections in practical scenarios, all the algorithms developed in this thesis assumed that the sampling rates between the nodes are synchronized. However, due to intrinsic differences in operating conditions of different devices, there often exists a mismatch between the sampling rates of nodes. Such mismatch may become even larger in WSNs formed by different devices in which each device operates at a different nominal sampling rate. This

can lead to a significant performance degradation in the developed distributed algorithms, since they all assumed a time-invariant coherence between the different sensor signal observations. Therefore, it is of great value to further study efficient methods that can detect and compensate such sampling frequency offsets in a blind framework (as in e.g., [192, 193]) and integrate them in the multi-task WSNs which rely on (different) coherent SP techniques.

It is possible that the topology of a WSN changes in between iterations, e.g., because of mobility of the nodes, communication link failures, or malfunction of nodes. Distributed algorithms that let the nodes still converge to the centralized estimates without having to re-converge to a new set of parameters for the new topology, are indeed more desirable. Therefore, further research efforts have to be carried out towards achieving this goal.

Appendix A

Subspace projection for robust LCMV beamforming

LCMV beamforming with subspace projection for multi-speaker speech enhancement

Amin Hassani, Alexander Bertrand and Marc Moonen

Published in proceedings of the *IEEE International Conference on Acoustics, Speech and Signal processing (ICASSP)*, Shanghai, China, Mar. 2016, pp. 91-95, DOI: 10.1109/ICASSP.2016.7471643.

Abstract

The linearly constrained minimum variance (LCMV) beamformer has been widely employed to extract (a mixture of) multiple desired speech signals from a collection of microphone signals, which are also polluted by other interfering speech signals and noise components. In many practical applications, the LCMV beamformer requires that the subspace corresponding to the desired and interferer signals is either known, or estimated by means of a data-driven procedure, e.g., using a generalized eigenvalue decomposition (GEVD). In practice, however, it often occurs that insufficient relevant samples are available to accurately estimate these subspaces, leading to a beamformer with poor output performance. In this section we propose a subspace projection-based approach to improve the performance of the LCMV beamformer by exploiting the available data more efficiently. The improved performance achieved by this approach is demonstrated by means of simulation results.

A.1 Introduction

Sensor arrays allow space-time signal processing which often improves the performance of parameter- or signal-of-interest estimation, when compared with single-sensor based estimation [3, 4]. In audio and speech enhancement applications, microphone arrays have been widely used [2]. A common problem is to extract (a mixture of) multiple desired speech signals from the microphone signals, which are also polluted by other interfering speech signals and noise components. To solve this problem, one can use a so-called beamforming approach [109]. A well-known approach is linearly constrained minimum variance (LCMV) beamforming which aims at minimizing the total power of the beamformer output, under a set of linear constraints that control the array beam pattern such that the signals coming from the desired directions remain undistorted while signals coming from the interfering directions are rejected [109].

Basically, there are two main classes of LCMV beamformers. The first class assumes that each individual room impulse response (RIR) (or equivalently acoustic transfer function (ATF)) between each source and each microphone is known [134]. In this case, the LCMV beamformer will estimate the mixture of desired source signals (that have not yet been distorted by the RIRs). The second class deals with cases where the RIRs are not known *a priori* and hence have to be estimated on the fly based on statistical properties of the microphone signals. This class is often referred to as *blind* LCMV beamforming [134, 161]. In practice however, estimating individual RIRs may not be straightforward, as it is usually required that there are sufficient signal segments in which only *one* of the sources is active, i.e., for each of the individual desired and interfering sources [194]. In [134, 161], the authors proposed a beamforming framework in which the unknown ATFs of the desired and interfering sources are replaced by respective bases for the desired sources and interfering sources subspaces spanned by the columns of the true ATFs. The resulting response then estimates the mixture of the desired source signals as observed by an arbitrarily chosen *reference microphone* and suppresses the interfering source signals. In this section we consider such an LCMV beamformer for which the desired sources and interfering sources subspaces must be estimated based on the microphone signals.

To estimate the desired sources and interfering sources subspaces, an eigenvalue decomposition- (EVD-) based approach can be used (as in [195]). However, a generalized EVD (GEVD-) based subspace estimation is better suited for scenarios with spatially correlated noise, as it directly incorporates the estimated noise covariance matrix such that each resulting subspace estimate is aimed to have the highest output signal to noise ratio (SNR) [103, 119, 120].

For the estimation of the desired sources and interfering sources subspaces, in practice we must first construct the relevant sample covariance matrices based on the time segments during which only the desired or interfering sources are active, namely *'desired-sources-only'* and *'interfering-sources-only'* segments, respectively. This procedure in practice requires a voice activity detector (VAD) that is able to distinguish between such segments (e.g., as in [163]). Note that in this way the samples from the segments during which both the desired source(s) and the interfering source(s) are simultaneously active will be discarded for the estimation of the individual subspaces. In practice, however, it often happens that insufficient 'desired-sources-only' and 'interfering-sources-only' samples are available to accurately estimate these individual subspaces. In this section we propose a subspace projection-based approach which improves the output performance of the blind LCMV beamformer based on the projection of the individual subspace estimates onto the joint signal subspace of all the desired and interfering sources present in the environment. Basically, the motivation behind this is the fact that now all segments can be involved for the estimation of the joint signal subspace (except for '*noise-only*' segments). Hence the accuracy and tracking performance of this joint subspace estimation is expected to be higher compared to the individual subspace estimates.

A.2 Data model and problem statement

We consider a microphone array with M microphones, in which the captured signal at microphone m, $m = 1, \ldots, M$ can be described in the frequency domain as

$$y_m(\omega) = d_m(\omega) + i_m(\omega) + n_m(\omega) \tag{A.1}$$

where $d_m(\omega)$ is the desired source signals component and $i_m(\omega)$ is the interfering speech signals component, and where $n_m(\omega)$ denotes the additive noise component which includes both spatially correlated and uncorrelated noise contributions. Although $i_m(\omega)$ can also be considered as noise, it is not included in $n_m(\omega)$, because we aim to control the suppression of the interferers, possibly targeting a complete removal. In (A.1), ω is the discrete frequency-domain variable where the resolution is defined by the discrete Fourier transform (DFT) of size L. For the sake of brevity, ω will be omitted in the sequel. We assume that there are N_d desired speech sources, and N_i interfering speech sources and that these numbers are known (although they could also be estimated in practice). Hence $d_m = \sum_{d=1}^{N_d} a_{dm} s_d$ and $i_m = \sum_{i=1}^{N_i} a_{im} s_i$, where a_{dm} and a_{im} denote the ATFs from the desired speech source s_d and the interfering speech source s_i to microphone m, respectively. The stacked version of all microphone

signals is represented as

$$\mathbf{y} = \mathbf{A}_d\mathbf{s}_d + \mathbf{A}_i\mathbf{s}_i + \mathbf{n} \triangleq \mathbf{d} + \mathbf{i} + \mathbf{n} \tag{A.2}$$

where $\mathbf{A}_d = [\mathbf{a}_{d_1} \dots \mathbf{a}_{d_{N_d}}]$, $\mathbf{A}_i = [\mathbf{a}_{i_1} \dots \mathbf{a}_{i_{N_i}}]$ are $M \times N_d$ and $M \times N_i$ steering matrices, respectively, with $\mathbf{a}_x$ denoting the RIR (ATF) from the source x to the microphone array. In (A.2), $\mathbf{s}_d$ and $\mathbf{s}_i$ are stacked signal vectors containing the N_d desired speech source signals and N_i interfering speech source signals, respectively.

In this section we consider the problem of extracting the mixture of the desired speech signals as it is observed at the reference microphone, from the noisy microphone signals $\mathbf{y}$ and with an LCMV beamformer. This extraction is assumed to be carried out in scenarios where insufficient 'desired-sources-only' and 'interfering-sources-only' samples are available to accurately estimate the individual subspaces spanned by the columns of $\mathbf{A}_d$ and $\mathbf{A}_i$, respectively.

A.3 LCMV beamforming

LCMV beamforming in general applies a linear M-dimensional estimator $\mathbf{w}$ to the M-channel signal $\mathbf{y}$ to estimate the desired signal $\bar{d} = \mathbf{w}^H\mathbf{y}$, where H denotes the conjugate transpose operator, and where overline $(\bar{.})$ denotes the estimate. To design an LCMV beamformer that estimates the unreverberated source signals $\mathbf{s}_d$, the steering matrices $\mathbf{A}_d$ and $\mathbf{A}_i$ have to be known [134]. When instead of estimating $\mathbf{s}_d$ the aim is to estimate mixture of the desired speech signals as captured by the reference microphone, a modified LCMV beamformer can be designed which requires only estimates of $\mathbb{Q}_d$ and $\mathbb{Q}_i$, where $\mathbb{Q}_d$ is an $M \times N_d$ matrix at which the columns define a unitary basis for the desired sources subspace spanned by the columns of $\mathbf{A}_d$, and where $\mathbb{Q}_i$ is an $M \times N_i$ matrix where its columns define a unitary basis for the interfering sources subspace spanned by the columns of $\mathbf{A}_i$ [134,161]. The resulting LCMV problem can be expressed as [161]

$$\min_{\mathbf{w}} E\{|\mathbf{w}^H\mathbf{y}|^2\} \tag{A.3}$$

$$s.t.\ \mathbb{Q}^H\mathbf{w} = \mathbf{f} \tag{A.4}$$

where $\mathbb{Q} \triangleq [\mathbb{Q}_d\ \mathbb{Q}_i]$, and where $\mathbf{f}$ is the vector of desired responses defined as $\mathbf{f} = [\mathbb{q}_d^T\ \mathbf{0}]^T$ where $\mathbb{q}_d$ is the j-th column of $\mathbb{Q}_d^H$, with j denoting the reference microphone. In the sequel and without loss of generality (w.l.o.g.), we assume

that the first microphone is chosen as the reference microphone, i.e., $j = 1$. The solution of (A.3)-(A.4) is then given by

$$\mathbf{w} = \mathbf{R}_{yy}^{-1}\mathbb{Q}(\mathbb{Q}^H\mathbf{R}_{yy}^{-1}\mathbb{Q})^{-1}\mathbf{f}. \tag{A.5}$$

Note that (A.5) has to be computed for each frequency bin separately. The resulting output signal, namely $\bar{d}_{\text{ref}}$, can be then described as $\bar{d}_{\text{ref}} = \mathbf{w}^H\mathbf{y} = \sum_{d=1}^{N_d} a_{d1}s_d + \hat{\mathbf{w}}^H\mathbf{n}$, which verifies the fact that this solution estimates the mixture of the desired speech signals as captured by the first microphone, while fully cancelling the interfering speech signals and while suppressing the ambient noise as much as possible [161].

To estimate $\mathbb{Q}_d$ and $\mathbb{Q}_i$ based on the microphone signal $\mathbf{y}$ in (A.2), we first define the following source-activity-based correlation matrices:

$$\mathbf{R}_{yy}^{d} = \mathbf{A}_d\mathbf{\Pi}_d\mathbf{A}_d^H + \mathbf{R}_{nn} \tag{A.6}$$

$$\mathbf{R}_{yy}^{i} = \mathbf{A}_i\mathbf{\Pi}_i\mathbf{A}_i^H + \mathbf{R}_{nn} \tag{A.7}$$

where $\mathbf{\Pi}_d = \text{diag}\{P_{d_1} \dots P_{d_{N_d}}\}$ and $\mathbf{\Pi}_i = \text{diag}\{P_{i_1} \dots P_{i_{N_i}}\}$, with P_x being the power of x-th source signal, and where $\mathbf{R}_{nn} = E\{\mathbf{n}\mathbf{n}^H\}$. Note that in practice the correlation matrices $\mathbf{R}_{yy}^{d}$ and $\mathbf{R}_{yy}^{i}$ can be estimated via sample averaging over the 'desired-sources-only' and 'interfering-sources-only' segments, respectively, requiring an oracle algorithm that can distinguish between these segments [134, 161]. An EVD-based approach can then be used to estimate the subspaces (e.g., as in [134, 161, 195]). In [134, 161], the authors proposed a procedure to choose a set of N_d and N_i eigenvectors (EVCs) of $\mathbf{R}_{yy}^{d}$ and $\mathbf{R}_{yy}^{i}$ that span the same subspace as $\mathbb{Q}_d$ and $\mathbb{Q}_i$, respectively.

A.4 Projection-based subspace estimation

The estimation of $\mathbb{Q}_d$ and $\mathbb{Q}_i$, as explained in Section A.3, may yield poor results if (A.6) and (A.7) can not be accurately estimated, e.g., when there are insufficient 'desired-sources-only' and/or 'interfering-sources-only' segments or samples available. Indeed, in the procedure described in Section A.3, a large part of the data is not used, namely the signal segments in which desired and interfering sources are simultaneously active. In this section, we propose a method which also exploits these signal segments, which leads to an improved speech enhancement performance.

Here we employ a GEVD-based subspace estimation although a similar strategy can be used for other subspace estimation techniques. Define $\mathbf{X}_d$ and $\mathbf{X}_i$ as $M \times M$ matrices containing the generalized EVCs (GEVCs) of the ordered

matrix pair $(\mathbf{R}_{yy}^{d}, \mathbf{R}_{nn})$ and $(\mathbf{R}_{yy}^{i}, \mathbf{R}_{nn})$, respectively, in their columns. Note that $\mathbf{R}_{nn}$ can be estimated from the 'noise-only' segments when all the desired and interfering speech sources are inactive. We assume (w.l.o.g.) that 1) the GEVCs are sorted such that their corresponding generalized eigenvalues (GEVLs) are sorted in descending order 2) the GEVCs are scaled such that $\mathbf{X}^H \mathbf{R}_{nn} \mathbf{X} = \mathbf{I}_M$. Now let $\mathbf{Q}_d = (\mathbf{X}_d)^{-H}$ and $\mathbf{Q}_i = (\mathbf{X}_i)^{-H}$. It can then be verified that the first N_d columns of $\mathbf{Q}_d$ and the first N_i columns of $\mathbf{Q}_i$ span the same subspace as $\mathbb{Q}_d$ and $\mathbb{Q}_i$, respectively [119].

As mentioned earlier, because of insufficient 'desired-sources-only' or 'interfering-sources-only' segments, $\mathbb{Q}_d$ and $\mathbb{Q}_i$ will be poorly estimated, which may often result in inadequate LCMV beamforming outputs. In such conditions we propose the following subspace projection-based approach such that the discarded samples associated with the segments during which the desired and interfering sources are simultaneously active can also be exploited. Excluding the samples of the 'noise-only' segments, all other segments are then used to estimate

$$\mathbf{R}_{yy}^{d,i} = \mathbf{A}_d \mathbf{\Pi}_d \mathbf{A}_d^H + \mathbf{A}_i \mathbf{\Pi}_i \mathbf{A}_i^H + \mathbf{R}_{nn}. \tag{A.8}$$

We now define $\mathbf{X}_{d,i}$ as the full-rank matrix containing the GEVCs of the ordered matrix pair $(\mathbf{R}_{yy}^{d,i}, \mathbf{R}_{nn})$. The joint $(N_d + N_i)$-dimensional desired sources and interfering sources subspace $\mathbb{Q}_{d,i}$ can then be defined as the first $(N_d + N_i)$ columns of the matrix $\mathbf{Q}_{d,i} = (\mathbf{X}_{d,i})^{-H}$.

