Applied and Numerical Harmonic Analysis


Gitta Kutyniok
Holger Rauhut
Robert J. Kunsch
Editors

# Compressed Sensing in Information Processing

Birkhäuser

# Applied and Numerical Harmonic Analysis

Gitta Kutyniok • Holger Rauhut • Robert J. Kunsch
Editors

# Compressed Sensing in Information Processing

Birkhäuser

*Editors*
Gitta Kutyniok
Mathematisches Institut
Ludwig Maximilian University of Munich
München, Bayern, Germany

Holger Rauhut
Lehrstuhl für Mathematik
RWTH Aachen University
Aachen, Nordrhein-Westfalen, Germany

Robert J. Kunsch
Lehrstuhl für Mathematik
RWTH Aachen University
Aachen, Nordrhein-Westfalen, Germany

ISSN 2296-5009 ISSN 2296-5017 (electronic)
Applied and Numerical Harmonic Analysis
ISBN 978-3-031-09744-7 ISBN 978-3-031-09745-4 (eBook)
https://doi.org/10.1007/978-3-031-09745-4

Mathematics Subject Classification: 94Axx, 65F22, 68U10, 90C25, 15B52, 86A10



This book is published under the imprint Birkhäuser, www.birkhauser-science.com by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

# ANHA Series Preface

The *Applied and Numerical Harmonic Analysis* (ANHA) book series aims to provide the engineering, mathematical, and scientific communities with significant developments in harmonic analysis, ranging from abstract harmonic analysis to basic applications. The title of the series reflects the importance of applications and numerical implementation, but richness and relevance of applications and implementation depend fundamentally on the structure and depth of theoretical underpinnings. Thus, from our point of view, the interleaving of theory and applications and their creative symbiotic evolution are axiomatic.

Harmonic analysis is a wellspring of ideas and applicability that has flourished, developed, and deepened over time within many disciplines and by means of creative cross-fertilization with diverse areas. The intricate and fundamental relationship between harmonic analysis and fields such as signal processing, partial differential equations (PDEs), and image processing is reflected in our state-of-the-art ANHA series.

Our vision of modern harmonic analysis includes a broad array of mathematical areas, e.g., wavelet theory, Banach algebras, classical Fourier analysis, time-frequency analysis, deep learning, and fractal geometry, as well as the diverse topics that impinge on them.

For example, wavelet theory can be considered an appropriate tool to deal with some basic problems in digital signal processing, speech and image processing, geophysics, pattern recognition, biomedical engineering, and turbulence. These areas implement the latest technology from sampling methods on surfaces to fast algorithms and computer vision methods. The underlying mathematics of wavelet theory depends not only on classical Fourier analysis but also on ideas from abstract harmonic analysis, including von Neumann algebras and the affine group. This leads to a study of the Heisenberg group and its relationship to Gabor systems, and of the metaplectic group for a meaningful interaction of signal decomposition methods.

The unifying influence of wavelet theory in the aforementioned topics illustrates the justification for providing a means for centralizing and disseminating information from the broader, but still focused, area of harmonic analysis. This will be a key

role of ANHA. We intend to publish with the scope and interaction that such a host of issues demands.

Along with our commitment to publish mathematically significant works at the frontiers of harmonic analysis, we have a comparably strong commitment to publish major advances in the following applicable topics in which harmonic analysis plays a substantial role:

*Analytic Number theory * Antenna Theory * Artificial Intelligence * Biomedical Signal Processing * Classical Fourier Analysis * Coding Theory * Communications Theory * Compressed Sensing * Crystallography and Quasi-Crystals * Data Mining * Data Science * Deep Learning * Digital Signal Processing * Dimension Reduction and Classification * Fast Algorithms * Frame Theory and Applications * Gabor Theory and Applications * Geophysics * Image Processing * Machine Learning * Manifold Learning * Numerical Partial Differential Equations * Neural Networks * Phaseless Reconstruction * Prediction Theory * Quantum Information Theory * Radar Applications * Sampling Theory (Uniform and Non-uniform) and Applications * Spectral Estimation * Speech Processing * Statistical Signal Processing * Super-resolution * Time Series * Time-Frequency and Time-Scale Analysis * Tomography * Turbulence * Uncertainty Principles *Waveform design * Wavelet Theory and Applications

The above point of view for the ANHA book series is inspired by the history of Fourier analysis itself, whose tentacles reach into so many fields.

In the last two centuries, Fourier analysis has had a major impact on the development of mathematics, on the understanding of many engineering and scientific phenomena, and on the solution of some of the most important problems in mathematics and the sciences. Historically, Fourier series were developed in the analysis of some of the classical PDEs of mathematical physics; these series were used to solve such equations. In order to understand Fourier series and the kinds of solutions they could represent, some of the most basic notions of analysis were defined, e.g., the concept of "function." Since the coefficients of Fourier series are integrals, it is no surprise that Riemann integrals were conceived to deal with uniqueness properties of trigonometric series. Cantor's set theory was also developed because of such uniqueness questions.

A basic problem in Fourier analysis is to show how complicated phenomena, such as sound waves, can be described in terms of elementary harmonics. There are two aspects of this problem: first, to find, or even define properly, the harmonics or spectrum of a given phenomenon, e.g., the spectroscopy problem in optics; second, to determine which phenomena can be constructed from given classes of harmonics, as done, for example, by the mechanical synthesizers in tidal analysis.

Fourier analysis is also the natural setting for many other problems in engineering, mathematics, and the sciences. For example, Wiener's Tauberian theorem in Fourier analysis not only characterizes the behavior of the prime numbers but is a fundamental tool for analyzing the ideal structures of Banach algebras. It also provides the proper notion of spectrum for phenomena such as white light. This

latter process leads to the Fourier analysis associated with correlation functions in filtering and prediction problems. These problems, in turn, deal naturally with Hardy spaces in complex analysis, as well as inspiring Wiener to consider communications engineering in terms of feedback and stability, creating his cybernetics. This latter theory develops concepts to understand complex systems such as learning and cognition and neural networks, and it is arguably a precursor of deep learning and its spectacular interactions with data science and AI.

Nowadays, some of the theory of PDEs has given way to the study of Fourier integral operators. Problems in antenna theory are studied in terms of unimodular trigonometric polynomials. Applications of Fourier analysis abound in signal processing, whether with the fast Fourier transform (FFT), or filter design, or the adaptive modeling inherent in time-frequency-scale methods such as wavelet theory.

The coherent states of mathematical physics are translated and modulated Fourier transforms, and these are used, in conjunction with the uncertainty principle, for dealing with signal reconstruction in communications theory. We are back to the raison d'etre of the ANHA series!

College Park, MD — *John Benedetto*
*Wojciech Czaja*
Boston, MA — *Kasso Okoudjou*

# Preface

In April 2014, the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) established the Priority Program 1798 "Compressed Sensing in Information Processing" (CoSIP). The objective of this volume is to offer a comprehensive overview of the scientific highlights obtained in the course of this Priority Program, mainly during the second phase that started in July 2018.

Compressed sensing is an area of research with broad applications in electrical engineering, computer science, and physics. It refers to situations where few measurements already suffice to reconstruct a signal or image, despite the fact that the acquired information leads to an underdetermined system of linear equations. The key insight here is that most real-world signals are inherently sparse, that is, for many natural classes of signals, there exist building blocks such that decompositions of such signals with respect to these building blocks exhibit only a small number of non-zero coefficients. It is remarkable that randomness has been proven most successful in the acquisition step, enabling for a minimal number of measurements. Furthermore, there exist efficient reconstruction algorithms which make this approach feasible in practice.

The area of compressed sensing has attracted great interest of researchers in mathematics and applied sciences since around 2004. A lot of recent research – both in theory and application – are motivated by wireless communication and multiple-input multiple-output channels (MIMO), which gain increasing importance with the advent of digital technologies like the Internet of Things. In particular Chaps. 10, 11, and 13 present an application-driven view point on wireless networks, while Chap. 12 brings MIMO in context with radar imaging. These applications also push forward the development of theory on different models of sparsity such as hierarchical sparsity (see Chap. 1) or low-rank matrix recovery (Chap. 2), as well as theory on covariance estimation (see Chaps. 3 and 4) and recovery algorithms (see Chaps. 5–8). The rise of machine learning and deep neural networks likewise leaves its imprint on compressed sensing-related topics in theory-driven research (see Chaps. 7–9) as well as in research motivated by applications (see Chaps. 10, 13, and 14). Last but not least, the problem of effectively acquiring compressive

measurements is still a challenge in particular applications, see Chap. 15 on moving microphones and Chap. 16 on spherical near-field antenna measurements.

Overall, the network of SPP 1798 comprised more than 60 scientists, and altogether 13 projects were funded in the second period and contributed to this volume (Chaps. 8 and 9 are from the same project, the same holds for Chaps. 10 and 11). With Chap. 16, we also welcome a contribution from a project that has been associated to CoSIP. The aim of this volume is of course not to give a complete presentation of all results that have been obtained by participants of the Priority Program but rather to collect the scientific highlights in order to demonstrate the impact of CoSIP on further researches. The editors and authors hope that this volume will arouse interest in the reader on the various new developments related to compressed sensing that have been promoted by the Priority Program. For further information concerning SPP 1798, please visit https://www.mathc.rwth-aachen.de/spp1798ii/.

München, Germany — Gitta Kutyniok
Aachen, Germany — Holger Rauhut
Aachen, Germany — Robert J. Kunsch
October 2021

# Acknowledgements

First of all, the authors and the editors of this book thank the Deutsche Forschungsgemeinschaft (DFG) for the support within the DFG Priority Program 1798 "Compressed Sensing in Information Processing" (CoSIP) over two entire funding periods, not to forget the additional support in face of the COVID-19 pandemic. Moreover, the editors thank all authors who helped peer review other chapters of this book. We also feel very grateful to Rudolf Mathar, Arash Behboodi, and Maximilian März who contributed a lot to the success of the SPP 1798 as part of the coordination team for long periods of time. Special thanks are devoted to the DFG representatives Iris Leuthen-Schmittuz and Damian Dudek for the very productive cooperation. Last but not least, we thank Leonardo Galli for technical assistance and support during the compilation of this book.