Note that in theory, the columns of $\mathbb{Q}_{d,i}$ and the columns of $[\mathbb{Q}_d \ \mathbb{Q}_i]$ span the same signal subspace. In practice however, because of the discrepancies between the data segments based on which the correlation matrices (A.6)-(A.8) are estimated, this does not hold anymore. This can be corrected by the projection of the poorly estimated $\mathbb{Q}_d$ and $\mathbb{Q}_i$ onto the joint subspace estimate $\mathbb{Q}_{d,i}$. Hence we define the projected individual subspace estimates as

$$\mathbb{Q}_d^{\text{proj}} \triangleq \mathbb{Q}_{d,i} (\mathbb{Q}_{d,i}^T \mathbb{Q}_{d,i})^{-1} \mathbb{Q}_{d,i}^T \mathbb{Q}_d \tag{A.9}$$

$$\mathbb{Q}_i^{\text{proj}} \triangleq \mathbb{Q}_{d,i} (\mathbb{Q}_{d,i}^T \mathbb{Q}_{d,i})^{-1} \mathbb{Q}_{d,i}^T \mathbb{Q}_i \tag{A.10}$$

The subspace projection-based version of the LCMV beamformer solution (A.5) can then be expressed as

$$\mathbf{w}_{\text{proj}} = (\mathbf{R}_{yy}^{d,i})^{-1} \mathbb{Q}_{\text{proj}} (\mathbb{Q}_{\text{proj}}^H (\mathbf{R}_{yy}^{d,i})^{-1} \mathbb{Q}_{\text{proj}})^{-1} \mathbf{f}_{\text{proj}} \tag{A.11}$$

where $\mathbb{Q}_{\text{proj}} \triangleq [\mathbb{Q}_d^{\text{proj}} \ \mathbb{Q}_i^{\text{proj}}]$ and where $\mathbf{f}_{\text{proj}} \triangleq [\mathfrak{q}_{\text{proj}}^T \ \mathbf{0}]^T$, with $\mathfrak{q}_{\text{proj}}$ being the first column of $(\mathbb{Q}_d^{\text{proj}})^H$. The actual output of the beamformer (A.11), i.e., $\bar{d}_{\text{proj}} = \mathbf{w}_{\text{proj}}^H \mathbf{y}$, will be evaluated in the next section via simulation results.

A.5 Simulation results

In this section, the improved performance achieved by the subspace projection-based LCMV solution (A.11) is demonstrated by means of simulation results. For this goal, two different scenarios are simulated. The first scenario assumes multiple desired and multiple interfering sources in the enclosure, with narrowband source signals. This scenario allows us to easily perform Monte Carlo (MC) simulations to better investigate the benefits of the proposed approach in different conditions. The second scenario tests the proposed approach for multi-talker speech enhancement where the desired and interfering sources produce different speech signals (English sentences).

A.5.1 Simulated scenario with narrowband source signals

A setup with different position of nodes and sources, and with different narrowband source signals is considered in each MC run. Further specifications of this scenario are as follows: $M = 10$, $N_d = 2$, $N_i = 3$, total number of samples= 20000, number of samples in which both desired and interfering sources are active= 7000 and MC runs= 1000. Number of available 'desired-sources-only' and 'interfering-sources-only' samples, namely Nb_{only}, are assumed to be equal. Nb_{only} is then varied from 1 to 5000 (see Figure A.1). The remaining $13000 - 2Nb_{\text{only}}$ samples are 'noise-only'. All desired and interfering sources have the same power P. The noise consists of two randomly placed spatial noise sources with power $0.5P$, as well as uncorrelated noise on each sensor which is 5% of the power of the first desired source as observed on the first sensor. The entries of the steering matrices are independently drawn from a uniform distribution over the interval $[0.5; 0.5]$. The same holds for the samples of all the involved source signals, followed by a proper scaling to modify their power. As a performance measure, we utilized the output *signal to interference plus noise ratio* (oSINR) at the reference sensor, defined as

$$\text{oSNRI} = 10\log_{10}\frac{E\{|\mathbf{w}^H\mathbf{d}|^2\}}{E\{|\mathbf{w}^H\mathbf{i}|^2\} + E\{|\mathbf{w}^H\mathbf{n}|^2\}} \tag{A.12}$$

(expectations are taken over all frequency-time points). Figure A.1 compares the output oSNRI of the proposed subspace projection-based LCMV beamformer (A.11) to that of (A.5), as a function of the number of available 'desired-sources-only' and 'interference-sources-only' samples. As can be seen, the proposed approach significantly outperforms when insufficient 'desired-sources-only' and 'interference-only' samples are available. Note that two figures eventually converge to each other when sufficiently large number of relevant samples are available.

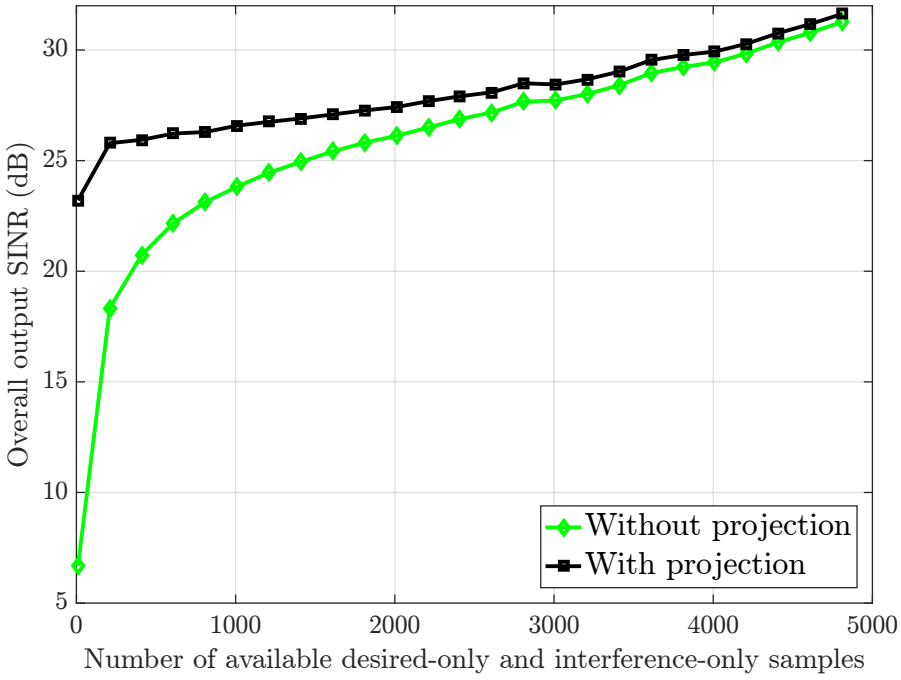


Figure A.1: MC results based on narrowband source signals

A.5.2 Multi-talker speech enhancement

In this scenario we simulate a cubic room with dimensions $5m \times 5m \times 5m$ and with surface reflection coefficient $\beta = 0.2$ using the image method [152]. The RIRs were simulated based on the modified version in [164]. A uniform linear microphone array consisting of $M = 10$ omni-directional microphones with inter-microphone distance of $5cm$ is considered where the center microphone is located at the position $[x = 2.5m, y = 1.5m]$. A desired speech source, an interfering speech source and a babble noise source is located at $[x = 1m, y = 2m]$, $[x = 4m, y = 2m]$ and $[x = 2.5m, y = 3.5m]$, respectively. We use a sampling frequency of $F_s = 16kHz$, a Hann-windowed DFT with size $L = 512$ and with 50% overlap. To avoid including the effect of VAD errors, we here use an ideal VAD with the ability of distinguishing between the desired and interfering speech sources. Both the desired and interfering speech sources produce short sentences with the same power $P_s = P_i$, with 7 seconds of overlapping activity and with some silence periods in between sentences (see top plot of Figure A.2). The power of the babble noise source is $0.5P_s$. An additional spatially

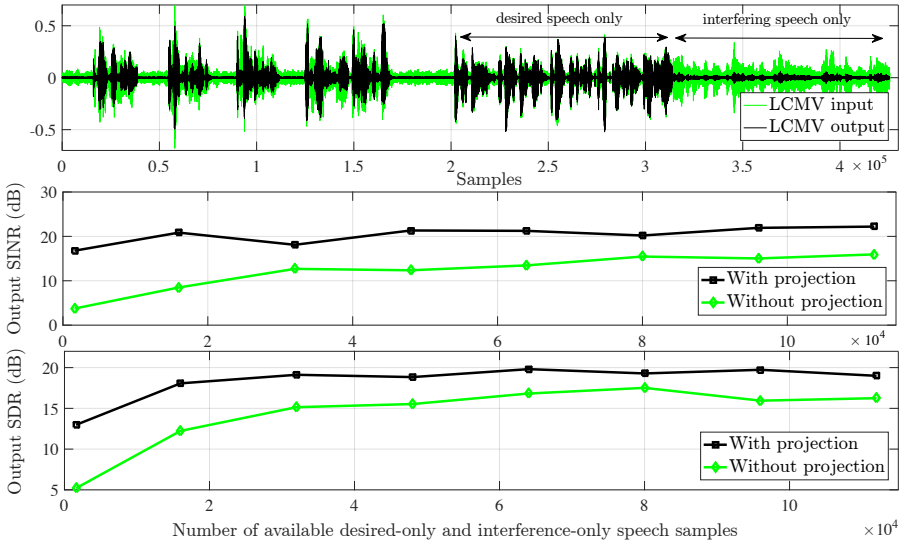


Figure A.2: Results based on a simulated room with speech signals

uncorrelated noise component at each microphone is simulated with a white Gaussian signal with 5% of the power of the desired speech signal as observed at the first microphone. To evaluate the performance, we again increase the number of available samples in 'desired-sources-only' and 'interfering-sources-only' segments, varying from $0.1F_s$ to $7F_s$. Besides oSINR, we here also consider the output signal to distortion ratio (oSDR) at the first microphone, defined as

$$\text{oSDR} = 10\log_{10}\frac{E\{|\mathbf{d}|^2\}}{E\{|\mathbf{d}-\mathbf{w}^H\mathbf{d}|^2\}} \tag{A.13}$$

In the simulated scenario, input SNR$\approx 9.5dB$, input SIR$\approx 2.2dB$ and input SINR$\approx 1.5dB$, measured at the first microphone. The middle and bottom part of Figure A.2 evaluate the performance of the LCMV beamformer output with the projection-based approach in terms of the output SINR and SDR. These convincing results again verify that the proposed projection-based approach delivers a significantly better performance. This improvement is indeed achieved at the cost of more complex computations due to the need for the computation of the full joint subspace $\mathbb{Q}_{d,i}$ which in turn requires to perform an extra GEVD. Note that a sufficiently large number of available samples lets the plots in Figure A.2 to converge to each other (not shown here).

A.6 Conclusion

In this section, we have proposed a subspace projection-based approach to increase the performance of an LCMV beamformer in conditions where insufficient relevant samples are available to accurately estimate the subspaces of the desired sources and interfering sources, respectively. We have considered a GEVD-based method for subspace estimation in combination with a subspace projection step, which allows to better estimate the desired sources and interfering sources subspaces. This improvement is achieved at the cost of more complex computations, as the poorly estimated subspaces have to be projected onto the larger joint subspace, which itself requires an extra GEVD. The improved performance achieved by this subspace projection-based approach has been demonstrated by means of simulation results.

Appendix B

APPENDIX to Chapter 3

B.1 Proof of Theorem II

In this section we provide the convergence proof for the VA, which basically estimates the R principal network-wide GEVCs in a distributed fashion. Table 3.2 summarizes the different steps of this algorithm. Therefore the objective is to show that in this case the iterative updates of $\underline{\underline{\mathbb{X}}}^i$ (defined in (3.51)) converge to the network-wide GEVCs matrix $\hat{\mathbb{X}}$ (in (3.36)). It is noted that, even when the GEVCs are normalized according to (3.12), the GEVCs still have a sign ambiguity, which is ignored here for the sake of an easy exposition, i.e., we prove convergence up to a sign ambiguity. It is noted that this sign ambiguity can be easily resolved in practice by selecting the proper sign for each column of $\widetilde{\underline{\underline{\mathbb{X}}}}_q^i$ such that $\|\underline{\underline{\mathbb{X}}}_q^{i+1} - \underline{\underline{\mathbb{X}}}_q^i\|$ is minimized in (3.46).

It has been shown in [196] that $\hat{\mathbb{X}}$ is the solution of the following network-wide constrained optimization problem:

$$\hat{\mathbb{X}} = \underset{\mathbb{X}}{\operatorname{argmax}}\, f(\mathbb{X}) \tag{B.1}$$

$$\text{s.t. } \mathbb{X}^H \mathbf{R}_{nn} \mathbb{X} = \mathbf{I}_R \tag{B.2}$$

with $f(\mathbb{X}) = \operatorname{Tr}\{\mathbb{X}^H \mathbf{R}_{yy} \mathbb{X}\}$.

The algorithm in Table 3.2 is derived in [1] by guaranteeing a monotonic increase of the objective function $f(\underline{\underline{\mathbb{X}}}^i)$ under the constraint (B.2), i.e., the following Lemma follows directly from the results in [1]:

Lemma A.I. *$f(\underline{\underline{\mathbb{X}}}^i)$ increases monotonically in each iteration of the VA,*

i.e., $f(\underline{\mathbb{X}}^{i+1}) \geq f(\underline{\mathbb{X}}^{i}), \forall i \in \mathbb{N}_0$. *Furthermore, all matrices* $\underline{\mathbb{X}}^{i}, \forall i \in \mathbb{N}$, *satisfy* $\underline{\mathbb{X}}^{i\,H}\mathbf{R}_{nn}\underline{\mathbb{X}}^{i} = \mathbf{I}_R$.

It is noted that such a monotonic increase implies convergence of $f(\underline{\mathbb{X}}^{i})$, but not necessarily convergence of the argument $\underline{\mathbb{X}}^{i}$. In the sequel, we will formally prove the latter.

In general an equilibrium point $\underline{\mathbb{X}}^{*}$ of the VA is defined as a point where if $\underline{\mathbb{X}}^{i} = \underline{\mathbb{X}}^{*}$ at iteration $i = i^*$, then $\underline{\mathbb{X}}^{j+1} = \underline{\mathbb{X}}^{j}, \forall j \geq i^*$, where the superscript $(.^*)$ denotes an equilibrium condition. An equilibrium point $\underline{\mathbb{X}}^{*}$ is said to be stable if there exist no infinitesimally small perturbations on $\underline{\mathbb{X}}^{*}$ that let the VA algorithm converge away from this point.

The first theorem addresses the equilibrium point(s) of the VA and their stability.
Theorem A.II. *Let* $\mathcal{X}^*$ *denote the set of all equilibrium points of the VA and let* $\underline{\mathbb{X}}^{*} \in \mathcal{X}^*$. *Then* $\underline{\mathbb{X}}^{*}$ *can only contain GEVCs of the ordered matrix pair* $(\mathbf{R}_{yy}, \mathbf{R}_{nn})$ *in its columns. Furthermore,* $\mathcal{X}^*$ *always contains* $\hat{\mathbb{X}}$, *i.e., the matrix containing the R principal GEVCs, and this is the only stable equilibrium point under the VA update rules.*

Proof: We assume (w.l.o.g.) that node q is the updating node at iteration i of the VA. Since $\widetilde{\underline{\mathbb{X}}}_q^{i+1}$ contains the principal GEVCs of $(\underline{\mathbf{R}}^{i}_{\tilde{y}_q\tilde{y}_q}, \underline{\mathbf{R}}^{i}_{\tilde{n}_q\tilde{n}_q})$, we can write (compare with (3.20))

$$\underline{\mathbf{R}}^{i}_{\tilde{y}_q\tilde{y}_q}\widetilde{\underline{\mathbb{X}}}_q^{i+1} = \underline{\mathbf{R}}^{i}_{\tilde{n}_q\tilde{n}_q}\widetilde{\underline{\mathbb{X}}}_q^{i+1}\widetilde{\underline{\mathbb{L}}}_q^{i+1}. \tag{B.3}$$

We define the $M \times P_q$ compression matrix $\mathbf{C}_q^i$ as

$$\mathbf{C}_q^i = \left[\begin{array}{c|c|c} \mathbf{0} & \mathbf{B}^i_{<q} & \mathbf{0} \\ \hline \mathbf{I}_{M_q} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{B}^i_{>q} \end{array}\right] \tag{B.4}$$

where $\mathbf{0}$ is an all-zero matrix of proper dimension, and with

$$\mathbf{B}^i_{<q} = \text{Blkdiag}(\underline{\mathbb{X}}^i_1, \ldots, \underline{\mathbb{X}}^i_{(q-1)}) \tag{B.5}$$

$$\mathbf{B}^i_{>q} = \text{Blkdiag}(\underline{\mathbb{X}}^i_{(q+1)}, \ldots, \underline{\mathbb{X}}^i_K). \tag{B.6}$$

From the definition of $\widetilde{\underline{\mathbf{y}}}_q^i$ and $\widetilde{\underline{\mathbf{n}}}_q^i$, and considering the fact that $\underline{\mathbf{F}}_k^i = \underline{\mathbb{X}}_k^i, \forall k \in \mathcal{K}\backslash q$ (see step 2 in Table 3.2), it can be easily verified that (B.3) can be written as

$$\boxed{\mathbf{C}_q^{i\,H}\mathbf{R}_{yy}\mathbf{C}_q^i}\,\widetilde{\underline{\mathbb{X}}}_q^{i+1} = \boxed{\mathbf{C}_q^{i\,H}\mathbf{R}_{nn}\mathbf{C}_q^i}\,\widetilde{\underline{\mathbb{X}}}_q^{i+1}\widetilde{\underline{\mathbb{L}}}_q^{i+1} \tag{B.7}$$

Now, let us assume that the algorithm has reached an equilibrium point, i.e., $\{\underline{\mathbb{X}}^{i+1} = \underline{\mathbb{X}}^{i}\} \in \mathcal{X}^*$. Since $\underline{\mathbb{X}}_k^{i+1} = \underline{\mathbb{X}}_k^i \underline{\mathbf{G}}_k$ in (3.48), it follows that $\underline{\mathbf{G}}_k = \mathbf{I}_R, \forall k \in \mathcal{K}$ and that

$$\widetilde{\underline{\mathbb{X}}}_q^{i+1} = \left[\underline{\mathbb{X}}_q^{i\,T}\, \mathbf{I}_R \dots \mathbf{I}_R\right]^T. \tag{B.8}$$

Note that also in an equilibrium point GELVs do not change over the iterations, i.e., $\widetilde{\underline{\mathbb{L}}}_q^{i+1} = \widetilde{\underline{\mathbb{L}}}_q^{i}$. Hence in this point we have that

$$\mathbf{C}_q^{i\,H} \mathbf{R}_{yy} \underline{\mathbb{X}}^i = \mathbf{C}_q^{i\,H} \mathbf{R}_{nn} \underline{\mathbb{X}}^i \widetilde{\underline{\mathbb{L}}}_q^i. \tag{B.9}$$

Considering (B.5)-(B.7), with selecting the first M_q rows of (B.9) we have

$$\mathbf{U}_q \mathbf{R}_{yy} \underline{\mathbb{X}}^i = \mathbf{U}_q \mathbf{R}_{nn} \underline{\mathbb{X}}^i \widetilde{\underline{\mathbb{L}}}_q^i \tag{B.10}$$

where $\mathbf{U}_q = \left[\mathbf{0}_{(M_q \times \sum_{j=1}^{q-1} M_j)} \,|\, \mathbf{I}_{M_q} \,|\, \mathbf{0}_{(M_q \times \sum_{j=q+1}^{K} M_j)}\right]$. If an equilibrium point is reached, then the same reasoning can be performed for all the other nodes in the WSN as well. Consequently, by stacking the K matrix equations as defined in (B.10), $\forall q \in \mathcal{K}$, we have

$$\mathbf{R}_{yy} \underline{\mathbb{X}}^i = \begin{bmatrix} \mathbf{U}_1 \mathbf{R}_{nn} \underline{\mathbb{X}}^i \widetilde{\underline{\mathbb{L}}}_1^i \\ \vdots \\ \mathbf{U}_K \mathbf{R}_{nn} \underline{\mathbb{X}}^i \widetilde{\underline{\mathbb{L}}}_K^i \end{bmatrix}. \tag{B.11}$$

By left-multiplying (B.9) with the matrix $\left[\underline{\mathbb{X}}_q^{i\,H}\, \mathbf{I}_R \dots\, \mathbf{I}_R\right]$, and since $\underline{\mathbb{X}}^{i\,H} \mathbf{R}_{nn} \underline{\mathbb{X}}^i = \mathbf{I}_R$ we obtain

$$\underline{\mathbb{X}}^{i\,H} \mathbf{R}_{yy} \underline{\mathbb{X}}^i = \widetilde{\underline{\mathbb{L}}}_q^i. \tag{B.12}$$

Since the lefthand side of (B.12) is independent of q and since it holds for all $q \in \mathcal{K}$, we conclude that $\widetilde{\underline{\mathbb{L}}}_k^i = \widetilde{\underline{\mathbb{L}}}_q^i \triangleq \underline{\mathbb{L}}^i$, $\forall k, q \in \mathcal{K}$, where $\underline{\mathbb{L}}^i$ is an $R \times R$ diagonal matrix. Based on this, we can rewrite (B.11) as

$$\mathbf{R}_{yy} \underline{\mathbb{X}}^i = \mathbf{R}_{nn} \underline{\mathbb{X}}^i \underline{\mathbb{L}}^i \tag{B.13}$$

where $\underline{\mathbb{X}}^{i\,H} \mathbf{R}_{nn} \underline{\mathbb{X}}^i = \mathbf{I}_R$ (see Lemma A.1). Hence, and since we have assumed that $\underline{\mathbb{X}}^i$ is an arbitrary equilibrium point in $\mathcal{X}^*$, we conclude that any equilibrium point can only contain GEVCs of the ordered matrix pair $(\mathbf{R}_{yy}, \mathbf{R}_{nn})$. Note that this is a necessary condition for $\underline{\mathbb{X}}^i$ to be an equilibrium point, but not a sufficient condition.

The fact that $\hat{\mathbb{X}} \in \mathcal{X}^*$ follows straightforwardly from the fact that $\hat{\mathbb{X}}$ maximizes (B.1)-(B.2), and the fact that the VA results in a monotonic increase of $f(\underline{\mathbb{X}}^i)$

(Lemma A.1) under the constraint (B.2). Note that the assumption (3.26) also assures that (B.1)-(B.2) has a unique maximum. Even if this assumption does not hold, i.e. if $\hat{\mathbb{X}}$ is not unique, the fix in Appendix B will ensure that $\hat{\mathbb{X}} \in \mathcal{X}^*$, i.e., $\hat{\mathbb{X}}$ does not change under the VA updates.

Finally we must show that $\hat{\mathbb{X}}$ is the only stable equilibrium point. Essentially, an equilibrium point $\underline{\mathbb{X}}^*$ is stable under the VA update rules if any small perturbation of $\underline{\mathbb{X}}^*$ does not lead to an increase of the objective function $f(\underline{\mathbb{X}}^*)$, i.e.,

$$\exists\, \zeta > 0,\ \forall \Delta\boldsymbol{\Upsilon} \in \Upsilon^{M\times R} : \|\Delta\boldsymbol{\Upsilon}\|_F \leq \zeta \\ \Rightarrow f(\underline{\mathbb{X}}^* + \Delta\boldsymbol{\Upsilon}) \leq f(\underline{\mathbb{X}}^*) \tag{B.14}$$

where $\Upsilon^{M\times R}$ is the set containing all possible perturbations such that $\underline{\mathbb{X}}^H \mathbf{R}_{nn} \underline{\mathbb{X}} = \mathbf{I}_R$ is not violated when $\underline{\mathbb{X}} = \underline{\mathbb{X}}^* + \Delta\boldsymbol{\Upsilon}$. Indeed, this condition follows from the fact that f is monotonically increasing under the VA updates (Lemma A.1), and hence equilibrium points $\underline{\mathbb{X}}^*$ that do not satisfy the condition (B.14) are unstable under the VA update rules since a small perturbation may lead to $f(\underline{\mathbb{X}}^* + \Delta\mathbf{U}) \geq f(\underline{\mathbb{X}}^*)$. In this case, considering that $f(\underline{\mathbb{X}}^{i+1}) \geq f(\underline{\mathbb{X}}^i)$, $\forall i \in \mathbb{N}$, the VA cannot return to the original equilibrium point $\underline{\mathbb{X}}^*$. If $\underline{\mathbb{X}}^*$ contains GEVCs that are not in the columns of $\hat{\mathbb{X}}$, it does not satisfy (B.14). Therefore, $\hat{\mathbb{X}}$ is the only point that both satisfies the necessary condition for equilibria (B.13) and the stability condition (B.14), and hence it is the only stable equilibrium point in $\mathcal{X}^*$.

□

Theorem A.III. *For any initialization of the VA,* $\lim_{i\to\infty} \underline{\mathbb{X}}^i$ *exists, i.e., the VA converges. Furthermore the R principal local GEVLs converges, i.e.,* $lim_{i\to\infty}\, \underline{\widetilde{\mathbb{L}}}_k^i = \hat{\mathbb{L}}$, $\forall k \in \mathcal{K}$.

Proof: The first step to prove this theorem is to show that, as $i \to \infty$, the distance between two consecutive updates of the VA vanishes[1], i.e., $\lim\limits_{i\to\infty} \|\underline{\mathbb{X}}^{i+1} - \underline{\mathbb{X}}^i\|_F = 0$. To show this, consider the fact that since $f(\underline{\mathbb{X}}^i)$ increases monotonically (Lemma A.1), and since it has an upper bound, we have that

$$\lim_{i\to\infty} \left(f(\underline{\mathbb{X}}^{i+1}) - f(\underline{\mathbb{X}}^i)\right) = 0. \tag{B.15}$$

In the VA, $\underline{\widetilde{\mathbb{X}}}_q^{i+1}$ contains the R principal GEVCs of $(\underline{\mathbf{R}}^i_{\tilde{y}_q\tilde{y}_q}, \underline{\mathbf{R}}^i_{\tilde{n}_q\tilde{n}_q})$ (computed based on a GEVD at the updating node q at iteration i), and we know that it maximizes $f(\underline{\mathbb{X}}^i)$ within a constraint set $\mathcal{C}$, which is dictated by the parameterization of $\underline{\mathbb{X}}^i$ in the VA (similar to (3.33) for $\mathbf{W}_k^i$, see [1] for more

[1] It is noted that this is again a necessary condition for convergence, but not sufficient.

details). Due to assumption (3.26), we know that this maximum is unique, and together with the continuity of the objective function f within $\mathcal{C}$, it follows that (assuming the sign ambiguity is resolved)

$$\exists\, \zeta > 0, \exists\, \mu > 0, \forall\, \underline{\underline{\mathbb{X}}} \in \mathcal{C} : \, |f(\underline{\underline{\mathbb{X}}}^{i+1}) - f(\underline{\underline{\mathbb{X}}})| < \mu$$

$$\Rightarrow \|\underline{\underline{\mathbb{X}}}^{i+1} - \underline{\underline{\mathbb{X}}}\|_F < \zeta. \tag{B.16}$$

With (B.15) and (B.16) we can conclude that

$$\lim_{i\to\infty} \|\underline{\underline{\mathbb{X}}}^{i+1} - \underline{\underline{\mathbb{X}}}^{i}\|_F = 0. \tag{B.17}$$

The proof of Theorem A.I relied on the fact that $\{\underline{\underline{\mathbb{X}}}^{i+1} = \underline{\underline{\mathbb{X}}}^{i}\} \in \mathcal{X}^*$, which is used to obtain (B.9) from (B.7). However, if $\underline{\underline{\mathbb{X}}}^{i} \notin \mathcal{X}^*$, then $\underline{\underline{\mathbb{X}}}^{i+1} \neq \underline{\underline{\mathbb{X}}}^{i}$ and therefore an error term $\mathbf{\Theta}_q^i$ should be added in (B.9), i.e.,

$$\mathbf{C}_q^{i\,H}\mathbf{R}_{yy}\underline{\underline{\mathbb{X}}}^{i} = \mathbf{C}_q^{i\,H}\mathbf{R}_{nn}\underline{\underline{\mathbb{X}}}^{i}\underline{\underline{\widetilde{\mathbb{L}}}}_q^{i} + \mathbf{\Theta}_q^i. \tag{B.18}$$