# Contents

# Contributors

**Khaled Ardah** Ilmenau University of Technology, Ilmenau, Germany

**Jeremy Olaore Aweda** University of Rostock, Rostock, Germany

**Arya Bangun** RWTH Aachen University, Aachen, Germany

**Mahdi Barzegar Khalilsarai** Technical University of Berlin, Berlin, Germany

**Arash Behboodi** RWTH Aachen University, Aachen, Germany

**Ali Bereyhi** Friedrich–Alexander University Erlangen–Nürnberg, Erlangen, Germany

**Igor Bjelaković** Fraunhofer Heinrich Hertz Institute, Berlin, Germany

**Giuseppe Caire** Technical University of Berlin, Berlin, Germany

**Jorge Luis Chau** Leibniz Institute of Atmospheric Physics at the University of Rostock, Kühlungsborn, Germany

**Cosme Culotta-López** RWTH Aachen University, Aachen, Germany

**Sjoerd Dirksen** Utrecht University, Utrecht, Netherlands

**Jens Eisert** Free University of Berlin, Berlin, Germany

**Robert F. H. Fischer** Ulm University, Ulm, Germany

**Axel Flinth** Chalmers University of Technology, Gothenburg, Sweden

**Massimo Fornasier** Technical University of Munich, Garching, Germany

**Matthias Frey** Technical University of Berlin, Berlin, Germany

**Tim Fuchs** Technical University of Munich, Garching, Germany

**Benedikt Groß** Free University of Berlin, Berlin, Germany

**David Gross** University of Cologne, Cologne, Germany

**Martin Haardt** Ilmenau University of Technology, Ilmenau, Germany

**Saeid Haghighatshoar** SynSense AG, Zurich, Switzerland

**Dirk Heberling** RWTH Aachen University, Aachen, Germany

**Danfeng Hong** Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing, China

**Peter Jung** Technical University of Berlin, Berlin, Germany
Technical University of Munich, Munich, Germany

**Alihan Kaplan** Technical University of Munich, Munich, Germany

**Fabrice Katzberg** University of Lübeck, Lübeck, Germany

**Felix Krahmer** Technical University of Munich, Munich, Germany

**Gerhard Kramer** Technical University of Munich, Munich, Germany

**Richard Kueng** Johannes Kepler University Linz, Linz, Austria

**Gitta Kutyniok** Ludwig Maximilian University of Munich, Munich, Germany
University of Tromsø, Tromsø, Norway

**Dae Gwan Lee** Catholic University of Eichstätt-Ingolstadt, Eichstätt, Germany

**Ron Levie** Ludwig Maximilian University of Munich, Munich, Germany

**Tianyi Liu** Technical University of Darmstadt, Darmstadt, Germany

**Johannes Maly** Catholic University of Eichstätt-Ingolstadt, Eichstätt, Germany

**Rudolf Mathar** RWTH Aachen University, Aachen, Germany

**Frederic Matter** Technical University of Darmstadt, Darmstadt, Germany

**Alfred Mertins** University of Lübeck, Lübeck, Germany

**Ralf R. Müller** Friedrich–Alexander University Erlangen–Nürnberg, Lübeck, Germany

**Lars Palzer** Technical University of Munich, Munich, Germany

**Marius Pesavento** Technical University of Darmstadt, Darmstadt, Germany

**Götz E. Pfander** Catholic University of Eichstätt-Ingolstadt, Eichstätt, Germany

**Marc E. Pfetsch** Technical University of Darmstadt, Darmstadt, Germany

**Volker Pohl** Technical University of Munich, Munich, Germany

**Michael Rauchensteiner** Technical University of Munich, Munich, Germany

**Holger Rauhut** RWTH Aachen University, Aachen, Germany

**Ingo Roth** Technology Innovation Institute, Abu Dhabi, UAE
Free University of Berlin, Berlin, Germany

**Ekkehard Schnoor** RWTH Aachen University, Aachen, Germany

**Hermann Schulz-Baldes** Friedrich–Alexander University Erlangen–Nürnberg, Erlangen, Germany

**Carmen Sippel** Ulm University, Ulm, Germany

**Yi Song** Technical University of Berlin, Berlin, Germany

**Sławomir Stańczak** Technical University of Berlin, Berlin, Germany

**Dominik Stöger** Catholic University of Eichstätt-Ingolstadt, Eichstätt, Germany

**Juan Miguel Urco** Leibniz Institute of Atmospheric Physics at the University of Rostock, Rostock, Germany
University of Illinois Urbana-Champaign, Champaign, IL, USA

**Tobias Weber** University of Rostock, Rostock, Germany

**Gerhard Wunder** Free University of Berlin, Berlin, Germany

**Tianyu Yang** Technical University of Berlin, Berlin, Germany

**Çağkan Yapar** Technical University of Berlin, Berlin, Germany

**Xinping Yi** University of Liverpool, Liverpool, UK

**Xiao Xiang Zhu** Technical University of Munich, Munich, Germany

# Chapter 1
# Hierarchical Compressed Sensing


**Jens Eisert, Axel Flinth, Benedikt Groß, Ingo Roth, and Gerhard Wunder**

## 1.1 Introduction

The field of compressed sensing studies the recovery of structured signals from linear measurements [12, 19]. Originally focusing on the structure of sparsity of vectors, the framework was quickly extended to the structure of low-rankness of matrices. These structures are simultaneously restrictive and rich. They are restrictive so that they allow for signal recovery using far fewer linear measurements than the ambient dimensions suggest and rich in that they naturally appear in a plethora of applications. That being said, in many practically relevant applications, the signals feature a more restrictive structure than mere sparsity or low-rankness. A particularly important broad class arising in a wealth of contexts is *hierarchically structured signals*. Such structures are in the focus of this book chapter.

J. Eisert
Dahlem Center for Complex Quantum Systems and Department of Mathematics and Computer Science, Freie Universität Berlin, Berlin, Germany
e-mail: jense@physics.fu-berlin.de

A. Flinth
Institute for Electrical Engineering, Chalmers University of Technology, Gothenburg, Sweden
e-mail: flinth@chalmers.se

B. Groß · G. Wunder (✉)
Department of Mathematics and Computer Science, Freie Universität Berlin, Berlin, Germany
e-mail: benedikt.gross@fu-berlin.de; g.wunder@fu-berlin.de

I. Roth
Quantum Research Centre, Technology Innovation Institute, Abu Dhabi, UAE

Dahlem Center for Complex Quantum Systems, Freie Universität Berlin, Berlin, Germany
e-mail: i.roth@fu-berlin.de



G. Kutyniok et al. (eds.), *Compressed Sensing in Information Processing*, Applied and Numerical Harmonic Analysis, https://doi.org/10.1007/978-3-031-09745-4_1

The perhaps simplest examples are constituted by *hierarchically sparse vectors*. A two-level hierarchically sparse vector is a vector consisting of multiple blocks with a restricted support as follows: only a small number of the blocks have non-vanishing entries and the blocks are themselves sparse. An illustrative example can be given via imagining a telecommunication base station responsible for handling a large set of potential users. If in each instance, only a few users actively transmit, and the messages that are transmitted are sparsely representable, the vector compiling all messages in its blocks is hierarchically sparse. The hierarchically sparse vectors will serve as the main illustrative example of the entire chapter. It is straightforward to generalize this notion for vectors with a hierarchy of nested blocks with sparsity assumptions restricting the number of non-vanishing blocks on each level.

Another hierarchical structure of interest is given by replacing the sparsity constraint on the vector-valued blocks by a low-rank assumption of matrix-valued blocks. A motivating example here can be found in quantum tomography, where quantum states can be modelled as low-rank Hermitian matrices. Hierarchical structures of quantum states arise here in the tasks of performing quantum tomography with a partially uncalibrated measurement device or de-mixing sparse sums of quantum states.

An intriguing feature of hierarchically structured signals is that their recovery task is amenable to efficient thresholding algorithms. In general, thresholding algorithms such as the iterative hard-thresholding pursuit are built on the insight that, in contrast to the original recovery problem, the projection onto the set of structured signals is efficient and in fact often particularly simple. This allows one to employ algorithmic strategies such as projective gradient descent.

For hierarchically sparse signals, it turns out that the calculation of the projection has the same computational complexity as the thresholding onto sparse signals. Furthermore, the hierarchical structure allows for the parallelization of the projections for the blocks on each level yielding potential for further reducing the time complexity by exploiting the restrictive structure. Based on this insight, we formally introduce variants of the *iterative hard-thresholding (IHT) algorithm* and the *hard-thresholding pursuit (HTP)* for hierarchically sparse signals.

For the IHT and HTP algorithm, recovery guarantees for measurement maps that act close to isometrically, on sparse vectors, exist. Due to their similarity, the recovery algorithms for hierarchically sparse signals inherit the recovery guarantees from the original IHT and HTP provided the measurement map exhibits a *restricted isometry property (RIP)* that is adapted to the hierarchically structured signal set. We refer to the modified RIP restricted to hierarchically sparse signals as the *hierarchically restricted isometry property (HiRIP)*.

In this chapter, we derive a series of theoretical results concerning the HiRIP. Requiring only HiRIP instead of RIP for the measurement opens up the possibility of exploiting multiple benefits. First, standard measurement ensembles such as random Gaussian matrices can achieve HiRIP with a reduced sampling complexity compared to RIP. The achievable logarithmic improvement mirrors the reduced complexity of the restricted signal set compared to standard sparse

vectors. Second, we introduce an ensemble of operators that do have the HiRIP, but not RIP in any parameter regime. We give this flexible class of operators the name *hierarchical measurements* since they are naturally adapted to hierarchical structures. Hierarchical measurements combine different measurement maps on each level of the hierarchy and, as we show, inherit HiRIP from standard RIP and coherence properties of their constituent maps. An important instance of hierarchical measurement is Kronecker products of measurements such that each factor acts on the blocks of a certain hierarchy level.

Finally, we illustrate how the framework of hierarchical compressed sensing can be applied in applications in machine-type communications and quantum technologies providing motivating examples and evaluations of practical performances.

Let us end with an outline of the remainder of the chapter. In Sects. 1.2 and 1.3, respectively, we formally introduce hierarchically sparse vectors and present the algorithms used to recover them. Section 1.4 is devoted to theoretical results concerning the hierarchical restricted isometry property (HiRIP) and step by step develops a flexible toolkit to establish the HiRIP for large classes of measurement ensembles. In Sect. 1.5, we move on to discussing the sparse, low-rank signal model, including how the recovery algorithms can be adapted. In Sect. 1.6, we provide a more specific discussion of selected applications. We close with a conclusion as well as a small outlook in Sect. 1.7.

## 1.2 Hierarchically Sparse Vectors

We consider structured sparse signals that are vectors over the field $\mathbb{K}$, referring to either the reals $\mathbb{R}$ or the complex numbers $\mathbb{C}$, and are hierarchically structured into blocks. The support is restricted by sparsity assumptions on one or multiple hierarchy levels. The simplest instance of hierarchically sparse signals is two-level hierarchically sparse vector with constant block sizes and sparsities [20, 41–43].

**Definition 1.1 (Two-Level Hierarchically Sparse Vectors)** Let $N, n, s, \sigma \in \mathbb{N}$. A vector $x \in \mathbb{K}^{Nn}$ is called $(s, \sigma)$-*hierarchically sparse*, if it consists of blocks $x_i \in \mathbb{K}^n$, $x^\top = (x_1^\top, \dots, x_i^\top, \dots, x_N^\top)^\top$, where at most $s$ blocks $x_i$ are non-zero, and each of the non-zero blocks is at most $\sigma$-sparse.