An error term then also appears in (B.11) and (B.12), which are derived from (B.9), i.e.,

$$\mathbf{R}_{yy}\underline{\underline{\mathbb{X}}}^{i} + \mathbf{\Omega}^i = \begin{bmatrix} \mathbf{U}_1\mathbf{R}_{nn}\underline{\underline{\mathbb{X}}}^{i}\,\underline{\underline{\widetilde{\mathbb{L}}}}_1^{i} \\ \vdots \\ \mathbf{U}_K\mathbf{R}_{nn}\underline{\underline{\mathbb{X}}}^{i}\,\underline{\underline{\widetilde{\mathbb{L}}}}_K^{i} \end{bmatrix} \tag{B.19}$$

$$\underline{\underline{\mathbb{X}}}^{i\,H}\mathbf{R}_{yy}\underline{\underline{\mathbb{X}}}^{i} + \mathbf{\Omega}_q^i = \underline{\underline{\widetilde{\mathbb{L}}}}_q^{i} \tag{B.20}$$

where $\mathbf{\Omega}^i$ and $\mathbf{\Omega}_q^i$, $\forall q \in \mathcal{K}$ are error terms. However, from (B.17), it follows that the error term $\mathbf{\Theta}_q^i$ vanishes in (B.18) if $i \to \infty$, and therefore also the error terms $\mathbf{\Omega}^i$ and $\mathbf{\Omega}_q^i$ vanish, i.e., from (B.20) we have

$$\lim_{i\to\infty} \|\underline{\underline{\mathbb{X}}}^{i\,H}\mathbf{R}_{yy}\underline{\underline{\mathbb{X}}}^{i} - \underline{\underline{\widetilde{\mathbb{L}}}}_k^{i}\|_F = 0 \quad \forall k \in \mathcal{K}. \tag{B.21}$$

When defining

$$\underline{\underline{\mathbb{L}}}^{i} \triangleq \underline{\underline{\mathbb{X}}}^{i\,H}\mathbf{R}_{yy}\underline{\underline{\mathbb{X}}}^{i} \tag{B.22}$$

and using (B.21), we find from (B.19) that

$$\lim_{i\to\infty} \|\mathbf{R}_{yy}\underline{\underline{\mathbb{X}}}^{i} - \mathbf{R}_{nn}\underline{\underline{\mathbb{X}}}^{i}\underline{\underline{\mathbb{L}}}^{i}\|_F = 0. \tag{B.23}$$

Since $\underline{\underline{\widetilde{\mathbb{L}}}}_k^{i}$, $\forall k \in \mathcal{K}$, are diagonal matrices (by construction), it follows from (B.21) and (B.22) that $\lim_{i\to\infty} \underline{\underline{\mathbb{L}}}^{i}$ is also diagonal. This fact together with (B.23), shows

that $\underline{\mathbb{X}}^i$ converges to the GEVCs of $\mathbf{R}_{yy}$ and $\mathbf{R}_{nn}$ when $i \to \infty$. Furthermore, it follows from (B.17) that the columns of $\underline{\mathbb{X}}^i$ cannot switch between different GEVCs when $i \to \infty$. This also implies that the corresponding R principal GEVLs converge, and hence together with (B.13), we can conclude that $\lim_{i\to\infty} \underline{\widetilde{\mathbb{L}}}_q^i = \hat{\mathbb{L}}$, $\forall q \in \mathcal{K}$, which proves the theorem. □

We can safely assume that in practice because of estimation errors and/or numerical noise, the VA algorithm will diverge away from other possible equilibrium points. Since $\hat{\mathbb{X}}$ is the only stable equilibrium point (Theorem A.II), and based on Theorem A.III, we conclude that the VA algorithm converges to the matrix $\hat{\mathbb{X}}$ containing the network-wide GEVCs of $(\mathbf{R}_{yy}, \mathbf{R}_{nn})$.

B.2 Algorithm fixes for special cases

Rank deficient $\underline{\mathbf{R}}^i_{\tilde{y}_q\tilde{y}_q}$ and/or $\underline{\mathbf{R}}^i_{\tilde{n}_q\tilde{n}_q}$

In the rare case where $\underline{\mathbf{R}}^i_{\tilde{y}_q\tilde{y}_q}$ and/or $\underline{\mathbf{R}}^i_{\tilde{n}_q\tilde{n}_q}$ is rank deficient[2], then the local GEVD solution at node q in iteration i is ill-defined. Basically, rank deficiency of these matrices occurs when there is a node k for which $\underline{\mathbb{X}}^i_k$ has linearly dependent columns. Assuming w.l.o.g. that the rank deficiency occurs in $\underline{\mathbf{R}}^i_{\tilde{y}_q\tilde{y}_q}$, then there are two ways to circumvent this problem:

- The linearly dependent column in $\underline{\mathbb{X}}^i_k$, and its corresponding entries in $\underline{\mathbf{z}}^i_k$ is removed when constructing $\underline{\mathbf{R}}^i_{\tilde{y}_q\tilde{y}_q}$ at the updating node q. As the removed signal observations can be reconstructed from the other signal observations in $\underline{\mathbf{z}}^i_k$ (Step 2 in Table 3.1), this removal cannot counteract the monotonic increase of $f(\underline{\mathbb{X}}^i_k)$ in the VA.
- Node k replaces the linearly dependent column in $\underline{\mathbb{X}}^i_k$ by random entries, yielding a new $\underline{\mathbf{z}}^i_k$ in which all channels are linearly independent. Note that this replacement provides the other nodes with additional information, and therefore this is a better option compared to the first one.

Degenerate principal GEVLs

In the rare case where the n-th largest GEVL of $(\underline{\mathbf{R}}^i_{\tilde{y}_q\tilde{y}_q}, \underline{\mathbf{R}}^i_{\tilde{n}_q\tilde{n}_q})$ is degenerate, i.e., $\tilde{\lambda}^i_n = \tilde{\lambda}^i_{n+1}$ (with $n \leq R$), $\underline{\widetilde{\mathbb{X}}}^{i+1}_q$ is ill-defined in its n-th and $(n+1)$-th

[2]Although these fixes only consider $\underline{\mathbf{R}}^i_{\tilde{y}_q\tilde{y}_q}$ and/or $\underline{\mathbf{R}}^i_{\tilde{n}_q\tilde{n}_q}$ in the VA, note that they are equally applicable for $\mathbf{R}^i_{\tilde{y}_q\tilde{y}_q}$ and/or $\mathbf{R}^i_{\tilde{n}_q\tilde{n}_q}$ in the GEVD-based DANSE algorithm.

column. One possible fix is to skip node q in the current update round, assuming that the problem will not occur in the next update round. In case this simple fix does not solve the problem, we also propose an alternative fix. For the sake of an easy exposition and w.l.o.g., we consider the worst case, i.e., we assume that all R columns are ill-defined, i.e., $\tilde{\lambda}_1^i = \tilde{\lambda}_2^i = \ldots = \tilde{\lambda}_{R+1}^i$. In this case, the GEVCs corresponding to these GEVLs span an $(R+1)$-dimensional subspace $\mathcal{S}$. A unique $\widetilde{\underline{\mathbb{X}}}_q^{i+1}$ can then be defined as

$$\widetilde{\underline{\mathbb{X}}}_q^{i+1} = \underset{\widetilde{\underline{\mathbb{X}}}_q}{\operatorname{argmin}} \|\widetilde{\underline{\mathbb{X}}}_q - \check{\mathbf{X}}_q\|_F \tag{B.24}$$

$$\text{s.t. } \mathit{Range}\left(\widetilde{\underline{\mathbb{X}}}_q\right) = \mathcal{S} \text{ and } \widetilde{\underline{\mathbb{X}}}_q^T \underline{\mathbf{R}}_{\tilde{n}_q \tilde{n}_q}^i \widetilde{\underline{\mathbb{X}}}_q = \mathbf{I}_R \tag{B.25}$$

where $\check{\mathbf{X}}_q = \left[\underline{\mathbb{X}}_q^i \ \mathbf{I}_R \ \ldots \mathbf{I}_R\right]$. Note that this fix also ensures convergence, since $\widetilde{\underline{\mathbb{X}}}_q^{i+1} = \check{\mathbf{X}}_q$ implies that $\underline{\mathbb{X}}^{i+1} = \underline{\mathbb{X}}^i$ (this can be seen from (3.46)-(3.48)).

Appendix C

APPENDIX to Chapter 6

C.1 Distributed GEVD algorithm

If all nodes $k \in \mathcal{K}$ would merely aim at obtaining the S principal network-wide GEVCs $\hat{\mathbb{X}}$ in a distributed fashion, the Distributed Adaptive Covariance-matrix Generalized Eigenvector Estimation (DACGEE) algorithm from [1] could instead be applied. Note that unlike the distributed MDMT algorithm where nodes undertake different *node-specific* SP tasks, the DACGEE algorithm computes the network-wide GEVCs $\hat{\mathbb{X}}$ in a distributed fashion, without any node-specific aspect.

A similar set of notations as in the distributed MDMT algorithm (Table 6.1) will be used in the DACGEE algorithm, where an underline will be used to make a distinction, leading to $\underline{\mathbf{R}}^i_{\tilde{y}_k\tilde{y}_k}$, $\underline{\mathbf{R}}^i_{\tilde{n}_k\tilde{n}_k}$, $\underline{\widetilde{\mathbf{y}}}^i_k$, $\underline{\widetilde{\mathbf{X}}}^i_k$, $\underline{\widetilde{\mathbf{L}}}^i_k$, $\underline{\mathbb{X}}^i_k$, $\underline{\widetilde{\mathbf{Q}}}^i_k$, $\underline{\mathbb{Q}}^i$, $\underline{\mathbb{Q}}^i_k$, $\underline{\mathbf{F}}^i_k$ and $\underline{\mathbf{z}}^i_k$. The reason of introducing such a distinction is due to the fact that in the DACGEE algorithm the nodes use different compression matrices $\underline{\mathbf{F}}_k$, leading first to different signal vectors $\underline{\widetilde{\mathbf{y}}}^i_k = [\mathbf{y}^T_k \ \underline{\mathbf{z}}^{i\,T}_{-k}]^T$ and then to different correlation matrices $(\underline{\mathbf{R}}^i_{\tilde{y}_k\tilde{y}_k}, \underline{\mathbf{R}}^i_{\tilde{n}_k\tilde{n}_k})$ and so on. Table C.1 summarized the DACGEE algorithm in a fully-connected WSN. For details of the algorithm we refer to [1].

Table C.1: The DACGEE algorithm [1]

1. Set $i \leftarrow 0$, $q \leftarrow 1$, and initialize all $\underline{\mathbf{F}}_k^0$ and $\underline{\mathbb{X}}_k^0$, $\forall\, k \in \mathcal{K}$, with random entries.
2. Each node $k \in \mathcal{K}$ broadcasts N new S-channel compressed sensor signal $\underline{\mathbf{z}}_k^i[iN+j] = \underline{\mathbf{F}}_k^{i\,H}\mathbf{y}_k[iN+j], \quad j = 1 \ldots N$.
3. Each node $k \in \mathcal{K}$ first updates $\underline{\mathbf{R}}_{\tilde{y}_k\tilde{y}_k}^i$ and $\underline{\mathbf{R}}_{\tilde{n}_k\tilde{n}_k}^i$ and then compute $\widetilde{\underline{\mathbb{X}}}_k^{i+1}$ from the GEVD of $(\underline{\mathbf{R}}_{\tilde{y}_k\tilde{y}_k}^i, \underline{\mathbf{R}}_{\tilde{n}_k\tilde{n}_k}^i)$, normalized such that $\big(\widetilde{\underline{\mathbb{X}}}_k^{i+1}\big)^H \underline{\mathbf{R}}_{\tilde{n}_k\tilde{n}_k}^i \widetilde{\underline{\mathbb{X}}}_k^{i+1} = \mathbf{I}_S$. Then it partitions $\widetilde{\underline{\mathbb{X}}}_k^{i+1}$ as
$$\underline{\mathbb{X}}_k^{i+1} = \big[\mathbf{I}_{M_k}\ \mathbf{O}\big]\, \widetilde{\underline{\mathbb{X}}}_k^{i+1} \tag{C.2}$$
$$\underline{\mathbf{G}}_{-k} = \big[\mathbf{O}\ \mathbf{I}_{S(K-1)}\big]\, \widetilde{\underline{\mathbb{X}}}_k^{i+1} \tag{C.3}$$
4. **Updating node** q: updates its $\underline{\mathbf{F}}_q^{i+1} = \underline{\mathbb{X}}_q^{i+1}$ and broadcast $\underline{\mathbf{G}}_{-q} = \Big[\underline{\mathbf{G}}_1^T \ \cdots\ \underline{\mathbf{G}}_{(q-1)}^T\, \underline{\mathbf{G}}_{(q+1)}^T \ \cdots\ \underline{\mathbf{G}}_K^T\Big]^T$ to all the other nodes.
5. Each node $k \in \mathcal{K}\backslash\{q\}$ updates $\underline{\mathbb{X}}_k^{i+1} = \underline{\mathbb{X}}_k^i \underline{\mathbf{G}}_k$.
6. $i \leftarrow i+1$ and $q \leftarrow (q \mod K) + 1$ and return to step 2.

For future purposes, we further introduce the concatenated $M \times S$ matrix $\underline{\mathbb{X}}^i$ in the DACGEE algorithm and also define the partitioning of $\hat{\mathbb{X}}$ as follows

$$\underline{\mathbb{X}}^i \triangleq \begin{bmatrix} \underline{\mathbb{X}}_1^i \\ \vdots \\ \underline{\mathbb{X}}_K^i \end{bmatrix} \qquad \hat{\mathbb{X}} \triangleq \begin{bmatrix} \hat{\mathbb{X}}_1 \\ \vdots \\ \hat{\mathbb{X}}_K \end{bmatrix} \tag{C.1}$$

with $\hat{\mathbb{X}}_k$ denoting the $M_k \times R$ submatrix of $\hat{\mathbb{X}}$ corresponding to node k. In the DACGEE algorithm, $\lim_{i\to\infty} \underline{\mathbb{X}}^i = \hat{\mathbb{X}}$ (see Result C-1 in Appendix C.2.)

C.2 Proof of Theorem I

It will be shown that the convergence of the distributed MDMT algorithm in Table 6.1 follows from the convergence of the DACGEE algorithm in Table C.1, which was proven in [123]:

Table C.2: $\mathbf{G}_{kn}$ transformations after convergence of the MDMT algorithm

node k task	$\forall n \in \mathcal{K}^{\text{MWF}} \setminus k$	$\forall n \in \mathcal{K}^{\text{LCMV}} \setminus k$	$\forall n \in \mathcal{K}^{\text{DOA}} \setminus k$
$k \in \mathcal{K}^{\text{MWF}}$	$\mathbf{G}_{kn}^{\infty} = (\widetilde{\mathbf{\Phi}}_n^{\infty})^{-1}\hat{\mathbf{\Psi}}_k$	$\mathbf{G}_{kn}^{\infty} = (\widetilde{\mathbf{\Theta}}_n^{\infty})^{-1}\hat{\mathbf{\Psi}}_k$	$\mathbf{G}_{kn}^{\infty} = \hat{\mathbf{\Psi}}_k$
$k \in \mathcal{K}^{\text{LCMV}}$	$\mathbf{G}_{kn}^{\infty} = (\widetilde{\mathbf{\Phi}}_n^{\infty})^{-1}\hat{\mathbf{\Theta}}_k$	$\mathbf{G}_{kn}^{\infty} = (\widetilde{\mathbf{\Theta}}_n^{\infty})^{-1}\hat{\mathbf{\Theta}}_k$	$\mathbf{G}_{kn}^{\infty} = \hat{\mathbf{\Theta}}_k$
$k \in \mathcal{K}^{\text{DOA}}$	$\mathbf{G}_{kn}^{\infty} = (\widetilde{\mathbf{\Phi}}_n^{\infty})^{-1}$	$\mathbf{G}_{kn}^{\infty} = (\widetilde{\mathbf{\Theta}}_n^{\infty})^{-1}$	$\mathbf{G}_{kn}^{\infty} = \mathbf{I}_{M_n}$

- **Result C-1**: with the DACGEE algorithm, the concatenated matrix $\underline{\mathbb{X}}^i$ (see (C.1)) converges to the matrix $\hat{\mathbb{X}}$ containing the S principal network-wide GEVCs of $(\mathbf{R}_{yy}, \mathbf{R}_{nn})$, i.e., $\lim_{i\to\infty} \underline{\mathbb{X}}^i = \hat{\mathbb{X}}$. Moreover $\lim_{i\to\infty} \widetilde{\underline{\mathbb{L}}}_k^i = \hat{\mathbb{L}}$, $\forall k \in \mathcal{K}$.

  We will not replicate the convergence proof of the DACGEE algorithm here, but instead we will only focus on the key ingredients that allow to prove that the distributed MDMT algorithm of Table 6.1 inherits the convergence Result C-1 of the DACGEE algorithm of Table C.1. To this end, we first adopt the following results from [123].

- **Result C-2**: with the DACGEE algorithm, the matrix $\underline{\mathbb{Q}}_k^i = [\mathbf{I}_{M_k}\ \mathbf{0}]\widetilde{\underline{\mathbf{Q}}}_k^i[\mathbf{I}_{M_k}\ \mathbf{0}]^T$ converges to the corresponding $\hat{\mathbb{Q}}_k$, i.e., $\lim_{i\to\infty} \underline{\mathbb{Q}}_k^i = \hat{\mathbb{Q}}_k, \forall k \in \mathcal{K}$.
  Proof: see [120].

- **Result C-3**: After convergence of the DACGEE algorithm, we have that $\lim_{i\to\infty} \widetilde{\underline{\mathbb{X}}}_k^i = \left[\hat{\mathbb{X}}_k^T\ \mathbf{I}_R \dots \mathbf{I}_R\right]^T, \forall k \in \mathcal{K}$.
  Proof: see [123].

- **Result C-4**: Let $\mathbf{C}$ and $\mathbf{D}$ denote the $m \times m$ matrix containing GEVCs and GEVLs of the matrix pair $(\mathbf{A}, \mathbf{B}) \in \mathbb{C}^{m\times m}$, respectively, where $\mathbf{A}$ and $\mathbf{B}$ are full-rank matrices and where GEVCs in $\mathbf{C}$ are scaled such that $\mathbf{C}^H\mathbf{B}\mathbf{C} = \mathbf{I}_m$. With any invertible matrix $\mathbf{J} \in \mathbb{C}^{m\times m}$, the GEVCs and GEVLs of the matrix pair $(\mathbf{J}\mathbf{A}\mathbf{J}^H, \mathbf{J}\mathbf{B}\mathbf{J}^H)$ become $\mathbf{J}^{-H}\mathbf{C}$ and $\mathbf{D}$, respectively.
  Proof: see [123].

- **Result C-5**: When at iteration i of the DACGEE algorithm, a node-specific $S \times S$ column transformation is applied to the compression matrices $\underline{\mathbf{F}}_n^i$ of all nodes $n \in \mathcal{K} \setminus k$, then at node k in the *next* iteration, the first M_k rows of $\widetilde{\underline{\mathbb{X}}}_k^{i+1}$, i.e., $\underline{\mathbb{X}}_k^{i+1}$, and $\widetilde{\underline{\mathbb{L}}}_k^{i+1}$ remain *unchanged*.
  Proof: see [123].

For all $k \in \mathcal{K}$, the first step in the distributed MDMT algorithm is always the computation of a GEVD of the matrix pencil $(\mathbf{R}_{\tilde{y}_k\tilde{y}_k}^i, \mathbf{R}_{\tilde{n}_k\tilde{n}_k}^i)$, as it is also done

in the DACGEE algorithm. Furthermore, it is known from Result C-5 that an $S \times S$ column transformation on the compression matrix $\underline{\mathbf{F}}_k^i$ has no influence on the update of $\widetilde{\mathbb{X}}_k^i$. Furthermore, comparing the compression matrices $\mathbf{F}_k^i$ of the distributed MDMT algorithm in (6.61a)-(6.61c), it can be seen that these are equal to the DACGEE compression matrices $\underline{\mathbf{F}}_k^i$ up to the $S \times S$ column transformations with $\widetilde{\boldsymbol{\Phi}}_k$ at nodes $k \in \mathcal{K}^{\mathrm{MWF}}$ and $\widetilde{\boldsymbol{\Theta}}_k^i$ at nodes $k \in \mathcal{K}^{\mathrm{LCMV}}$. Based on Result C-5, it can then be concluded that (assuming both algorithms are initialized with the same values)

$$\underline{\mathbb{X}}_k^i = \mathbb{X}_k^i, \quad \forall i \in \mathbb{N} \tag{C.4}$$

$$\underline{\widetilde{\mathbb{L}}}_k^i = \widetilde{\mathbb{L}}_k^i, \quad \forall i \in \mathbb{N} \tag{C.5}$$

This relationship is indeed of great importance, since it is used to link the correlation matrices of the DACGEE algorithm and the distributed MDMT algorithm at node k as follows:

$$\mathbf{R}_{\tilde{y}_k \tilde{y}_k}^i = \widetilde{\mathbf{J}}_k^i \, \underline{\mathbf{R}}_{\tilde{y}_k \tilde{y}_k}^i \, \widetilde{\mathbf{J}}_k^{iH} \tag{C.6}$$

$$\mathbf{R}_{\tilde{n}_k \tilde{n}_k}^i = \widetilde{\mathbf{J}}_k^i \, \underline{\mathbf{R}}_{\tilde{n}_k \tilde{n}_k}^i \, \widetilde{\mathbf{J}}_k^{iH} \tag{C.7}$$

where the transformation matrix $\widetilde{\mathbf{J}}_k^i$ is

$$\widetilde{\mathbf{J}}_k^i = \begin{bmatrix} \mathbf{I}_{M_k} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{B}_{<k}^i & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{B}_{>k}^i \end{bmatrix} \tag{C.8}$$

with

$$\mathbf{B}_{<k}^i \triangleq \mathrm{Blkdiag}(\mathbf{B}_1^i, \ldots, \mathbf{B}_{(k-1)}^i) \tag{C.9}$$

$$\mathbf{B}_{>k}^i \triangleq \mathrm{Blkdiag}(\mathbf{B}_{(k+1)}^i, \ldots, \mathbf{B}_K^i) \tag{C.10}$$

where Blkdiag(.) creates a block-diagonal matrix with the matrices in its argument on the block diagonal, and where $\mathbf{B}_n^i$ is equal to $(\widetilde{\boldsymbol{\Phi}}_n^i)^H$, $(\widetilde{\boldsymbol{\Theta}}_n^i)^H$, or $\mathbf{I}_S$, depending on whether node n is an MWF, LCMV or DOA node, respectively. Now Result C-4 can be used to link the GEVCs at node k such that

$$\widetilde{\mathbf{X}}_k^i = (\widetilde{\mathbf{J}}_k^i)^{-H} \underline{\widetilde{\mathbf{X}}}_k^i \quad \Rightarrow \quad \widetilde{\mathbb{X}}_k^i = (\widetilde{\mathbf{J}}_k^i)^{-H} \underline{\widetilde{\mathbb{X}}}_k^i. \tag{C.11}$$

From this and considering the fact that $\widetilde{\mathbf{X}}_k^i = (\widetilde{\mathbf{Q}}_k^i)^{-H}$ and $\underline{\widetilde{\mathbf{X}}}_k^i = (\underline{\widetilde{\mathbf{Q}}}_k^i)^{-H}$ we can further write

$$\widetilde{\mathbf{Q}}_k^i = \widetilde{\mathbf{J}}_k^i \underline{\widetilde{\mathbf{Q}}}_k^i \quad \Rightarrow \quad \widetilde{\mathbb{Q}}_k^i = \widetilde{\mathbf{J}}_k^i \underline{\widetilde{\mathbb{Q}}}_k^i \tag{C.12}$$

from which it follows that $\mathbb{Q}_k^i = \underline{\mathbb{Q}}_k^i$. With this, and from Result C-2, we can then conclude that the first M_k rows of $\widetilde{\mathbb{Q}}_k^i$ converge, i.e.,

$$\lim_{i\to\infty} \mathbb{Q}_k^i = \hat{\mathbb{Q}}_k, \quad \forall k \in \mathcal{K} \tag{C.13}$$

Substituting (C.11) in local solutions (6.55), (6.59) and (6.43) results in

$$\widetilde{\mathbf{W}}_k^{i+1} = \begin{cases} \left(\widetilde{\mathbf{J}}_k^i\right)^{-H} \underline{\widetilde{\mathbb{X}}}_k^i \widetilde{\mathbf{\Psi}}_k^i & \text{if } k \in \mathcal{K}^{\mathrm{MWF}} \quad \text{(C.14a)} \\ \left(\widetilde{\mathbf{J}}_k^i\right)^{-H} \underline{\widetilde{\mathbb{X}}}_k^i \widetilde{\mathbf{\Theta}}_k^i & \text{if } k \in \mathcal{K}^{\mathrm{LCMV}} \quad \text{(C.14b)} \\ \left(\widetilde{\mathbf{J}}_k^i\right)^{-H} \underline{\widetilde{\mathbb{X}}}_k^i & \text{if } k \in \mathcal{K}^{\mathrm{DOA}} \quad \text{(C.14c)} \end{cases}$$

of which it is known, based on Result C-1, that $\underline{\widetilde{\mathbb{X}}}_k^i, \forall k \in \mathcal{K}$ converges. Thus, the next step is to verify that $\widetilde{\mathbf{\Psi}}_k^i$ and $\widetilde{\mathbf{\Theta}}_k^i$ at all nodes $k \in \mathcal{K}^{\mathrm{MWF}}$ and $k \in \mathcal{K}^{\mathrm{LCMV}}$, respectively, also converge.

- Convergence of $\widetilde{\mathbf{\Psi}}_k^i$ at all nodes $k \in \mathcal{K}^{\mathrm{MWF}}$: Based on the definition of $\widetilde{\mathbf{\Psi}}_k^i$ in (6.56), this convergence is obtained when both $\widetilde{\mathbb{L}}_k^i$ and $\mathbb{Q}_k^{iH}$ converge. Note that, from (C.5) and Result C-1, it follows that $\lim_{i\to\infty} \widetilde{\mathbb{L}}_k^i = \hat{\mathbb{L}}$, $\forall k \in \mathcal{K}$. With this, and considering the convergence of $\mathbb{Q}_k^i$ in (C.13), it is concluded that

$$\widetilde{\mathbf{\Psi}}_k^\infty \triangleq \lim_{i\to\infty} \widetilde{\mathbf{\Psi}}_k^i = \hat{\mathbf{\Psi}}_k, \quad \forall k \in \mathcal{K}^{\mathrm{MWF}} \tag{C.15}$$

  which indeed also results in convergence of $\widetilde{\mathbf{\Phi}}_k^i$ at all nodes $k \in \mathcal{K}^{\mathrm{MWF}}$, i.e., $\widetilde{\mathbf{\Phi}}_k^\infty \triangleq \hat{\mathbf{\Phi}}_k, \forall k \in \mathcal{K}^{\mathrm{MWF}}$.

- Convergence of $\widetilde{\mathbf{\Theta}}_k^i$ at all nodes $k \in \mathcal{K}^{\mathrm{LCMV}}$: Based on the definition (6.60), this convergence is obtained when the first S rows and S columns of $\widetilde{\mathbb{Q}}_k^{iH}$ converge. Since $S < M_k$, this readily follows from (C.13), and hence

$$\widetilde{\mathbf{\Theta}}_k^\infty \triangleq \lim_{i\to\infty} \widetilde{\mathbf{\Theta}}_k^i = \hat{\mathbf{\Theta}}_k, \quad \forall k \in \mathcal{K}^{\mathrm{LCMV}}. \tag{C.16}$$

Now from Result C-3, after convergence of the DACGEE algorithm we have that $\lim_{i\to\infty} \underline{\widetilde{\mathbb{X}}}_k^i = \left[\hat{\mathbb{X}}_k^T \mathbf{I}_R \dots \mathbf{I}_R\right]^T, \forall k \in \mathcal{K}$. Considering this and using (C.15), after convergence of the DACGEE algorithm, the local estimator at each node

$k \in \mathcal{K}^{\text{MWF}}$ becomes (see (C.14a))

$$\lim_{i\to\infty} \widetilde{\mathbf{W}}_k^i = \left[\begin{array}{c} \hat{\mathbb{X}}_k \\ \hline (\mathbf{B}_1^{\infty})^{-H} \\ \vdots \\ (\mathbf{B}_{(k-1)}^{\infty})^{-H} \\ (\mathbf{B}_{(k+1)}^{\infty})^{-H} \\ \vdots \\ (\mathbf{B}_K^{\infty})^{-H} \end{array}\right] \hat{\mathbf{\Psi}}_k, \quad \forall k \in \mathcal{K}^{\text{MWF}}. \tag{C.17}$$

Now, comparing (C.17) with (6.47) implies that $\lim_{i\to\infty} \mathbf{G}_{kn}^i = (\mathbf{B}_n^{\infty})^{-H}\hat{\mathbf{\Psi}}_k, \forall k \in \mathcal{K}^{\text{MWF}}, \forall n \in \mathcal{K}\backslash k$. Similarly, $\lim_{i\to\infty} \mathbf{G}_{kn}^i$ can be expressed in the other nodes $k \in \mathcal{K}^{\text{LCMV}}$ and $k \in \mathcal{K}^{\text{DOA}}$, leading to the results shown in Table C.2. Plugging these results in the parameterization (6.48) eventually gives

$$\lim_{i\to\infty} \mathbf{W}_k^i = \begin{cases} \hat{\mathbb{X}}\hat{\mathbf{\Psi}}_k & \forall k \in \mathcal{K}^{\text{MWF}} \quad \text{(C.18a)} \\ \hat{\mathbb{X}}\hat{\mathbf{\Theta}}_k & \forall k \in \mathcal{K}^{\text{LCMV}} \quad \text{(C.18b)} \\ \hat{\mathbb{X}} & \forall k \in \mathcal{K}^{\text{DOA}} \quad \text{(C.18c)} \end{cases}$$

which verifies that after the convergence of all nodes, each node obtains its corresponding optimal network-wide centralized estimate, i.e., when $i \to \infty$, we have that $\mathbf{W}_k^i = \hat{\mathbf{W}}_k, \forall k \in \mathcal{K}$ and hence $\widetilde{\mathbf{s}}_k^i = \hat{\mathbf{s}}_k, \forall k \in \mathcal{K}^{\text{MWF}}$, and $\widetilde{d}_k^i = \hat{d}_k, \forall k \in \mathcal{K}^{\text{LCMV}}$, and $\widetilde{\theta}_{k_s}^i = \hat{\theta}_{k_s}, \forall k \in \mathcal{K}^{\text{DOA}}$. This proves Theorem I. □

Bibliography

[1] A. Bertrand and M. Moonen, "Distributed adaptive generalized eigenvector estimation of a sensor signal covariance matrix pair in a fully-connected sensor network," *Signal Processing*, vol. 106, pp. 209–214, Jan. 2015.