For brevity, we write $(s, \sigma)$-*sparse*, dropping the *hierarchically* in the following. We refer to the set of $(s, \sigma)$-sparse vectors in $\mathbb{K}^{Nn}$ as $\mathcal{S}_{s,\sigma}^{N,n}$ or simply $\mathcal{S}$ if the parameters are clear from the context. We also call the support $\operatorname{supp}(x) \subset [N] \times [n]$ of an $(s, \sigma)$-sparse vector a $(s, \sigma)$-*sparse support*, where $[n] \coloneqq \{1, \dots, n\}$. The definition of a two-level hierarchically sparse vectors can be generalized in several directions: We can allow different block sizes and block sparsities. Furthermore, each block is allowed to be a hierarchically sparse vector itself. This gives rise to a more general recursive definition of hierarchically sparse vectors with arbitrary many levels. The defining data of such a general hierarchically sparse vector can

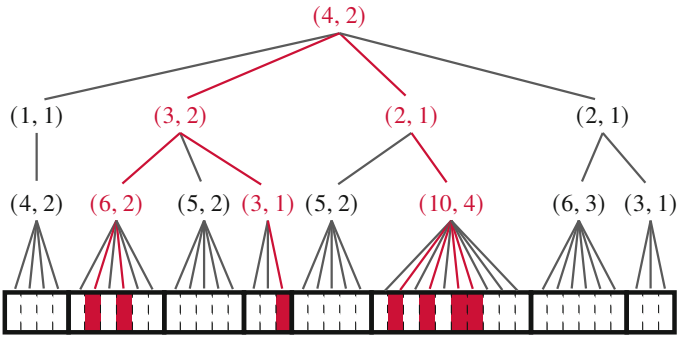



**Fig. 1.1** This figure shows an example of a hierarchically sparse vector. The grouping of the entries is encoded in a rooted tree. The children of a vertex constitute a block at their level. The pair of values at each vertex indicates the block size (the number of children) and the sparsity, i.e. the number of children with non-vanishing entries. The leaves of the tree are identified with the entries of the vector. The support of the vector drawn below and corresponding vertices with non-vanishing entries are highlighted in red. ©2020 IEEE. Reprinted, with permission, from Ref. [35]

be collected in a rooted tree consisting of nodes, labelled by block sizes and sparsities, see Fig. 1.1. We refer to Ref. [35] for a formal definition of general hierarchically sparse vectors. Other special cases of hierarchically sparse vectors have been considered in the literature. Prominent examples are *block sparse* [13, 14] and *level sparse* [3, 28] vectors.

Another setting where the hierarchical sparsity naturally emerges is so-called *bi-sparsity*, see e.g. Ref. [18]. In said reference, a Hermitian matrix $X \in \mathbb{K}^{n\times n}$ is called bisparse if there exists a set $S \subseteq [N]$ with $|S| \leq s$ so that $X_{ij}$ is non-zero, only if both $i$ and $j$ are in $S$. Clearly, any bisparse matrix can be interpreted as an $(s, s)$-sparse vector. More generally, a matrix $Y \in \mathbb{K}^{N\times n}$ with $Y_{ij}$ non-zero for $i \in S$ and $j \in \Sigma$ for sets with cardinalities $|S| = s, |, \Sigma| \leq \sigma$ can be regarded as $(s, \sigma)$-bisparse and in the same manner identified with an $(s, \sigma)$-sparse vector. Bisparsity is of course more restrictive than hierarchical sparsity, but the projection operator onto the set of bisparse matrices is—in stark contrast to its hierarchical sparsity counterpart—NP-hard to compute. Hierarchical sparsity can thus be seen as a relaxation of bisparsity which allows for more efficient recovery procedures. We refer to Ref. [18] for a more comprehensive discussion on these matters, as well as other ways to relax the bisparse structures. We encounter this relaxation in conjunction with blind deconvolution in Sect. 1.6.2 and a non-commutative analog of it in our discussion of blind quantum tomography in Sect. 1.6.3.

For simplicity and notational clarity, we content ourselves to present the framework for two-level hierarchically sparse vectors. It is straightforward to generalize the algorithmic strategies and most analytical results of this chapter to the general definition of hierarchically sparse vectors outlined above, see Ref. [35] for details.

## 1.3 Hierarchical Thresholding and Recovery Algorithms

We study the linear inverse problem of recovering an $(s, \sigma)$-hierarchically sparse vector $x \in \mathbb{K}^{Nn}$ from noisy linear measurements of the form

$$y = Mx + e,$$

where $M \in \mathbb{K}^{m \times Nn}$ is the linear measurement operator and $e \in \mathbb{K}^m$ encodes additive noise. The recovery task can be cast as the constraint optimization problem

$$\underset{x \in \mathbb{K}^{Nn}}{\text{minimize}} \ \frac{1}{2}\|y - Mx\|^2 \quad \text{subject to } x \text{ is } (s, \sigma)\text{-sparse.}, \tag{1.1}$$

where $\|y\| = \left[\sum_i |y_i|^2\right]^{1/2}$ denotes the $\ell_2$-norm.

So-called hard-thresholding algorithms solve the analogous optimization problem to (1.1) for standard $s$-sparse recovery by making use of the projection of a vector $z \in \mathbb{K}^n$ onto the set of $s$-sparse vectors. The projection onto $s$-sparse vectors,

$$\mathbb{T}_s(z) := \underset{x \in \mathbb{K}^n}{\text{argmin}} \|x - z\| \quad \text{subject to } x \ s\text{-sparse},$$

can be computed efficiently via hard thresholding, i.e. by setting to zero all but the $s$ largest entries in absolute value. Note that since the set of $s$-sparse vectors in $\mathbb{K}^n$ is not a convex set, the projection is non-unique. But for the arguments made here every solution works equally well. Using a quick-select algorithm [24], the average computational complexity of the thresholding operation is in $O(n)$ with worst-case complexity $O(n^2)$.

Following the blueprint of model-based compressed sensing [4], we can derive variants of standard hard-thresholding algorithms for the more restrictive sparsity structure under consideration here by modifying the thresholding operator accordingly. The projection of a vector $z \in \mathbb{K}^{Nn}$ onto the set $\mathcal{S}$ of $(s, \sigma)$-hierarchically sparse vectors,

$$\mathbb{T}_{s,\sigma}(z) = \underset{x \in \mathcal{S}}{\text{argmin}} \frac{1}{2}\|x - z\|^2 \,,$$

can be computed via *hierarchical hard thresholding*: first, the standard hard thresholding operation $\mathbb{T}_\sigma$ is applied to each block. Then, all but the $s$ blocks with largest $\ell_2$-norm are set to zero. The procedure is summarized as Algorithm 1 and illustrated in Fig. 1.2. We find that the average computational complexity of the hierarchical thresholding operation scales as $O(Nn)$, i.e. linear in the overall vector space dimension as for the standard hard thresholding. Furthermore, the hard thresholding and $\ell_2$-norm calculation of the different blocks can be parallelized, reducing the time complexity to $O(\max(N, n))$. The hierarchical thresholding operation can be

**Algorithm 1:** Hierarchical hard thresholding

**input** : $z \in \mathbb{K}^{Nn}$, sparsity levels $(s, \sigma)$
**for** $i \in [N]$ **do**
  $x_i = \mathbb{T}_\sigma(z_i)$;
  $\nu_i = \|x_i\|$;
**end**
$I = \operatorname{supp}(\mathbb{T}_s((\nu_1, \ldots, \nu_N)))$;
**for** $i \in [N] \setminus I$ **do**
  $x_i = 0$
**end**
**output** : $(s, \sigma)$-hierarchically sparse vector $x = (x_1^\top, \ldots, x_N^\top)^\top$

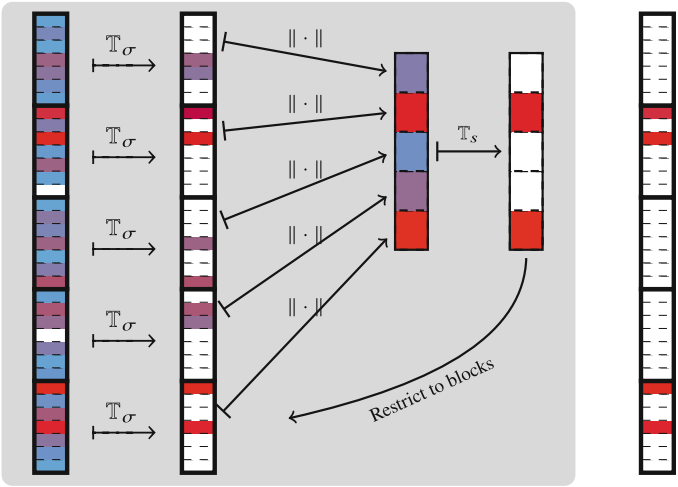



**Fig. 1.2** In this figure, the evaluation of the hierarchical thresholding operator $\mathbb{T}_{s,\sigma}$ is illustrated. Starting with a given dense vector (**a**), each block is thresholded to its best $\sigma$-sparse approximation (**b**). To determine the $s$ dominant blocks, the $\ell_2$-norm is calculated for each block. The resulting vector (**c**) of length $N$ is again thresholded to its best $s$-sparse approximation (**d**). The resulting blocks indicated by the $s$-sparse approximation (**d**) are selected from the $\sigma$-sparse approximation (**b**). The remaining $(s, \sigma)$-sparse support (**e**) is the output of $\mathbb{T}_{s,\sigma}$. ©2020 IEEE. Reprinted, with permission, from Ref. [35]

extended recursively to general hierarchically sparse signals described in Sect. 1.2 without increasing the overall computational complexity.

Equipped with an efficient thresholding operation, we can formulate recovery algorithms for hierarchically sparse signals following standard strategies. A particularly simple algorithm is the *iterative hard thresholding algorithm* [5] which performs a projected gradient descent. The resulting *hierarchical iterative hard-thresholding algorithm* (HiIHT, Algorithm 2 [53]) alternates gradient descent steps of the objective function (1.1) with the hard-thresholding operation $\mathbb{T}_{s,\sigma}$.