[2] M. Brandstein and D. Ward, *Microphone Arrays: Signal Processing Techniques and Applications*. Digital Signal Processing - Springer-Verlag, Springer, 2001.

[3] S. Pillai and C. Burrus, *Array signal Processing*. Signal Processing and Digital Filtering, Springer-Verlag, 1989.

[4] 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, 1996.

[5] W. Koenig, "Subjective effects in binaural hearing," *The Journal of the Acoustical Society of America*, vol. 22, pp. 61–62, jan 1950.

[6] N. W. MacKeith and R. R. A. Coles, "Binaural advantages in hearing of speech," *The Journal of Laryngology and Otology*, vol. 85, pp. 213–232, Mar. 1971.

[7] A. Markides, *Binaural Hearing Aids*. Academic Press Inc, 1977.

[8] R. H. Gifford and L. J. Revit, "Speech perception for adult cochlear implant recipients in a realistic background noise: Effectiveness of preprocessing strategies and external options for improving speech recognition in noise," *Journal of the American Academy of Audiology*, vol. 21, pp. 441–451, July 2010.

[9] J. Szurley, A. Bertrand, B. V. Dijk, and M. Moonen, "Binaural noise cue preservation in a binaural noise reduction system with a remote microphone signal," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, pp. 952–966, May 2016.

[10] A. Bertrand and M. Moonen, "Robust distributed noise reduction in hearing aids with external acoustic sensor nodes," *EURASIP Journal on Advances in Signal Processing*, vol. 2009, 2009.

[11] A. Flammini, P. Ferrari, D. Marioli, E. Sisinni, and A. Taroni, "Wired and wireless sensor networks for industrial applications," *Microelectronics Journal*, vol. 40, pp. 1322–1336, Sept. 2009.

[12] M. F. Othman and K. Shazali, "Wireless sensor network applications: A study in environment monitoring system," *Procedia Engineering*, vol. 41, pp. 1204–1210, 2012.

[13] A. Davis and H. Chang, "Airport protection using wireless sensor networks," in *Proceedings of IEEE Conference on Technologies for Homeland Security (HST)*, Nov. 2012.

[14] N. Xu, S. Rangwala, K. K. Chintalapudi, D. Ganesan, A. Broad, R. Govindan, and D. Estrin, "A wireless sensor network for structural monitoring," in *Proceedings of International conference on embedded networked sensor systems*, pp. 13–24, ACM, Nov. 2004.

[15] D. Estrin, L. Girod, G. Pottie, and M. Srivastava, "Instrumenting the world with wireless sensor networks," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 4, pp. 2033–2036 vol.4, 2001.

[16] S. Haykin and K. J. R. Liu, *Handbook on array processing and sensor networks*, vol. 63. John Wiley & Sons, 2010.

[17] D. Culler, D. Estrin, and M. Srivastava, "Overview of sensor networks," *Computer*, vol. 37, pp. 41–49, Aug 2004.

[18] C. Buratti, A. Conti, D. Dardari, and R. Verdone, "An overview on wireless sensor networks technology and evolution," *Sensors*, vol. 9, pp. 6869–6896, 2009.

[19] H. Gharavi and B. Hu, "Cooperative diversity routing and transmission for wireless sensor networks," *IET Wireless Sensor Systems*, vol. 3, pp. 277–288, Dec. 2013.

[20] S. Bhattacharjee, P. Roy, S. Ghosh, S. Misra, and M. S. Obaidat, "Wireless sensor network-based fire detection, alarming, monitoring and prevention system for bord-and-pillar coal mines," *Journal of Systems and Software*, vol. 85, pp. 571–581, 2012.

[21] K. Bouabdellah, H. Noureddine, and S. Larbi, "Using wireless sensor networks for reliable forest fires detection," *Procedia Computer Science*, vol. 19, pp. 794–801, 2013.

[22] J. Liang and Q. Liang, "Design and analysis of distributed radar sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, pp. 1926–1933, Nov. 2011.

[23] A. Hassani, A. Bertrand, and M. Moonen, "Tracking of a rotating object in a wireless sensor network using fuzzy based adaptive IMM filter," in *Proceedings of IEEE Symposium on Communications and Vehicular Technology in the Benelux (SCVT)*, Nov. 2012.

[24] Y. Li, M. T. Thai, and W. Wu, *Wireless Sensor Networks and Applications*. Signals and Communication Technology, Springer, 2008.

[25] S. Movassaghi, M. Abolhasan, J. Lipman, D. Smith, and A. Jamalipour, "Wireless body area networks: A survey," *IEEE Communication Surveys Tutorials*, vol. 16, pp. 1658–1686, Mar. 2014.

[26] A. Bertrand, "Distributed signal processing for wireless EEG sensor networks," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 23, no. 6, pp. 923–935, 2015.

[27] I. F. Akyildiz, T. Melodia, and K. R. Chowdury, "Wireless multimedia sensor networks: A survey," *IEEE Wireless Communications*, vol. 14, pp. 32–39, Dec. 2007.

[28] B. Harjito and S. Han, "Wireless multimedia sensor networks applications and security challenges," in *Proceedings of International Conference on Broadband, Wireless Computing, Communication and Applications (BWCCA)*, pp. 842–846, Nov. 2010.

[29] B. Bejar and M. Vetterli, "Enhancing local - transmitting less - improving global," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2015.

[30] A. Bertrand, "Applications and trends in wireless acoustic sensor networks: a signal processing perspective," in *Proceedings of IEEE Symposium on Communications and Vehicular Technology in the Benelux (SCVT)*, Nov. 2011.

[31] Y. Jia, Y. Luo, Y. Lin, and I. Kozintsev, "Distributed microphone arrays for digital home and office," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2006.

[32] A. Bertrand, J. Callebaut, and M. Moonen, "Adaptive distributed noise reduction for speech enhancement in wireless acoustic sensor networks," in *Proceedings of the International Workshop on Acoustic Signal Enhancement (IWAENC)*, Aug. 2010.

[33] S. Markovich, S. Gannot, and I. Cohen, "A reduced bandwidth binaural MVDR beamformer," in *Proceedings of the International Workshop on Acoustic Signal Enhancement (IWAENC)*, 2010.

[34] S. Doclo, M. Moonen, T. Van den Bogaert, and J. Wouters, "Reduced-bandwidth and distributed MWF-based noise reduction algorithms for binaural hearing aids," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 17, no. 1, pp. 38–51, 2009.

[35] Y. Zeng and R. C. Hendriks, "Distributed delay and sum beamformer for speech enhancement via randomized gossip," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, pp. 260–273, jan 2014.

[36] R. Heusdens, G. Zhang, R. C. Hendriks, Y. Zeng, and W. B. Kleijn, "Distributed MVDR beamforming for (wireless) microphone networks using message passing," in *Proceedings of the International Workshop on Acoustic Signal Enhancement (IWAENC)*, Sept. 2012.

[37] Y. Zeng and R. C. Hendriks, "Distributed delay and sum beamformer for speech enhancement in wireless sensor networks via randomized gossip," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Mar. 2012.

[38] E. Hansler and G. Schmidt, *Acoustic Echo and Noise Control: A Practical Approach*. Wiley-Interscience, 2004.

[39] J. Benesty, T. Gänsler, D. R. Morgan, M. M. Sondhi, and S. L. Gay, *Advances in Network and Acoustic Echo Cancellation*. Springer Berlin Heidelberg, 2001.

[40] P. A. Naylor and N. D. Gaubitch, eds., *Speech Dereverberation*. Springer London, 2010.

[41] E. Habets, *Single- and multi-microphone speech dereverberation using spectral enhancement*. phdthesis, Technische Universiteit Eindhoven, 2007.

[42] T. van Waterschoot and M. Moonen, "Distributed estimation and equalization of room acoustics in a wireless acoustic sensor network," in *Proceedings of the European Signal Processing Conference (EUSIPCO)*, 2012.

[43] D. Ayllón, R. Gil-Pita, M. Rosa-Zurera, and G. Ramos-Auñón, "Limits of speech source localization in acoustic wireless sensor networks," in *Audio Engineering Society Convention 138*, May 2015.

[44] D. Ayllon, R. Gil-Pita, M. Rosa-Zurera, and H. Krim, "Real-time multiple DOA estimation of speech sources in wireless acoustic sensor networks," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2015.

[45] X. Sheng and Y.-H. Hu, "Maximum likelihood multiple-source localization using acoustic energy measurements with wireless sensor networks," *IEEE Transactions on Signal Processing*, vol. 53, pp. 44–53, jan 2005.

[46] A. Canclini, F. Antonacci, A. Sarti, and S. Tubaro, "Acoustic source localization with distributed asynchronous microphone networks," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, pp. 439–443, feb 2013.

[47] A. Canclini, P. Bestagini, F. Antonacci, M. Compagnoni, A. Sarti, and S. Tubaro, "A robust and low-complexity source localization algorithm for asynchronous distributed microphone networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, pp. 1563–1575, Oct. 2015.

[48] J. Chen, K. Yao, and R. Hudson, "Source localization and beamforming," *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 30–39, 2002.

[49] Y. Hioka and W. B. Kleijn, "Distributed blind source separation with an application to audio signals," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2011.

[50] L. Wang, J. D. Reiss, and A. Cavallaro, "Over-determined source separation and localization using distributed microphones," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, pp. 1573–1588, sep 2016.

[51] F. Nesta and M. Omologo, "Cooperative Wiener-ICA for source localization and separation by distributed microphone arrays," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2010.

[52] E. Robledo-Arnuncio and B.-H. Juang, "Blind source separation of acoustic mixtures with distributed microphones," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2007.

[53] J. Plata-Chaves, A. Bertrand, and M. Moonen, "Incremental multiple error filtered-x LMS for node-specific active noise control over wireless acoustic sensor networks," in *Proceedings of IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM)*, 2016.

[54] A. Gonzalez, M. de Diego, M. Ferrer, and G. Pinero, "Multichannel active noise equalization of interior noise," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 14, pp. 110–122, Jan. 2006.

[55] C. Antoñanzas, M. Ferrer, M. de Diego, and A. Gonzalez, "Blockwise frequency domain active noise controller over distributed networks," *Applied Sciences*, vol. 6, p. 124, Apr. 2016.

[56] M. Ferrer, M. de Diego, G. Piñero, and A. Gonzalez, "Active noise control over adaptive distributed networks," *Signal Processing*, vol. 107, pp. 82–95, Feb. 2015.

[57] S. Elliott and C. Boucher, "Interaction between multiple feedforward active control systems," *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 4, pp. 521–530, 1994.

[58] J. Lorente, C. Antonanzas, M. Ferrer, and A. Gonzalez, "Block-based distributed adaptive filter for active noise control in a collaborative network," in *Proceedings of the European Signal Processing Conference (EUSIPCO)*, pp. 310–314, 2015.

[59] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, pp. 215–233, Jan. 2007.

[60] A. Bertrand and M. Moonen, "Consensus-based distributed total least squares estimation in ad hoc wireless sensor networks," *IEEE Transactions on Signal Processing*, vol. 59, no. 5, pp. 2320–2330, 2011.

[61] C. Lopes and A. Sayed, "Incremental adaptive strategies over distributed networks," *IEEE Transactions on Signal Processing*, vol. 55, pp. 4064–4077, Aug 2007.

[62] J. Chen and A. H. Sayed, "Diffusion adaptation strategies for distributed optimization and learning over networks," *IEEE Transactions on Signal Processing*, vol. 60, no. 8, pp. 4289–4305, 2012.

[63] A. H. Sayed, S.-Y. Tu, J. Chen, X. Zhao, and Z. J. Towfic, "Diffusion strategies for adaptation and learning over networks: an examination of distributed strategies and network behavior," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 155–171, 2013.

[64] C. Lopes and A. Sayed, "Diffusion least-mean squares over adaptive networks: Formulation and performance analysis," *IEEE Transactions on Signal Processing*, vol. 56, pp. 3122–3136, July 2008.

[65] F. S. Cattivelli, C. G. Lopes, and A. H. Sayed, "Diffusion recursive least-squares for distributed estimation over adaptive networks," *IEEE Transactions on Signal Processing*, vol. 56, no. 5, pp. 1865–1877, 2008.

[66] M. O'Connor, W. B. Kleijn, and T. Abhayapala, "Distributed sparse MVDR beamforming using the bi-alternating direction method of multipliers," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016.

[67] A. Bertrand and M. Moonen, "Distributed adaptive node-specific signal estimation in fully connected sensor networks part I: sequential node updating," *IEEE Transactions on Signal Processing*, vol. 58, pp. 5277–5291, 2010.

[68] A. Bertrand and M. Moonen, "Distributed adaptive node-specific signal estimation in fully connected sensor networks part II: simultaneous and asynchronous node updating," *IEEE Transactions on Signal Processing*, vol. 58, pp. 5292–5306, 2010.

[69] A. Bertrand and M. Moonen, "Distributed node-specific LCMV beamforming in wireless sensor networks," *IEEE Transactions on Signal Processing*, vol. 60, pp. 233–246, January 2012.

[70] S. Markovich-Golan, S. Gannot, and I. Cohen, "Distributed multiple constraints generalized sidelobe canceler for fully connected wireless acoustic sensor networks," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, pp. 343–356, Feb. 2013.

[71] A. Bertrand and M. Moonen, "Distributed adaptive estimation of node-specific signals in wireless sensor networks with a tree topology," *IEEE Transactions on Signal Processing*, vol. 59, no. 5, pp. 2196–2210, 2011.