**Algorithm 2:** HiIHT algorithm

**input** : data $y \in \mathbb{K}^m$, measurement operator $M \in \mathbb{K}^{m\times Nn}$, sparsity levels $(s, \sigma)$
**initialize**: $x^{(0)} = 0$
**repeat**
  $\bar{x}^{(t)} = x^{(t-1)} + \tau^{(t)} M^* \left(y - Mx^{(t-1)}\right)$;
  $x^{(t)} = \mathbb{T}_{s,\sigma}\left(\bar{x}^{(t)}\right)$;
**until** *stopping criterion is met at* $t = t^*$
**output** : $(s, \sigma)$-sparse vector $x^{(t^*)}$

**Algorithm 3:** HiHTP algorithm

**input** : data $y \in \mathbb{K}^m$, measurement operator $M \in \mathbb{K}^{m\times Nn}$, sparsity levels $(s, \sigma)$
**initialize**: $x^{(0)} = 0$
**repeat**
  $\bar{x}^{(t)} = x^{(t-1)} + \tau^{(t)} M^* \left(y - Mx^{(t-1)}\right)$;
  $I^{(t)} = \operatorname{supp}\left(\mathbb{T}_{s,\sigma}\left(\bar{x}^{(t)}\right)\right)$;
  $x^{(t)} = \operatorname*{argmin}_x \frac{1}{2}\|y - Mx\|^2 \quad \text{subject to} \quad \operatorname{supp}(x) \subseteq I^{(t)}$;
**until** *stopping criterion is met at* $t = t^*$
**output** : $(s, \sigma)$-sparse vector $x^{(t^*)}$

Here, $\tau^{(t)}$ is a suitably chosen step size. The original IHT algorithms use constant steps $\tau^{(t)} = 1$ for all $t$. Alternative strategies include backtracking as in the normalized iterative hard thresholding (NIHT) algorithm [6].

Faster convergence can be achieved with an adaption of the hard-thresholding pursuit [17] to hierarchical sparsity, the HiHTP [35, 36]. Compared to the HiIHT, the HiHTP algorithm uses the result of the thresholded gradient step as a proxy to guess the support of the correct solution in each step. Subsequently, a linear least-squares problem is solved on the support guess. The solution can be computed via pseudo-inverse or an approximate method. Notably, with this modification, if the algorithm finds the correct solution, it does this in a finite number of steps to the precision of the least-squares problem solver. The HiHTP algorithm is given as Algorithm 3.

The computational complexity of both algorithms, HiIHT and HiHTP, is typically dominated by the matrix vector multiplication with the measurement matrix $M$ and $M^*$, scaling in general as $O(mNn)$. If a fast matrix vector multiplication is available for the measurement matrix, this scaling can be significantly improved.

The additional least-square solution in the HiHTP algorithm contributes $O(s\sigma m^2)$ operations. For this reason, HiIHT can be faster per iteration than the HiHTP in certain parameter regimes. Note that the computational complexity, featuring the overall vector space dimension $Nn$ and the total sparsity $s\sigma$, is identical to the complexity of the original IHT and HTP algorithms.

Modifications using hierarchically sparse thresholding can also be applied to other compressed sensing algorithms such as the *CoSAMP* [32], the *Subspace Pursuit* [10], or *Orthogonal Matching Pursuit*, see e.g. Refs. [30, 46] and the references therein. Proximal operators of the convex relaxations of the problem

(1.1) can be calculated using soft-thresholding operations yielding a hierarchical version of the LASSO algorithms [20, 41–43]. Due to their similarity, the HiIHT and HiHTP algorithms inherit their convergence proofs and recovery guarantees with slight modifications from their non-hierarchical counterparts. To this end, we make use of the variant of the *restricted isometry property (RIP)* [8] adapted to hierarchically sparse signals.

**Definition 1.2 (Hierarchical Restricted Isometry Property (HiRIP))** Given a linear operator $M : \mathbb{K}^{Nn} \to \mathbb{K}^m$, we denote by $\delta_{s,\sigma}$ the smallest constant such that

$$(1 - \delta_{s,\sigma})\|x\|^2 \leq \|Mx\|^2 \leq (1 + \delta_{s,\sigma})\|x\|^2$$

holds for all $(s, \sigma)$-hierarchically sparse vectors $x \in \mathbb{K}^{Nn}$.

We will also refer to the standard $s$-sparse RIP constant $\delta_s$, defined analogously with the bounds holding for all $s$-sparse vectors. The standard RIP constant dominates the HiRIP constant as $\delta_{s\sigma} \geq \delta_{s,\sigma}$ since $\mathcal{S}_{s,\sigma}$ is a subset of the set of $s \cdot \sigma$-sparse vectors. But as we will see below, using the HiRIP allows for a considerably more fine-grained analysis, yielding improvements in the sampling complexity.

In terms of a HiRIP condition, we can guarantee a robust and stable convergence to the correct solution for the hierarchical hard-thresholding algorithms. To this end, given $x \in \mathbb{K}^{Nn}$ and a support set $\Omega \subset [N] \times [n]$, we denote by $x\rfloor_\Omega$ the projection of $x$ onto the subspace of $\mathbb{K}^{Nn}$ indicated by $\Omega$.

**Theorem 1.1 (Recovery Guarantee for HiIHT and HiHTP [35, 53])** *Suppose the measurement operator $M : \mathbb{K}^{Nn} \to \mathbb{K}^m$ satisfies the HiRIP condition*

$$\delta_{3s,2\sigma} < \delta_*,$$

*where $\delta_*$ is a threshold, equal to $1/\sqrt{3}$ for the HiHTP algorithm and equal to $\sqrt{2} - 1$ for the HiIHT algorithm. Then, for $x \in \mathbb{K}^{Nn}$, $e \in \mathbb{K}^m$, and $\Omega \subset [N] \times [n]$ an $(s, \sigma)$-hierarchically sparse support set, the sequence $(x^k)_k$ defined by HiIHT (Algorithm 2) or HiHTP (Algorithm 3), respectively, with $y = Mx\rfloor_\Omega + e$ satisfies, for any $k \geq 0$,*

$$\|x^k - x\rfloor_\Omega\| \leq \rho^k \|x^0 - x\rfloor_\Omega\| + \tau\|e\|,$$

*where the constants $\rho$ and $\tau$ depend on which algorithm is used: for HiIHT,*

$$\rho^{HiIHT} = \sqrt{3}\delta_{3s,2\sigma}, \quad \tau^{HiIHT} = \frac{2.18}{1 - \rho^{HiIHT}},$$

*whereas for HiHTP,*

$$\rho^{HiHTP} = \left( \frac{2\delta_{3s,2\sigma}}{1 - \delta^2_{(2s,2\sigma)}} \right)^{1/2}, \quad \tau^{HiHTP} = \frac{5.15}{1 - \rho^{HiHTP}}.$$

The theorem's proof follows closely along the lines of the standard proofs for HTP and IHT as found, e.g. in Refs. [7, 17, 19]. A detailed proof can be found in Refs. [35, 53], respectively.

## 1.4 Hierarchically Restricted Isometric Measurements

The results of the last section make it clear that the HiRIP property has the same role for hierarchically sparse recovery as the RIP takes on for sparse recovery. If we can prove that an operator $A$, for appropriate hi-sparsity levels $(s, \sigma)$, has the HiRIP, it is guaranteed that HiHTP can recover $x$ from the measurements $Ax$. In this chapter, we will establish the HiRIP for several families of measurement operators, using more and more specialized techniques.

### *1.4.1 Gaussian Operators*

Let us first discuss the HiRIP properties of the arguably most well-known random construction of a measurement operator: the Gaussian random matrix. A random matrix $A \in \mathbb{K}^{m \times n}$ is thereby said to be *Gaussian* if the entries are i.i.d. distributed according to the standard normal distribution $\mathcal{N}(0, 1)$.

It has become a folklore result (see e.g. Ref. [19, Ch.9]) that if $A$ is Gaussian, the renormalized matrix $\frac{1}{\sqrt{m}} A$ has the $s$-RIP with high probability if

$$m \gtrsim s \log\left(\frac{n}{s}\right),$$

where the notation $\gtrsim f(x)$ means greater than $C \cdot f(x)$, with $C$ an unspecified universal numerical constant. It is therefore natural to ask how large $m$ needs to be in order for $\frac{1}{\sqrt{m}} A$ to have the $(s, \sigma)$-HiRIP. Since $(s, \sigma)$-sparsity is more restrictive than $s\sigma$-sparsity, we surely will not need more than const. $\cdot\, s\sigma \cdot \log\left(\frac{s\sigma}{nN}\right)$ measurements. But is the threshold lower for the HiRIP? And if so, how much?

In fact, the framework of *model-based compressed sensing* [4] gives us a standard route to answer this question for the Gaussian ensemble. Let us sketch this route in some detail. First, one realizes that for any normalized fixed $x \in \mathbb{K}^N$, the random vector $\frac{1}{\sqrt{m}} Ax$ is also Gaussian and as such obeys the following *measure concentration inequality*:

$$\mathbb{P}\left(\left|\left\|\tfrac{1}{\sqrt{m}}Ax\right\|^2-1\right|>\delta\right)\leq 2\exp\left(-cm\delta^2\right),$$

where $c$ is a numerical constant. For a fixed vector $x\in\mathbb{K}^n$, $\frac{1}{\sqrt{m}}A$ preserves its norm with high probability.

Second, we generalize the almost isometric behaviour to hold for all vectors supported on a certain $k$-dimensional subspace $V$. To this end, we first establish that it suffices that the measurement operator acts almost isometrically on a so-called $\rho$-*net* for the intersection of the Euclidean unit ball with $V$. A $\rho$-net for a set $M$ is a set $N$ with the property that for any $q\in M$, there exists a $p\in N$ with $\|q-p\|_2<\rho$. It is not hard to construct a $\rho$-net for the normalized elements of $V$ with cardinality [19]

$$|N|\leq C_{\text{net}}\left(1+\tfrac{2}{\rho}\right)^k.$$

By choosing $\rho$ suitably and applying a union bound over the $\rho$-net, we obtain for any support $S$ with $|S|=k$

$$\mathbb{P}\left(\left|\left\|\tfrac{1}{\sqrt{m}}Ax\right\|^2-1\right|>\delta\quad\forall x:\operatorname{supp}(x)=S\right)\leq C\lambda^k\exp\left(-\tilde{c}m\delta^2\right),\tag{1.2}$$

where $C$, $\lambda$, and $\tilde{c}$ are universal numerical constants. With (1.2) at our disposal, it is only one step to establish an isometry property for $\frac{1}{\sqrt{m}}A\in\mathbb{R}^{m\times N\cdot n}$ for an entire union of subspaces such as structured sparse vectors. For instance, in order to get the $(s,\sigma)$-HiRIP, we need to take a union bound over all $(s,\sigma)$-sparse supports $S$. There are $\binom{N}{s}\binom{n}{\sigma}^s$ such supports. Therefore

$$\mathbb{P}\left(\left|\left\|\tfrac{1}{\sqrt{m}}Ax\right\|^2-1\right|>\delta\quad\forall\,(s,\sigma)\text{-sparse }x\right)\leq C\binom{N}{s}\binom{n}{\sigma}^s\lambda^{s\sigma}\exp\left(-\tilde{c}m\delta^2\right).$$

This probability is dominated by $\epsilon$, if

$$m\geq\tilde{c}^{-1}\delta^{-2}\log\left(C\binom{N}{s}\binom{n}{\sigma}^s\lambda^{s\sigma}\epsilon^{-1}\right).$$

Using the Stirling approximation $\binom{p}{k}\sim\left(\frac{p}{k}\right)^k$, we obtain the more readable condition

$$m\gtrsim\delta^{-2}\left(s\log\left(\tfrac{N}{s}\right)+s\sigma\log\left(\tfrac{n}{\sigma}\right)+\log\left(\tfrac{1}{\epsilon}\right)\right).$$

Let us state this as a theorem.

**Theorem 1.2 (HiRIP for Gaussian Matrices)** *Let $A\in\mathbb{K}^{m,n\cdot N}$ be random Gaussian. Then there is a universal numerical constant $C>0$ so that if*

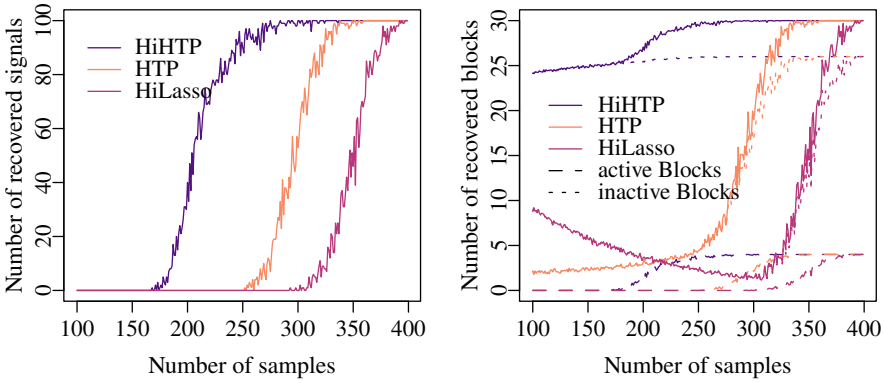



**Fig. 1.3** Left: the number of recovered signals from 100 noiseless Gaussian samples over the number of measurements $m$ for HTP, HiLasso, and HiHTP. The signals consist of $N = 30$ blocks of size $n = 100$ with $s = 4$ blocks having $\sigma = 20$ non-vanishing real entries. Right: the number of recovered blocks over the number of measurements $m$ for HTP and HiHTP. The dashed and dotted lines indicate the average number of correctly recovered zero and non-zero blocks, respectively. The solid lines show the total average number of recovered blocks. The signals consist of $N = 30$ blocks with $s = 4$ blocks having non-vanishing real entries. A signal or block is considered recovered if it deviates from the true signal by less than $10^{-5}$ in $\ell_2$-norm. ©2020 IEEE. Reprinted, with permission, from Ref. [35]

$$m \geq \frac{C}{\delta^2}\left(s\log\left(\frac{N}{s}\right) + s\sigma\log\left(\frac{n}{\sigma}\right) + \log\left(\frac{1}{\epsilon}\right)\right), \tag{1.3}$$

$\frac{1}{\sqrt{m}}A$ *has an* $(s,\sigma)$*-HiRIP constant* $\delta_{s,\sigma}(A) \leq \delta$ *with probability as least* $1-\epsilon$.

The difference of the condition (1.3) compared to the one needed to establish the standard RIP

$$m \geq \frac{C}{\delta^2}\left(s\sigma\log\left(\frac{Nn}{s\sigma}\right) + \log\left(\frac{1}{\epsilon}\right)\right) \tag{1.4}$$

is subtle. After all, both thresholds can be written as $s\sigma$ multiplied with logarithmic terms in the dimension of surrounding space. However, for certain parameter regimes, the difference is significant. Indeed, in the scenario that $N \gg n$, (1.3) can be much smaller than (1.4). This establishes that for Gaussian random matrices, hierarchical thresholding algorithms are theoretically expected to have an improved sampling complexity compared to their standard counterparts. Also in the non-asymptotic regime, one can observe an improved sample requirement in numerical simulation, see Fig. 1.3.

Note that the above discussion can be applied without problems to *sub-Gaussian* matrices. A matrix is sub-Gaussian if the entries $a_{i,j}$ are i.i.d. distributed according to a distribution that obeys $\mathbb{P}\left(|a_{i,j}| > t\right) \leq \alpha\exp\left(-\beta t^2\right)$ for some $\alpha, \beta > 0$.

### 1.4.2 Coherence Measures

The discussion in the last section very much relies on the random nature of the measurement operator. This is a common feature of compressed sensing-related theories—in order to obtain an optimal scaling, one practically has no choice other than to use a random construction. A possible route to still establish (non-optimal) RIP results for non-random matrices is to take a detour via so-called *coherence measures*. The simplest result is as follows [19, Prop 6.2]: if we define the *mutual coherence* of a matrix with normalized columns $a_i$ as

$$\mu(A) = \sup_{i,j} |\langle a_i, a_j \rangle|,$$

the RIP constants obey

$$\delta_s(A) \leq (s-1)\mu(A). \tag{1.5}$$

To establish analogous results for the HiRIP constants, we need to use coherence measures adapted to the block structure. Such measures have been introduced in Ref. [43] for the analysis of the HiLasso algorithm. To work with these coherence measures, it is convenient to introduce further notation to refer to the blocks of a vector individually. To this end, we use the Kronecker product of matrices in the convention

$$A \otimes B = \begin{pmatrix} a_{1,1}B & \dots & a_{1,N}B \\ \vdots & \ddots & \vdots \\ a_{m,1}B & \dots & a_{m,N}B \end{pmatrix},$$

where $a_{i,j}$ denotes the entries of $A$. The Kronecker product trivially also provides a Kronecker product on vectors $\mathbb{K}^N \times \mathbb{K}^n \to \mathbb{K}^{Nn}$ understood as $n \times 1$ and $N \times 1$ matrices, respectively. Using the basis $\{e_i\}_{i\in[N]}$, $(e_i)_j = \delta_{i,j}$ of $\mathbb{K}^N$, we can rewrite a blocked vector $x \in \mathbb{K}^{Nn}$ with blocks $x_i \in \mathbb{K}^n$, $i \in [N]$, as the sum of products $x = (x_1^\top, x_2^\top \dots, x_N^\top)^\top = \sum_{i\in[N]} e_i \otimes x_i$. The Kronecker product exemplifies the canonical vector space isomorphism of $\mathbb{K}^{Nn}$ with the tensor product space $\mathbb{K}^N \otimes \mathbb{K}^n$. Analogously, we identify the measurement matrices $A \in \mathbb{K}^{m\times N\cdot n}$ with linear operators $A : \mathbb{K}^N \otimes \mathbb{K}^n \to \mathbb{K}^m$. We refer to $A_i \in \mathbb{K}^{m\times n}$, $i \in [N]$, defined through $A_i(v) = A(e_i \otimes v)$, $v \in \mathbb{K}^n$, as the *block operators* of $A$. Now we introduce the specialized coherence measures.

**Definition 1.3 (Sub-coherence and Block Coherence)** Let $A : \mathbb{K}^N \otimes \mathbb{K}^n \to \mathbb{K}^m$ with block operators $A_i \in \mathbb{K}^{m\times n}$ and let $\{a_{i,j}\}_{j\in[n]}$ be the columns of the $i$th block operator. We define

1. The *sub-coherence* $\nu(A)$ of $A$ as the maximal mutual coherence of the block operators, i.e.

$$\nu(A) = \sup_i \mu(A_i) = \sup_i \sup_{j \neq k} |\langle a_{i,j}, a_{i,k} \rangle|.$$

2. The *sparse block coherence* $\mu_{\text{block}}^{\sigma\sigma}(A)$ of $A$ as

$$\mu_{\text{block}}^{\sigma\sigma}(A) = \sup_{i \neq j} \rho^{\sigma\sigma}(A_i^* A_j),$$

where $\rho^{\sigma\sigma}(B)$ denotes the $\sigma$*-sparse singular value* of a matrix $B \in \mathbb{K}^{N \times N}$,

$$\rho^{\sigma\sigma}(B) = \sup_{\substack{u,v \ \sigma\text{-sparse} \\ \|u\|=\|v\|=1}} |\langle u, Bv \rangle|.$$

Intuitively, $\nu(A)$ measures the coherence within each block, whereas $\mu_{\text{block}}^{\sigma\sigma}(A)$ measures the coherence between the blocks. Note that we have used a different normalization in the definition of the sparse block coherence compared to Ref. [43]. We can establish the following bounds on the HiRIP constants in terms of the coherence measures.

**Theorem 1.3 (HiRIP Through Coherence Bound)** *Let $A : \mathbb{K}^N \otimes \mathbb{K}^n \to \mathbb{K}^m$ be an operator with block operators $A_i$ and $s \in [N]$, $\sigma \in [n]$. It holds that*

1. $\sup_i \delta_\sigma(A_i) \leq \delta_{1,\sigma}(A)$ *and* $\mu_{\text{block}}^{\sigma\sigma}(A) \leq 2\delta_{2,\sigma}(A)$.
2. $\delta_{s,\sigma}(A) \leq \sup_i \delta_\sigma(A_i) + (s-1)\mu_{\text{block}}^{\sigma\sigma}(A)$.

*In addition, if all columns of the block operators $A_i$ are normalized, then*

$$\delta_{s,\sigma}(A) \leq (\sigma - 1)\nu(A) + (s-1)\mu_{\text{block}}^{\sigma\sigma}(A)\,.$$

***Proof***

1. Let $j \neq k$ and $x, y \in \mathbb{K}^n$ be $\sigma$-sparse normalized vectors. First, we have

$$|\|A_j x\|^2 - \|x\|^2| = |\|A(e_j \otimes x)\|^2 - \|e_j \otimes x\|^2| \leq \delta_{1,\sigma}(A),$$

since $e_j \otimes x$ is $(1, \sigma)$-sparse. This proves the first claim. For the second claim, we use the polarization identity to find

$$\langle A_j x, A_k y \rangle = \frac{1}{4} \sum_{\ell=0}^{3} i^\ell \left\| A_j x + i^\ell A_k y \right\|^2 = \frac{1}{4} \sum_{\ell=0}^{3} i^\ell \left\| A(e_j \otimes x + i^\ell e_k \otimes y) \right\|^2.$$

Since $e_j \otimes x$ and $e_k \otimes y$ have disjoint block supports, $e_j \otimes x + i^\ell e_k \otimes y$ are $(2, \sigma)$-sparse for all $\ell$. Hence,

$$\left| \frac{1}{4} \sum_{\ell=0}^{3} i^{\ell} \left\| A(e_j \otimes x + i^{\ell} e_k \otimes y) \right\|^2 - \frac{1}{4} \sum_{\ell=0}^{3} i^{\ell} \left\| e_j \otimes x + i^{\ell} e_k \otimes y \right\|^2 \right| \leq \delta_{2,\sigma} \cdot \frac{1}{4} \sum_{\ell=0}^{3} \left\| e_j \otimes x + i^{\ell} e_k \otimes y \right\|^2 .$$

Now we use that $\left\| e_j \otimes x + i^{\ell} e_k \otimes y \right\|^2 = 2$ for all $\ell$. This proves both that the final bound above equals $\frac{1}{2}\delta_{s,\sigma}$ and that $\sum_{\ell=0}^{3} i^{\ell} \left\| e_j \otimes x + i^{\ell} e_k \otimes y \right\|^2 = 0$, yielding the claim.