[72] J. Plata-Chaves, A. Bertrand, M. Moonen, S. Theodoridis, and A. Zoubir, "Heterogeneous and multi-task wireless sensor networks - algorithms, applications and challenges," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, pp. 450 – 465, Apr. 2017.

[73] S. Srinivasan and A. C. den Brinker, "Rate-constrained beamforming in binaural hearing aids," *EURASIP Journal on Advances in Signal Processing*, vol. 2009, pp. 1–10, 2009.

[74] I. Himawan, I. McCowan, and S. Sridharan, "Clustered blind beamforming from ad-hoc microphone arrays," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, pp. 661–676, May 2011.

[75] G. Mateos, I. D. Schizas, and G. B. Giannakis, "Distributed recursive least-squares for consensus-based in-network adaptive estimation," *IEEE Transactions on Signal Processing*, vol. 57, no. 11, pp. 4583–4588, 2009.

[76] A. G. Dimakis, S. Kar, J. M. F. Moura, M. G. Rabbat, and A. Scaglione, "Gossip algorithms for distributed signal processing," *Proceedings of the IEEE*, vol. 98, no. 11, pp. 1847–1864, 2010.

[77] F. S. Cattivelli and A. H. Sayed, "Diffusion LMS strategies for distributed estimation," *IEEE Transactions on Signal Processing*, vol. 58, no. 3, pp. 1035–1048, 2010.

[78] S. Chouvardas, K. Slavakis, and S. Theodoridis, "Adaptive robust distributed learning in diffusion sensor networks," *IEEE Transactions on Signal Processing*, vol. 59, no. 10, pp. 4692–4707, 2011.

[79] A. Dogandzic and B. Zhang, "Distributed estimation and detection for sensor networks using hidden Markov random field models," *IEEE Transactions on Signal Processing*, vol. 54, pp. 3200–3215, Aug. 2006.

[80] J. Fang and H. Li, "Distributed estimation of Gauss Markov random fields with one-bit quantized data," *IEEE Signal Processing Letters*, vol. 17, pp. 449–452, May 2010.

[81] A. Koppel, B. M. Sadler, and A. Ribeiro, "Proximity without consensus in online multi-agent optimization," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3726–3730, 2016.

[82] J. Plata-Chaves, N. Bogdanovic, and K. Berberidis, "Distributed incremental-based RLS for node-specific parameter estimation over adaptive networks," in *Proceedings of the European Signal Processing Conference (EUSIPCO)*, 2013.

[83] N. Bogdanovic, J. Plata-Chaves, and K. Berberidis, "Distributed incremental-based LMS for node-specific adaptive parameter estimation," *IEEE Transactions on Signal Processing*, vol. 62, pp. 5382–5397, Oct 2014.

[84] J. Plata-Chaves, N. Bogdanovic, and K. Berberidis, "Distributed diffusion-based LMS for node-specific parameter estimation over adaptive networks," *IEEE Transactions on Signal Processing*, vol. 13, no. 63, pp. 3448–3460, 2015.

[85] J. Plata-Chaves, M. H. Bahari, M. Moonen, and A. Bertrand, "Unsupervised diffusion-based LMS for node-specific parameter estimation over wireless sensor networks," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016.

[86] B. Cornelis, M. Moonen, and J. Wouters, "Reduced-bandwidth multi-channel wiener filter based binaural noise reduction and localization cue preservation in binaural hearing aids," *Signal Processing*, vol. 99, pp. 1–16, jun 2014.

[87] A. Hassani, A. Bertrand, and M. Moonen, "Distributed node-specific direction-of-arrival estimation in wireless acoustic sensor networks," in *Proceedings of the European Signal Processing Conference (EUSIPCO)*, 2013.

[88] F. de la Hucha Arce, F. Rosas, M. Moonen, M. Verhelst, and A. Bertrand, "Energy-vs-performance trade-offs in speech enhancement in wireless acoustic sensor networks," in *Proceedings of the European Signal Processing Conference (EUSIPCO)*, Aug. 2015.

[89] G. Anastasi, M. Conti, M. D. Francesco, and A. Passarella, "Energy conservation in wireless sensor networks: A survey," *Ad Hoc Networks*, vol. 7, pp. 537–568, may 2009.

[90] T. Srisooksai, K. Keamarungsi, P. Lamsrichan, and K. Araki, "Practical data compression in wireless sensor networks: A survey," *Journal of Network and Computer Applications*, vol. 35, pp. 37–59, 2012.

[91] N. Kimura and S. Latifi, "A survey on data compression in wireless sensor networks," in *Proceedings of International Conference on Information Technology: Coding and Computing*, Apr. 2005.

[92] K. Zhang, X. Li, P. Zhang, and H. Li, "Optimal linear estimation fusion - part VI: sensor data compression," in *Proceedings of International Conference of Information Fusion*, pp. 221–228, July 2003.

[93] M. Gastpar, P. L. Dragotti, and M. Vetterli, "The distributed Karhunen - Loeve transform," *IEEE Transactions on Information Theory*, vol. 52, pp. 5177–5196, dec 2006.

[94] I. Schizas, G. Giannakis, and Z.-Q. Luo, "Distributed estimation using reduced-dimensionality sensor observations," *IEEE Transactions on Signal Processing*, vol. 55, pp. 4284–4299, aug 2007.

[95] Z.-Q. Luo, G. Giannakis, and S. Zhang, "Optimal linear decentralized estimation in a bandwidth constrained sensor network," in *Proceedings of International Symposium on Information Theory (ISIT)*, 2005.

[96] C. Yu and G. Sharma, "Distributed estimation using reduced dimensionality sensor observations: A separation perspective," in *Proceedings of Annual Conference on Information Sciences and Systems*, Mar. 2008.

[97] T. C. Lawin-Ore and S. Doclo, "Analysis of rate constraints for MWF-based noise reduction in acoustic sensor networks," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011.

[98] G. Rombouts and M. Moonen, "An integrated approach to acoustic noise and echo cancellation," *Signal Processing*, vol. 85, pp. 849–871, apr 2005.

[99] R. Serizel, M. Moonen, J. Wouters, and S. H. Jensen, "Binaural integrated active noise control and noise reduction in hearing aids," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, pp. 1113–1118, May 2013.

[100] O. Schwartz, S. Gannot, and E. A. P. Habets, "Multi-microphone speech dereverberation and noise reduction using relative early transfer functions," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, pp. 240–251, Feb. 2015.

[101] I. Kodrasi and S. Doclo, "Joint dereverberation and noise reduction based on acoustic multichannel equalization," in *Proceedings of the International Workshop on Acoustic Signal Enhancement (IWAENC)*, Sept. 2014.

[102] K. Ngo, S. Doclo, A. Spriet, M. Moonen, J. Wouters, and S. H. Jensen, "An integrated approach for noise reduction and dynamic range compression in hearing aids," in *Proceedings of the European Signal Processing Conference (EUSIPCO)*, pp. 1–5, Aug 2008.

[103] R. Serizel, M. Moonen, B. Van Dijk, and J. Wouters, "Low-rank approximation based multichannel Wiener filtering algorithms for noise reduction in cochlear implants," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 22, no. 4, pp. 785–799, 2014.

[104] A. Bertrand and M. Moonen, "Efficient sensor subset selection and link failure response for linear MMSE signal estimation in wireless sensor networks," in *Proceedings of the European Signal Processing Conference (EUSIPCO)*, pp. 1092–1096, Aug. 2010.

[105] A. Spriet, M. Moonen, and J. Wouters, "Robustness analysis of multichannel Wiener filtering and generalized sidelobe cancellation for multimicrophone noise reduction in hearing aid applications," *IEEE Transactions on Speech and Audio Processing*, vol. 13, pp. 487–503, July 2005.

[106] J. Szurley, A. Bertrand, and M. Moonen, "Distributed adaptive node-specific signal estimation in heterogeneous and mixed-topology wireless sensor networks," *Signal Processing*, vol. 117, no. 12, pp. 44–60, 2015.

[107] D. Shah, "Gossip algorithms," *Foundations and Trends in Networking*, vol. 3, no. 1, pp. 1–125, 2007.

[108] J. Szurley, A. Bertrand, and M. Moonen, "Topology-independent distributed adaptive node-specific signal estimation in wireless sensor networks," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 3, pp. 130–144, Mar. 2017.

[109] B. D. Van Veen and K. M. Buckley, "Beamforming: A versatile approach to spatial filtering," *IEEE ASSP magazine*, vol. 5, no. 2, pp. 4–24, 1988.

[110] J. Capon, "High-resolution frequency-wavenumber spectrum analysis," *Proceedings of the IEEE*, vol. 57, no. 8, pp. 1408–1418, 1969.

[111] O. Frost, "An algorithm for linearly constrained adaptive array processing," *Proceedings of the IEEE*, vol. 60, no. 8, pp. 926–935, 1972.

[112] S. Doclo and M. Moonen, "GSVD-based optimal filtering for single and multimicrophone speech enhancement," *IEEE Transactions on Signal Processing*, vol. 50, pp. 2230–2244, 2002.

[113] S. Doclo, A. Spriet, J. Wouters, and M. Moonen, "Frequency-domain criterion for the speech distortion weighted multichannel wiener filter for robust noise reduction," *Speech Communication*, vol. 49, pp. 636–656, jul 2007.

[114] R. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Transactions on Antennas and Propagation*, vol. 34, pp. 276–280, 1986.

[115] 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, 1989.

[116] H. Van Trees, *Detection, Estimation, and Modulation Theory, Optimum Array Processing*. Wiley, 2004.

[117] A. Bertrand and M. Moonen, "Distributed adaptive estimation of covariance matrix eigenvectors in wireless sensor networks with application to distributed PCA," *Signal Processing*, vol. 104, pp. 120–135, 2014.

[118] A. Hassani, A. Bertrand, and M. Moonen, "Cooperative integrated noise reduction and node-specific direction-of-arrival estimation in a fully connected wireless acoustic sensor network," *Signal Processing*, vol. 107, pp. 68–81, Feb. 2015.

[119] A. Hassani, A. Bertrand, and M. Moonen, "Distributed GEVD-based signal subspace estimation in a fully-connected wireless sensor network," in *Proceedings of the European Signal Processing Conference (EUSIPCO)*, Sept. 2014.

[120] A. Hassani, A. Bertrand, and M. Moonen, "Distributed signal subspace estimation based on local generalized eigenvector matrix inversion," in *Proceedings of the European Signal Processing Conference (EUSIPCO)*, pp. 1386–1390, 2015.

[121] A. Scaglione, R. Pagliari, and H. Krim, "The decentralized estimation of the sample covariance," in *Proceedings of Asilomar Conference on Signals, Systems and Computers*, pp. 1722–1726, Oct 2008.

[122] G. H. Golub and C. F. Van Loan, *Matrix Computations, 3rd ed.* Baltimore, MD: John Hopkins Univ. Press, 1996.

[123] A. Hassani, A. Bertrand, and M. Moonen, "GEVD-based low-rank approximation for distributed adaptive node-specific signal estimation in a fully-connected wireless sensor network," *IEEE Transactions on Signal Processing*, vol. 64, pp. 2557–2572, May 2016.

[124] I. Schizas, G. Giannakis, and Z.-Q. Luo, "Distributed estimation using reduced-dimensionality sensor observations," *IEEE Transactions on Signal Processing*, vol. 55, pp. 4284–4299, Aug 2007.

[125] J. Fang and H. Li, "Distributed adaptive quantization for wireless sensor networks: From delta modulation to maximum likelihood," *IEEE Transactions on Signal Processing*, vol. 56, pp. 5246–5257, Oct 2008.

[126] F. Cattivelli and A. Sayed, "Diffusion strategies for distributed kalman filtering and smoothing," *IEEE Transactions on Automatic Control*, vol. 55, pp. 2069–2084, Sept 2010.

[127] S. Das and J. Moura, "Distributed kalman filtering with dynamic observations consensus," *IEEE Transactions on Signal Processing*, vol. 63, pp. 4458–4473, Sept 2015.

[128] R. Abdolee, B. Champagne, and A. Sayed, "Estimation of space-time varying parameters using a diffusion LMS algorithm," *IEEE Transactions on Signal Processing*, vol. 62, pp. 403–418, Jan 2014.

[129] A. Bertrand and M. Moonen, "Distributed signal estimation in sensor networks where nodes have different interests," *Signal Processing*, vol. 92, pp. 1679 –1690, July 2012.

[130] J. Plata-Chaves, A. Bertrand, and M. Moonen, "Distributed signal estimation in a wireless sensor network with partially-overlapping node-specific interests or source observability," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015.

[131] A. Bertrand and M. Moonen, "Distributed eye blink artifact removal in a wireless EEG sensor network," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5890–5894, May 2014.

[132] S. Markovich-Golan, A. Bertrand, M. Moonen, and S. Gannot, "Optimal distributed minimum-variance beamforming approaches for speech enhancement in wireless acoustic sensor networks," *Signal Processing*, vol. 107, pp. 4–20, Feb 2015.

[133] S. Markovich-Golan, S. Gannot, and I. Cohen, "Distributed multiple constraints generalized sidelobe canceler for fully connected wireless acoustic sensor networks," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 21, pp. 343–356, Feb 2013.

[134] S. Markovich, S. Gannot, and I. Cohen, "Multichannel eigenspace beamforming in a reverberant noisy environment with multiple interfering speech signals," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 17, pp. 1071–1086, Aug 2009.

[135] I. Tashev, *Sound Capture and Processing: Practical Approaches*. Wiley, 2009.

[136] P. Stoica and K. Sharman, "Maximum likelihood methods for direction-of-arrival estimation," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 38, no. 7, pp. 1132–1143, 1990.

[137] 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, 1991.

[138] H. Wang and M. Kaveh, "Coherent signal-subspace processing for the detection and estimation of angles of arrival of multiple wide-band sources," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 33, no. 4, pp. 823–831, 1985.

[139] J. Evans, D. Sun, and J. Johnson, "Application of advanced signal processing techniques to angle of arrival estimation in ATC navigation and surveillance systems," tech. rep., DTIC Document, 1982.

[140] F. Sellone, "Robust auto-focusing wideband DOA estimation," *Signal Processing*, vol. 86, pp. 17–37, Jan. 2006.

[141] S. Chandran and M. Ibrahim, "DOA estimation of wide-band signals based on time-frequency analysis," *IEEE Journal of Oceanic Engineering*, vol. 24, no. 1, pp. 116–121, 1999.

[142] M. Wax, T.-J. Shan, and T. Kailath, "Spatio-temporal spectral analysis by eigenstructure methods," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 4, pp. 817–827, 1984.

[143] T. Pham and B. Sadler, "Wideband array processing algorithms for acoustic tracking of ground vehicles," *US Research Laboratory*, 1997.

[144] M. Pesavento, A. 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, 2002.