2. Let $x = \sum_i e_i \otimes x_i$ be an $(s, \sigma)$-sparse and normalized signal. There exists an $S \subseteq [N]$ with $|S| = s$ so that $x_i = 0$ for $i \notin S$. We have

$$\|Ax\|^2 = \sum_{i=1}^{N} \|A_i x_i\|^2 + \sum_{i \neq j} \langle A_i x_i, A_j x_j \rangle.$$

Each $x_i$ is $\sigma$-sparse and, thus, $|\|A_i x_i\|^2 - \|x_i\|^2| \leq \delta_\sigma(A_i)\|x_i\|^2$. Taking the sum over $i$ yields

$$\left| \|x\|^2 - \sum_{i=1}^{N} \|A_i x_i\|^2 \right| \leq \sup_i \delta_\sigma(A_i)\|x\|^2.$$

We still need to deal with the cross-block terms. Let the support of $x_i$ be denoted $S_i$, the orthogonal projection onto the space supported on $S_i$ with $P_{S_i}$, and define $V$ as the subspace with the same support as $x$. Consider the operator $C : V \to V$,

$$y = \sum_i e_i \otimes y_i \mapsto \sum_{i \in S} e_i \otimes P_{S_i} \left( \sum_{k \in S \setminus \{i\}} A_i^* A_k y_k \right).$$

We have

$$\sum_{i \neq j} \langle A_i x_i, A_j x_j \rangle = \langle x, Cx \rangle,$$

and $C$ is Hermitian. The latter implies that $|\langle x, Cx \rangle| \leq \lambda \|x\|^2$, where $\lambda$ is the magnitude of the largest eigenvalue of $C$. To estimate $\lambda$, let $v = \sum_i e_i \otimes v_i$ be a normalized eigenvector for $C$ and $i$ such that $\|v_i\|$ is maximal. We have $\lambda v_i = P_{S_i} \sum_{k \in S} A_i^* A_k v_k$, and consequently

$$\lambda \|v_i\|^2 = \langle v_i, P_{S_i} \sum_{k \in S} A_i^* A_k v_k \rangle = \sum_{\substack{k \in S \\ k \neq i}} \langle A_i v_i, A_k v_k \rangle$$

$$\leq \sum_{\substack{k\in S\\ k\neq i}} \mu^{\sigma\sigma}_{\text{block}}(A)\|v_i\|\|v_k\| \leq (s-1)\mu^{\sigma\sigma}_{\text{block}}(A)\|v_i\|^2.$$

In the second step, we have used that $v_i = P_{S_i} v_i$, since $v \in V$. In the penultimate step, we have used that $v_i$ and $v_k$ all are $\sigma$-sparse and that each index $k$ in the sum is different from $i$. In the final step, we have used the optimality of $i$. This proves that $\lambda \leq (s-1)\mu^{\sigma\sigma}_{\text{block}}(A)$ and therefore the claim.

Finally, the addition of the theorem follows from the claim with (1.5). □

The above result can be applied to construct a large family of operators that have suitably small HiRIP constants without exhibiting RIP in this regime. Consider $N$ pairwise orthogonal, $p$-dimensional subspaces of $\mathbb{K}^m$, and $E_i : \mathbb{K}^p \to \mathbb{K}^m$ isometric embeddings onto them. Let further $C \in \mathbb{K}^{p\times n}$ be a fixed matrix with $\delta_\sigma(C) = \delta < 1$. We consider the operator

$$A : \mathbb{K}^N \otimes \mathbb{K}^n \to \mathbb{K}^m,\ x \mapsto \sum_{i=1}^N E_i C x_i.$$

The block operators of $A$ are given by $E_i C$, $i \in [N]$, and each of them is compressively encoding $\mathbb{K}^n$ into one of the mutually orthogonal subspaces. Due to the fact that the $E_i$ are isometric, $\delta_\sigma(A_i) = \delta_\sigma(C)$ for each $i$. The pairwise orthogonality of the subspaces implies that $A_i^* A_j = 0$ for $i \neq j$, so that $\mu^{\sigma\sigma}_{\text{block}}(A) = 0$. Theorem 1.3 then implies that $\delta_{s,\sigma}(A) \leq \delta(C)$ for any $s$.

The above construction will generically not result in an operator with small $\delta_{s\sigma}(A)$. To this end, suppose that $s\sigma \leq n$ and $p \leq n - s\sigma$. Then, there exists an $s\sigma$-sparse $w \in \mathbb{K}^n$ with $Cw = 0$. Now the vector $x = (w, 0, \ldots, 0)$ is $s\sigma$-sparse, but $\left|\|Ax\|^2 - \|x\|^2\right| = \left|\|0\|^2 - \|x\|^2\right| = \|x\|^2$. We conclude that $\delta_{s\sigma}(A) \geq 1$.

A disadvantage of this construction is that necessarily $m \geq Np \geq N\sigma$. This is a considerably worse scaling than we found for Gaussian random matrices, which exhibit the HiRIP for $m \gtrsim s\sigma$ up to log-factors. The scaling in $N$ as opposed to the sparsity parameter on the block-level $s$ arises from the encoding into mutually orthogonal subspaces. The idea of 'mixing' block operators can, however, be driven a lot further to avoid this overhead as we will see in the next section.

### *1.4.3 Hierarchical Measurement Operators*

As we saw above, a measurement operator on $\mathbb{K}^N \otimes \mathbb{K}^n$ can always be thought of as a mixture of block operators, say

$$B(x) = \sum_{i=1}^N B_i x_i.$$

The inequalities in Theorem 1.3, part 1 imply that in order for $B$ to have a small HiRIP constant, we need each block operator to be well-conditioned, and in addition that the blocks are incoherent. What can we do when they are not?

Assume that instead of just observing $Bx$, we are allowed to sample a few different linear combinations of the vectors $B_i x_i$,

$$y = \left( \sum_{i=1}^{N} a_{i,j} B_i x_i \right)_{j \in [M]} = \sum_{i=1}^{N} a_i \otimes B_i x_i ,$$

with $a_i = (a_{j,i})_{j \in [M]} \in \mathbb{K}^M$. Can this make recovery easier? Let us define such measurement operators that act hierarchically on the block structure of the vectors as *hierarchical measurement operators*.

**Definition 1.4 (Hierarchical Measurement Operators)** Let $A \in \mathbb{K}^{M,N}$ and $B_i \in \mathbb{K}^{m,n}$, $i = 1, \ldots, N$, be given and denote the $i$th column of $A$ by $a_i$. We call the operator

$$\mathcal{H} : \mathbb{K}^N \otimes \mathbb{K}^n \to \mathbb{K}^M \otimes \mathbb{K}^m, \qquad x \mapsto \sum_{i=1}^{N} a_i \otimes B_i x_i ,$$

the *hierarchical measurement operator defined by $A$ and $(B_i)_{i \in [N]}$*.

The structure and naming of hierarchical operators makes it easy to believe that they are an excellent fit for hierarchically sparse recovery. They are, however, by no means only of academic interest. We will discuss this more thoroughly in Sect. 1.6. For now, the practical interest might already become apparent by noting that an important special case of hierarchical measurement operators is the following: in the case of $B_i = B$ being equal, the hierarchical operator is the same as the *Kronecker product* $A \otimes B$ of the matrices $A$ and $B$. How do the hierarchical isometry constants of $\mathcal{H}$ relate to the ones of $A$ and the $B_i$s? In order to discuss this question, we begin by proving the following lemma.

**Lemma 1.1 (RIP Implies Nuclear Norm Isometry)** *Let $X \in \mathbb{K}^{N \times N}$ have the property that for some sets $S$ and $\overline{S}$ of cardinality $s$, $X_{i,j} = 0$ if either $i \notin S$ or $j \notin \overline{S}$.*

*1. If $X$ is positive-definite Hermitian, which in particular implies $S = \overline{S}$,*

$$\left| \langle A^* A, X \rangle - \|X\|_* \right| \leq \delta_s(A) \|X\|_* .$$

*2. If $S$ and $\overline{S}$ are disjoint,*

$$|\langle A^* A, X \rangle| \leq \delta_{2s}(A) \|X\|_* .$$

*Here, $\|X\|_*$ denotes the nuclear norm, also known as the trace norm, of $X$, i.e. the sum of its singular values.*

***Proof*** Consider a singular value decomposition of $X$, $X = \sum_{i=1}^N \sigma_i v_i u_i^*$. We have $\langle A^*A, X\rangle = \sum_{i=1}^N \sigma_i \langle Au_i, Av_i\rangle$. Due to the assumption, for all $i$ with $\sigma_i \neq 0$, $\mathrm{supp}(v_i) \subset S$ and $\mathrm{supp}(v_i) \subset \overline{S}$.

1. For $X$ positive-definite, the $\sigma_i$ are the eigenvalues of $X$, and $u_i = v_i$. Since each $u_i$ is $s$-sparse, it holds that

$$\left|\langle A^*A, X\rangle - \|X\|_*\right| \leq \sum_{i=1}^N \sigma_i \left|\langle Au_i, Au_i\rangle - 1\right| \leq \sum_{i=1}^N \sigma_i \cdot \delta_s(A) = \delta_s(A)\|X\|_*.$$

2. Ref. [19, Prop. 6.3] states that since the supports of $u_i$ and $v_i$ are disjoint, we have $|\langle Au_i, Av_i\rangle| \leq \delta_{2s}(A)$. This in turn implies

$$|\langle A^*A, X\rangle| \leq \sum_{i=1}^N \sigma_i |\langle Au_i, Av_i\rangle| \leq \sum_{i=1}^N \sigma_i \delta_{2s}(A) = \delta_{2s}(A)\|X\|_*\,.$$

□

We now prove that $\mathcal{H}$ inherits the HiRIP from the RIP of its constituent matrices, in that $\delta_{s,\sigma}(\mathcal{H})$ can be bounded in terms of $\delta_s(A)$ and the constants $\delta_\sigma(B_i)$.