[145] A. Swindlehurst, B. Ottersten, R. Roy, and T. Kailath, "Multiple invariance ESPRIT," *IEEE Transactions on Signal Processing*, vol. 40, no. 4, pp. 867–881, 1992.

[146] A. Swindlehurst, P. Stoica, and M. Jansson, "Exploiting arrays with multiple invariances using MUSIC and MODE," *IEEE Transactions on Signal Processing*, vol. 49, no. 11, pp. 2511–2521, 2001.

[147] R. Kozick and B. M. Sadler, "Near-field localization of acoustic sources with imperfect spatial coherence, distributed processing, and low communication bandwidth," in *Proceedings of Aerospace Sensing, Simulation, and Controls*, pp. 52–63, 2001.

[148] G. Müller and M. Möser, *Handbook of engineering acoustics*. Springer, 2013.

[149] M. Azimi-Sadjadi, A. Pezeshki, and N. Roseveare, "Wideband DOA estimation algorithms for multiple moving sources using unattended acoustic sensors," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 44, pp. 1585–1599, Oct 2008.

[150] R. Serizel, M. Moonen, B. Van Dijk, and J. Wouters, "Rank-1 approximation based multichannel Wiener filtering algorithms for noise reduction in cochlear implants," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013.

[151] 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.

[152] J. B. Allen and D. A. Berkley, "Image method for efficiently simulating small room acoustics," *The Journal of the Acoustical Society of America*, vol. 65, no. 4, pp. 943–950, 1979.

[153] S. Chouvardas, M. Muma, K. Hamaidi, S. Theodoridis, and A. M. Zoubir, "Distributed robust labeling of audio sources in heterogeneous wireless sensor networks," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5783–5787, 2015.

[154] F. K. Teklehaymanot, M. Muma, B. Béjar, P. Binder, A. M. Zoubir, and M. Vetterli, "Robust diffusion-based unsupervised object labelling in distributed camera networks," in *Proceedings of IEEE AFRICON*, pp. 1–6, 2015.

[155] J. Chen, C. Richard, and A. H. Sayed, "Multitask diffusion adaptation over networks," *IEEE Transactions on Signal Processing*, vol. 62, no. 16, pp. 4129–4144, 2014.

[156] V. C. Gogineni and M. Chakraborty, "Diffusion adaptation over clustered multitask networks based on the affine projection algorithm," 2015 [Online]. Available: http://arxiv.org/abs/1507.08566.

[157] V. C. Gogineni and M. Chakraborty, "Distributed multi-task APA over adaptive networks based on partial diffusion," 2015 [Online]. Available: http://arxiv.org/abs/1509.09157.

[158] R. Nassif, C. Richard, A. Ferrari, and A. H. Sayed, "Multitask diffusion adaptation over asynchronous networks," *IEEE Transactions on Signal Processing*, vol. 64, pp. 2835–2850, June 2016.

[159] A. Hassani, J. Plata-Chaves, A. Bertrand, and M. Moonen, "Multi-task wireless acoustic sensor network for node-specific speech enhancement and DOA estimation," in *Proceedings of IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM)*, 2016.

[160] A. Hassani, A. Bertrand, and M. Moonen, "LCMV beamforming with subspace projection for multi-speaker speech enhancement," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016.

[161] S. M. Golan, S. Gannot, and I. Cohen, "Subspace tracking of multiple sources and its application to speakers extraction," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2010.

[162] E. Hadad, S. Doclo, and S. Gannot, "The binaural LCMV beamformer and its performance analysis," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, pp. 543–558, March 2016.

[163] A. Bertrand and M. Moonen, "Energy-based multi-speaker voice activity detection with an ad hoc microphone array," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 85–88, March 2010.

[164] E. Habets, "Room impulse response (RIR) generator," 2010.

[165] R. M. Baumgärtel, M. Krawczyk-Becker, D. Marquardt, C. Völker, H. Hu, T. Herzke, G. Coleman, K. Adiloğlu, S. M. A. Ernst, T. Gerkmann, S. Doclo, B. Kollmeier, V. Hohmann, and M. Dietz, "Comparing binaural pre-processing strategies I: Instrumental evaluation," *Trends in Hearing*, vol. 19, pp. 1–16, 2015.

[166] Y. Hu and P. C. Loizou, "A comparative intelligibility study of single-microphone noise reduction algorithms," *The Journal of the Acoustical Society of America*, vol. 122, no. 3, pp. 1777–1786, 2007.

[167] Y. Hu and P. C. Loizou, "Subjective comparison and evaluation of speech enhancement algorithms," *Speech Communication*, vol. 49, no. 7, pp. 588–601, 2007.

[168] J. E. Greenberg, P. M. Peterson, and P. M. Zurek, "Intelligibility-weighted measures of speech-to-interference ratio and speech system performance," *The Journal of the Acoustical Society of America*, vol. 94, pp. 3009–3010, 1993.

[169] ANSI S.3.5-1997, "American national standard methods for calculation of the speech intelligibility index." Tech. Rep., Acoustic Society of America„ June 1997.

[170] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, "An algorithm for intelligibility prediction of time-frequency weighted noisy speech," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 7, pp. 2125–2136, 2011.

[171] ITU-T Rec. P.862, "Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs." Tech. Rep., ITU- T, Feb. 2001.

[172] S. Gannot, E. Vincent, S. Markovich-Golan, and A. Ozerov, "A consolidated perspective on multimicrophone speech enhancement and source separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 4, pp. 692–730, 2017.

[173] B. Cornelis, M. Moonen, and J. Wouters, "Performance analysis of multichannel wiener filter-based noise reduction in hearing aids under second order statistics estimation errors," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, pp. 1368–1381, July 2011.

[174] A. Hassani, A. Bertrand, and M. Moonen, "Real-time implementation of distributed speech enhancement using the GEVD-based DANSE algorithm." available online at: https://youtu.be/dF4IfJU9xhE, 2016.

[175] M. Nilsson, S. D. Soli, and J. A. Sullivan, "Development of the hearing in noise test for the measurement of speech reception thresholds in quiet and in noise," *The Journal of the Acoustical Society of America*, vol. 95, pp. 1085–1099, 1994.

[176] A. Varga and H. J. M. Steeneken, "Assessment for automatic speech recognition: II. NOISEX-92: A database and an experiment to study the effect of additive noise on speech recognition systems," *Speech Communication*, vol. 12, pp. 247–251, July 1993.

[177] Auditec, "Auditory tests (revised)." Compact disc, Auditec, St. Louis, 1997.

[178] S. Moller and R. Heusdens, "Objective estimation of speech quality for communication systems," *Proceedings of the IEEE*, vol. 101, pp. 1955–1967, Sept. 2013.

[179] P. Mowlaee, R. Saeidi, M. G. Christensen, and R. Martin, "Subjective and objective quality assessment of single-channel speech separation algorithms," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012.

[180] Y. Hu and P. C. Loizou, "Evaluation of objective quality measures for speech enhancement," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, pp. 229–238, Jan. 2008.

[181] R. Crochiere, "A weighted overlap-add method of short-time fourier analysis/synthesis," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 28, no. 1, pp. 99–102, 1980.

[182] R. Brennan and T. Schneider, "A flexible filterbank structure for extensive signal manipulations in digital hearing aids," in *Proceedings of IEEE International Symposium on Circuits and Systems*, 1998.

[183] D. Ampeliotis, N. Bogdanovic, and K. Berberidis, "Coalitional games for a distributed signal enhancement application," in *Proceedings of the European Signal Processing Conference (EUSIPCO)*, 2015.

[184] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[185] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1989.

[186] L. Li and J. A. Chambers, "A new incremental affine projection-based adaptive algorithm for distributed networks," *Signal Processing*, vol. 88, no. 10, pp. 2599–2603, 2008.

[187] J. Chen and A. H. Sayed, "Distributed Pareto-optimal solutions via diffusion adaptation," in *Proceedings of IEEE Statistical Signal Processing Workshop (SSP)*, pp. 648–651, 2012.

[188] J. Chen and A. H. Sayed, "Distributed pareto optimization via diffusion strategies," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 2, pp. 205–220, 2013.

[189] J. Tsitsiklis, D. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Transactions on Automatic Control*, vol. 31, no. 9, pp. 803–812, 1986.

[190] R. C. Hendriks, Z. Erkin, and T. Gerkmann, "Privacy-preserving distributed speech enhancement for wireless sensor networks by processing in the encrypted domain," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2013.

[191] R. C. Hendriks, Z. Erkin, and T. Gerkmann, "Privacy preserving distributed beamforming based on homomorphic encryption," in *Proceedings of the European Signal Processing Conference (EUSIPCO)*, pp. 1–5, Sept. 2013.

[192] M. H. Bahari, A. Bertrand, and M. Moonen, "Blind sampling rate offset estimation based on coherence drift in wireless acoustic sensor networks," in *Proceedings of the European Signal Processing Conference (EUSIPCO)*, 2015.

[193] M. H. Bahari, A. Bertrand, and M. Moonen, "Blind sampling rate offset estimation for wireless acoustic sensor networks through weighted least-squares coherence drift estimation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, pp. 674–686, Mar. 2017.

[194] J. Benesty, J. Chen, Y. Huang, and J. Dmochowski, "On microphone-array beamforming from a MIMO acoustic signal processing perspective," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, pp. 1053–1065, Mar. 2007.

[195] Y. Ephraim and H. Van Trees, "A signal subspace approach for speech enhancement," *IEEE Transactions on Speech and Audio Processing*, vol. 3, pp. 251–266, Jul 1995.

[196] E. Kokiopoulou, J. Chen, and Y. Saad, "Trace optimization and eigenproblems in dimension reduction methods," *Numerical Linear Algebra with Applications*, vol. 18, no. 3, pp. 565–602, 2011.

Curriculum Vitae



Amin Hassani received his B.Sc. degree in electrical engineering (electronics) in 2005 and his M.Sc. degree (with honors) in electrical engineering (systems and control) in 2010 from Ferdowsi university of Mashhad, Iran. In 2012, he joined KU Leuven, and since 2013 he has been a Ph.D. candidate at the department of electrical engineering (ESAT) at KU Leuven, Belgium. During his Ph.D., he has worked as a research assistant on the FWO project 'Wireless Acoustic Sensor Networks for Extended Auditory Communication', and the European FP7 project 'HANDiCAMS - Heterogeneous Ad-Hoc Networks for Distributed, Cooperative, and Adaptive Multimedia Signal Processing'.

Currently he is working on the European H2020 project 'D-NOISE - Distributed Network of Active Noise Equalizers for Multi-User Sound Control'. From 2010-2012 he was a full-time lecturer and research assistant in the department of robotics and electrical engineering, Eqbal Lahoori institute of higher education, Mashhad, Iran. His research interests include array signal processing, beamforming and localization, wireless (acoustic) sensor networks, distributed signal and speech enhancement, and distributed active noise control and equalization.

Publication List

International Journal Papers

1. **Hassani A.**, Plata-Chaves J., Bahari M.H., Moonen M., Bertrand A., "Multi-Task Wireless Sensor Network for Joint Distributed Node-Specific Signal Enhancement, LCMV Beamforming and DOA Estimation", IEEE Journal of Selected Topics in Signal Processing, vol. 11, no. 3, Apr. 2017, pp. 518 - 533.
2. **Hassani A.**, Bertrand A., Moonen M., "GEVD-Based Low-Rank Approximation for Distributed Adaptive Node-Specific Signal Estimation in Wireless Sensor Networks", IEEE Transactions on Signal Processing, vol. 64, no. 10, May 2016, pp. 2557-2572.
3. **Hassani A.**, Bertrand A., Moonen M., "Cooperative Integrated Noise Reduction and Node-Specific Direction-of-Arrival Estimation in a Fully Connected Wireless Acoustic Sensor Network", Signal Processing, vol. 107, Feb. 2015, pp. 68-81.

Journal Papers in Preparation

1. **Hassani A.**, Bertrand A., Moonen M., "Distributed GEVD-based Speech Enhancement with Collaborating Microphone Arrays: Real-Time Implementation and Instrumental Evaluation".

International Conference Papers

1. **Hassani A.**, Plata-Chaves J., Bertrand A., Moonen M., "Multi-Task Wireless Acoustic Sensor Network for Node-Specific Speech Enhancement and DOA Estimation", in Proc. of the IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM), Rio De Janeiro, Brazil, Jul. 2016, pp. 1-5.

2. **Hassani A.**, Bertrand A., Moonen M., "LCMV Beamforming with Subspace Projection for Multi-Speaker Speech Enhancement", in Proc. of the IEEE International Conference on Acoustics, Speech and Signal processing (ICASSP), Shanghai, China, Mar. 2016, pp. 91-95.

3. **Hassani A.**, Bertrand A., Moonen M., "Distributed Signal Subspace Estimation Based on Local Generalized Eigenvector Matrix Inversion", in Proc. of the European Signal Processing Conference (EUSIPCO), Nice, France, Sep. 2015, pp. 1401-1405.

4. **Hassani A.**, Bertrand A., Moonen M., "Low-rank Approximation-Based Distributed Node-Specific Signal Estimation in a Fully-Connected Wireless Sensor Network", in Proc. of the 40th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brisbane, Australia, Apr. 2015, pp. 2839-2843.

5. **Hassani A.**, Bertrand A., Moonen M., "Distributed GEVD-Based Signal Subspace Estimation in a Fully-Connected Wireless Sensor Network", in Proc. of the European Signal Processing Conference (EUSIPCO), Lisbon, Portugal, Sep. 2014, pp. 1292-1296.

6. **Hassani A.**, Bertrand A., Moonen M., "Distributed Node-Specific Direction-of-Arrival Estimation in Wireless Acoustic Sensor Networks", in Proc. of the European Signal Processing Conference (EUSIPCO), Marrakesh, Morocco, Sep. 2013, pp. 1-5.

7. **Hassani A.**, Bertrand A., Moonen M., "Tracking of a Rotating Object in a Wireless Sensor Network Using Fuzzy Based Adaptive IMM Filter", in Proc. of the 19th IEEE Symposium on Communications and Vehicular Technology in the Benelux (SCVT2012), Eindhoven, Netherlands, Nov. 2012.

International Demonstration Abstract

1. **Hassani A.**, Bertrand A., Moonen M., "Real-Time Distributed Speech Enhancement with Two Collaborating Microphone Arrays", at IEEE International Conference on Acoustics, Speech and Signal processing (ICASSP), New Orleans, USA, Mar. 2017.



FACULTY OF ENGINEERING SCIENCE
DEPARTMENT OF ELECTRICAL ENGINEERING
Stadius Center for Dynamical Systems,
Signal Processing and Data Analytics
Kasteelpark Arenberg 10, bus 2446
B-3001 Leuven
Belgium
amin.hassani@esat.kuleuven.be