**Theorem 1.4 (Hierarchically Inherited HiRIP)** *Let $\mathcal{H}$ be the hierarchical operator defined by $A$ and $(B_i)_{i\in[N]}$. We have for $s, \sigma$ arbitrary*

$$\delta_{s,\sigma}(\mathcal{H}) \leq \delta_s(A) + \sup_i \delta_\sigma(B_i) + \delta_s(A) \cdot \sup_i \delta_\sigma(B_i).$$

***Proof*** Let $x$ be normalized and $(s,\sigma)$-sparse and $S$ such that $a_i = 0$ for $i \notin S$. We have

$$\|\mathcal{H}(x)\| = \sum_{i,j=1}^N \langle a_i \otimes (B_i x_i), a_j \otimes (B_j x_j)\rangle = \sum_{i,j=1}^N \langle a_i, a_j\rangle \langle B_i x_i, B_j x_j\rangle = \langle A^*A, G\rangle,$$

where $G \in \mathbb{K}^{N\times N}$ denotes the matrix with non-vanishing entries $G_{i,j} = \langle B_i x_i, B_j x_j\rangle$ for $i \in S$ and $j \in S$. By Lemma 1.1, part 1,

$$|\langle A^*A, G\rangle - \|G\|_*| \leq \delta_s(A)\|G\|_*. \tag{1.6}$$

It remains to estimate $\|G\|_*$. In order to do this, consider the operator $M : \mathbb{K}^{|S|} \to \mathbb{K}^m$, $c \mapsto \sum_{i\in S} c_i B_i x_i$. By construction, $G = M^*M$, and therefore, $\|G\|_* = \|M\|^2 = \sum_{i\in S} \|B_i x_i\|^2$, where $\|\cdot\|$ here refers to the Frobenius norm. Consequently,

$$\left| \|G\|_* - \|x\|^2 \right| \leq \sum_{i \in S} \left| \|B_i x_i\|^2 - \|x_i\|^2 \right| \leq \sum_{i \in S} \delta_\sigma(B_i) \|x_i\|^2. \tag{1.7}$$

Combining (1.6) and (1.7), we obtain

$$\begin{aligned} \left| \langle A^* A, G \rangle - \|x\|^2 \right| &\leq \left| \langle A^* A, G \rangle - \|G\|_* \right| + \left| \|G\|_* - \|x\|^2 \right| \\ &\leq \delta_s(A) \left( 1 + \sup_i \delta_\sigma(B_i) \right) \|x\|^2 + \sup_i \delta_\sigma(B_i) \|x\|^2, \end{aligned}$$

which proves the claim. □

The theorem shows that hierarchical operators are a rich class of operators which much more often have the HiRIP than the RIP. To make this precise, we take a look at the special case of Kronecker products $A \otimes B$. Theorem 1.4 implies that if $\delta_s(A)$ and $\delta_\sigma(B)$ are small, $\delta_{s,\sigma}(A \otimes B)$ is also small. This is in stark contrast to the RIP of Kronecker products. Indeed, Ref. [25] derived that

$$\delta_s(A \otimes B) \geq \max(\delta_s(A), \delta_s(B)).$$

That is, in order for $A \otimes B$ to have the $s$-RIP (nota bene, not the $s\sigma$-RIP), *both* $A$ and $B$ must have it. This obstacle leads to demanding performance bounds in applications [40].

The total number of measurements measured by a hierarchical operator is equal to $mM$. Together with the classical results on the RIP of Gaussian operators, the theorem implies that by choosing $A$ and $B$ Gaussian, we can hence build hierarchical operators having the $(s, \sigma)$-HiRIP using only

$$\text{const} \cdot s\sigma \log\left(\frac{n}{\sigma}\right) \log\left(\frac{N}{s}\right)$$

many measurements. This scaling is up to log-factors identical to the result Eq. (1.3) we established for fully Gaussian matrices. This is noteworthy, since while fully Gaussian matrix consists of $MN \cdot mn$ independent parameters, a Kronecker product $A \otimes B$ only has $MN + mn$. This constitutes a considerable de-randomization of the measurements, which can be e.g. exploited to reduce the storage complexity or to speed up calculations. We refer to Refs. [34, 35] for an extended discussion and an alternative direct proof of HiRIP for Kronecker product measurements.

Theorem 1.4 tells us that operators with small RIP constants can be combined to obtain an operator with a small HiRIP constant. We now take a look at the contrary question: To what extend are small RIP constants of the constituent operators required to bound the HiRIP constants of the hierarchical measurement operator?

In order to get a simple formulation of our first result, let us first note that there is an ambiguity in the definition of a hierarchical measurement operator. We can always simultaneously rescale $a_i$ and $B_i$ since $a_i \otimes B_i = (\lambda a_i) \otimes (\lambda^{-1} B_i)$. We

may thus w.l.o.g. assume that $\|a_i\| = 1$ for all $i$. Under this assumption, a small $(s, \sigma)$-HiRIP constant of $\mathcal{H}$ indeed implies small $\sigma$-RIP constants of all $B_i$.

**Proposition 1.1 ($\sigma$-RIP Bound from $(s, \sigma)$-HiRIP)** *Let $\mathcal{H}$ be a hierarchical measurement operator given by $A$ and $(B_i)_{i\in[N]}$. Assume that the columns of $A$ fulfil $\|a_i\| = 1$ for all $i$. Then, it holds that*

$$\sup_i \delta_\sigma(B_i) \leq \delta_{1,\sigma}(\mathcal{H}).$$

***Proof*** The $i$th block operator $\mathcal{H}_i$ of $\mathcal{H}$ is given by $a_i \otimes B_i \in \mathbb{K}^{Mm\times N}$. The normalization implies that $\|(a_i \otimes B_i)x\|^2 = \|a_i\|^2 \cdot \|B_i x\|^2 = \|B_i x\|^2$ for each $x \in \mathbb{K}^N$. Thus, $\delta_\sigma(B_i) = \delta_\sigma(\mathcal{H}_i)$, and the result follows from Theorem 1.3, part 1. □

The above result in essence states that for $\mathcal{H}$ to have the $(s, \sigma)$-HiRIP, it is necessary that all $B_i$ have the corresponding $\sigma$-RIP. Intriguingly, for the RIP requirement of $A$, the situation is very different. Indeed, if the $B_i$ are mapping into incoherent subspaces, $A$ does not need to have the RIP. The precise result is as follows.

**Theorem 1.5 (HiRIP with Block Incoherence)** *For a family $(B_i)_{i\in[N]}$, define the operator*

$$\mathcal{B} : \mathbb{K}^N \otimes \mathbb{K}^n \to \mathbb{K}^m, \quad x \mapsto \sum_{i=1}^N B_i x_i \,.$$

*Let $A \in \mathbb{K}^{M\times N}$ and natural numbers $s$, $\sigma$, and $t$ be given. The hierarchical operator $\mathcal{H}$ given by $A$ and $(B_i)_{i\in[N]}$ fulfils*

$$\delta_{ts,\sigma}(\mathcal{H}) \leq \sup_i \delta_\sigma(B_i) + \delta_s(A) \cdot \sup_i \delta_\sigma(B_i) + t\sqrt{s} \cdot \delta_{2s}(A) \cdot \mu_{\text{block}}^{(2\sigma,2\sigma)}(\mathcal{B}) \,.$$

***Proof*** Let $x = \sum_i e_i \otimes x_i$ be a $(ts, \sigma)$-sparse, normalized vector and $S \subset [N]$ be such that $x_i = 0$ for $i \notin S$. We may subdivide $S$ into $t$ disjoint sets $S_1, \ldots, S_t$ with cardinality $s$ each. For each pair $(k, \ell) \in [t] \times [t]$, we define a matrix $G^{k,\ell} \in \mathbb{K}^{N\times N}$ with non-vanishing entries $G^{k,\ell}_{i,j} = \langle B_i x_i, B_j x_j\rangle$ for $i \in S_k$ and $j \in S_\ell$. We may use the same reasoning as in the proof of Theorem 1.4 to argue that

$$\|\mathcal{H}(x)\|^2 = \sum_{k=1}^{\langle} \langle A^*A, G^{k,k}\rangle + \sum_{k\neq\ell} \langle A^*A, G^{k,\ell}\rangle \,.$$

Now, each matrix $G^{k,\ell}$ fulfils the assumption of Lemma 1.1, part 1 for $k = \ell$ and Lemma 1.1, part 2 for $k \neq \ell$. Hence,

$$\left| \|\mathcal{H}(x)\|^2 - \sum_{k=1}^{t} \left\| G^{k,k} \right\|_* \right| \leq \delta_s(A) \cdot \sum_{k=1}^{N} \left\| G^{k,k} \right\|_* + \delta_{2s}(A) \cdot \sum_{k \neq \ell} \left\| G^{k,\ell} \right\|_* .$$

Still in analogy to the proof of Theorem 1.4, we find that $\left| \left\| G^{k,k} \right\|_* - \|x_k\|^2 \right| \leq \sup_i \delta_\sigma(B_i) \|x_k\|^2$, and consequently

$$\left| \mathcal{H}(x) - \|x\|^2 \right| \leq \delta_s(A) \left( 1 + \sup_i \delta_\sigma(B_i) \right) + \delta_{2s}(A) \cdot \sum_{k \neq \ell} \left\| G^{k,\ell} \right\|_* .$$

It remains to bound the terms with $k \neq l$. First, let us note that, since $G^{k,\ell}$ has rank at most $s$, $\|G^{k,\ell}\|_* \leq \sqrt{s} \|G^{k,\ell}\|$. We now use the definition of the intra-block coherence to argue that

$$\left\| G^{k,\ell} \right\| = \sqrt{\sum_{i \in S_k, j \in S_\ell} |\langle B_i x_i, B_j x_j \rangle|^2} \leq \mu_{\text{block}}^{(2\sigma,2\sigma)} \sqrt{\sum_{i \in S_k, j \in S_\ell} \|x_i\|^2 \cdot \|x_j\|^2} .$$

Finally with

$$\sum_{k \neq \ell} \sqrt{\sum_{i \in S_k,} \|x_i\|^2} \cdot \sqrt{\sum_{j \in S_\ell} \|x_j\|^2} \leq \left( \sum_k \sqrt{\sum_{i \in S_k} \|x_i\|^2} \right)^2 \leq t \|x\|^2 ,$$

where we have used the Cauchy–Schwarz inequality in the final step, the claim follows. □

Note that the above result shows that $A$ does not need to have the $ts$-RIP in order for the hierarchical operator to exhibit the corresponding HiRIP. We may in particular choose $t = N/s$ and obtain an operator that acts isometrically on any vector with sparse blocks. In terms of sample complexity, the above result is still a bit opaque. By making a particular choice of $t$ and using the methods of Gaussian random matrices discussed in Sect. 1.4.1, one can derive the following result (see Ref. [16] for a proof).

**Proposition 1.2 (Sample Complexity)** *Let $(B_i)_i$ and $\mathcal{B}$ be as in Theorem 1.4.3. Assume that*

$$\left( t \mu_{\text{block}}^{(2\sigma,2\sigma)}(\mathcal{B}) \right)^2 \leq \frac{N}{\log(N)},$$

*and choose $A \in \mathbb{K}^{M \times N}$ as a Gaussian matrix. Let $\delta, \epsilon > 0$. Provided that*

$$M \geq C \left( \left( t \mu_{\text{block}}^{(2\sigma,2\sigma)}(\mathcal{B}) \right)^2 \cdot \frac{1}{\delta^2} \log \left( \frac{N \left(1 + \sup_i \delta_{\sigma(B_i)}\right)^2}{\left( t \mu_{\text{block}}^{(2\sigma,2\sigma)}(\mathcal{B}) \right)^2} \right) + \log\left(\frac{1}{\epsilon}\right) \right),$$

*where $C$ is a universal numerical constant, the hierarchical measurement operator $\mathcal{H}$ defined by $A$ and $(B_i)_{i \in N}$ obeys*

$$\delta_{t,\sigma}(\mathcal{H}) \leq \delta + \sup_i \delta_\sigma(B_i)$$

*with a probability at least $1 - \epsilon$.*

This proposition shows that if $\mu_{\text{block}}^{(2\sigma,2\sigma)}(\mathcal{B})$ is small enough, the number of 'Gaussian linear combinations' we take with $A$ does not have to grow linearly in $t$ in order to establish a $(t, \sigma)$-RIP—instead, only $(t \mu_{\text{block}}^{(2\sigma,2\sigma)}(\mathcal{B}))^2$ is needed.

The square dependence here on $(t \mu^{(2\sigma,2\sigma)})$ is of course inferior compared to the linear dependence of the sparsity we can achieve with the help of Theorem 1.4. It is unclear whether this is merely an artefact of the proof.

These results end our discussion of the hierarchical operators and with that our theoretical results on hierarchical restricted isometry properties.

## 1.5 Sparse De-mixing of Low-Rank Matrices

Generally, hierarchically sparse vectors arise from recursively assuming nested groupings of the vector entries to be sparsely non-vanishing. Another generalization of hierarchically structured vectors arises when we replace the sparsity assumption with another structure assumption such as a low rank when suitably reshape. One of the simplest of such examples is the de-mixing of a sparse sum of low-rank matrices from linear measurements. For $i \in [N]$, let $\mathcal{A}_i : \mathbb{K}^{n \times n} \to \mathbb{K}^m$ be linear maps and $\rho_i \in \mathbb{K}^{n \times n}$ be matrices of rank at most $r$. The problem of *de-mixing low-rank matrices* is to reconstruct the matrices $\rho_i$ given data of the form

$$y = \sum_{i=1}^{N} \mathcal{A}_i(\rho_i) \,.$$

A further structure assumption might be that out of the $N$ matrices $\rho_i$ actually only a number of $s$ are non-vanishing, giving rise to the problem of de-mixing a sparse sum. We can straightforwardly cast the problem as the reconstruction problem of a hierarchically structured vector. To this end, we set $X = \sum_{i=1}^{N} e_i \otimes \rho_i$. We can regard $X$ as a 'vector' in $\mathbb{K}^{Nn \times n}$ of matrix-valued blocks of rank-$r$ and at most $s$ vanishing blocks.

**Algorithm 4:** SDT algorithm

**input** : Data $y$, measurement $\mathcal{A}$, sparsity $s$ and rank $r$ of signal
**initialize**: $X^0 = 0$.
**repeat**
  Calculate step-widths $\mu^l$
  $X^{l+1} = \bar{\mathbb{T}}_{s,r}\left(X^l + \operatorname{diag}(\mu^l) P_{\mathcal{T}_{X^l}}\left(\mathcal{A}^*\left(y - \mathcal{A}(X^l)\right)\right)\right)$
**until** *stopping criterion is met at* $l = l^*$
**output** : Recovered signal $X^{l^*}$

Compared to $(s, \sigma)$-sparse vectors, we have replaced the non-vanishing $\sigma$-sparse blocks by low-rank matrices. The de-mixing problem of a sparse sum of low-rank matrices then is the task to reconstruct such a hierarchically (block) sparse, (block-wise) low-rank vector $X$ from linear measurements.

The principle strategy of hierarchical hard thresholding of Sect. 1.3 carries over to hierarchically sparse, low-rank vectors. The projection onto the set of rank-$r$ matrices is given by the hard thresholding of the singular values. Let $\rho \in \mathbb{K}^{n\times n}$ have singular value decomposition $U \operatorname{diag}(\Sigma) V^*$ with a vector of singular values $\Sigma \in \mathbb{K}^n$. We define

$$P_r(\rho) = U \operatorname{diag}(\mathbb{T}_r(\Sigma)) V^* .$$

Basically, replacing the application of $\mathbb{T}_\sigma$ in the hierarchically thresholding Algorithm 1 yields a projection onto hierarchically sparse, low-rank vectors which we will refer to as $\bar{\mathbb{T}}_{s,r}$.

Modifying the projective gradient descent of the HiIHT algorithm with this projection yields the so-called *sparse de-mixing thresholding* (SDT) algorithms, Algorithm 4 [38]. In contrast to the structure of a union of subspaces of sparse vectors, the set of rank $r$ matrices constitutes an embedded differential manifold in the linear vector space of all matrices. The geometrical structure can be exploited in iterative hard-thresholding algorithms by projecting the gradient of the embedding space in the descent step onto the tangent space of the manifold at the current iterate [1, 47, 49]. At point $\rho$, the tangent space of the manifold of rank-$r$ matrices is the linear span of the set of matrices that have the same row or column space as $\rho$ [1]. For a hierarchically sparse, low-rank vector $X = \sum_{i=1}^N e_i \otimes \rho_i$, we use the projection onto the tangent space for each block. We denote by $P_{V_i}$ and $P_{U_i}$ the projection onto the row and column space of $\rho_i$, respectively. For $\rho_i$ vanishing, we set the projections to be the identity. We define $P_{\mathcal{T}_X} : \mathbb{K}^{Nn\times n} \to \mathbb{K}^{Nn\times n}$ as $G = \sum_{i=1}^N e_i \otimes g_i \mapsto \sum_{i=1}^N e_i \otimes [g_i - (\mathrm{Id} - P_{U_i}) g_i (\mathrm{Id} - P_{V_i})]$. The particularity of the SDT algorithm is that we allow for a different step size for each matrix block. We refer to Ref. [38] for more details on the algorithm and Ref. [50] for an implementation. The SDT algorithm without the sparse thresholding operation to determine the block support coincides with the algorithm proposed in Ref. [45].

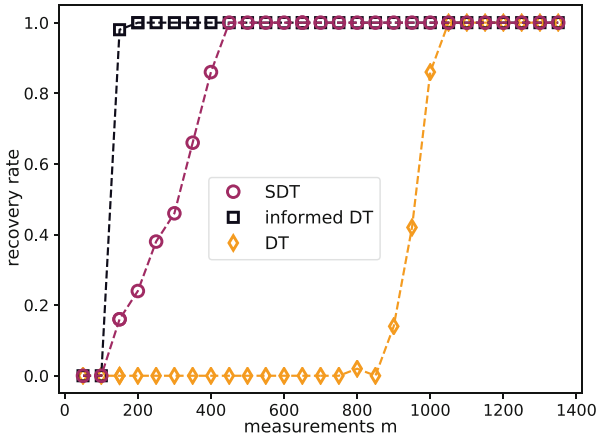



**Fig. 1.4** The figure (taken from Ref. [38]) displays the recovery rate for the SDT in different variants for different values of $m$ for random Gaussian measurements. *DT* refers to the SDT algorithms without the sparsity constraint (SD) and *informed DT* to the SDT algorithm restricted to the correct support. A signal is considered successfully recovered, if the algorithm output deviates from the true signal by less than $10^{-3}$ in Frobenius norm. Each point is averaged over 50 iterations and signal instances with $r = 1$, $n = 16$, $N = 10$, and $s = 3$. One observes nearly coinciding recovery performances for the informed DT and the SDT algorithm. In comparison, the DT algorithm requires significantly more samples for recovery

Following the blueprint of model-based compressed sensing, one can also establish a recovery guarantee based on a RIP condition custom-tailored to the hierarchical structure at hand. For random Gaussian measurement ensembles, this gives rise to a sampling complexity of

$$\delta^{-2}[s\log(N/s) + (2n+1)rs\log\tfrac{1}{\delta}]$$

to guarantee the correct recovery of $X \in \mathbb{K}^{Nn\times n}$ with at most $s$ non-vanishing blocks of rank $r$ [38, Theorem 6]. Many results derived in Sect. 1.4 that establish the HiRIP for hierarchically sparse vectors for different measurement ensembles can be generalized to hierarchically sparse, low-rank vectors. This allows one to guarantee recovery by the SDT algorithm for a large class of measurement ensembles.

Compared to an algorithm that does not exploit the sparsity of the de-mixing problem, the SDT algorithm can exhibit a significant improvement in the sampling complexity in relevant parameter regimes, Fig. 1.4.

Hierarchically sparse, low-rank vectors certainly constitute another important class of hierarchically structured signals as it encodes the de-mixing problem of a sparse sum of low-rank matrices. The theme of hierarchically combining low-rank and sparse structure assumptions in nested grouping of entries gives rise to a plethora of structures all of which can be efficiently reconstructed using recursive combinations of the hierarchical thresholding method introduced above.

## 1.6 Selected Applications

### *1.6.1 Channel Estimation in Mobile Communication*

In mobile communication, a lot of users are simultaneously communicating with a base station through electromagnetic waves. Let us model the messages a user wants to transmit with a sequence $c \in \mathbb{K}^n$. To send this message, the user must first translate the message to a wave. A popular scheme for this is so-called *OFDM (Orthogonal Frequency-Division Multiplexing)*. This scheme can be imagined as each $c_k$ giving rise to a complex exponential, a so-called tone, $b(\omega) = [1, e^{-i\omega t_1}, \ldots e^{-i\omega t_{n-1}}] \in \mathbb{K}^{1\times n}$, where $\omega$ is the frequency and $t_1, \ldots, t_{n-1}$ are some discretization times. In OFDM, a fixed grid of the form $\omega_k = 2\pi k\overline{\omega}, k \in [n]$, is used, where $\overline{\omega}$ is the normalized frequency. Mathematically, this corresponds to applying the discrete Fourier transform to $c$.

As the electromagnetic waves travel from the user to the base station, they scatter on random features, e.g. buildings and trees, in the environment. This scattering causes random phase and amplitude shifts, modelled by so-called complex gains $\rho_p$. It also means that a single transmission results in several incoming wave-fronts, each with a different angle of arrival. This situation can be utilized if the base station has several antennas arranged in an array: when the wave-front arrives at the antenna array, the wave-front travels slightly different distances before arriving at each antenna, i.e. if a '1' arrives at antenna 0, antenna $k$ will receive an '$a_k(\theta)$', where $\theta$ denotes the angle of the wave-front. Here, $a = [a_0, \ldots, a_{n-1}] : [-\pi, \pi] \to \mathbb{K}^{1\times n}$ is a function, often referred to as the *antenna manifold* in the communication literature. For the popular *uniform linear array (ULA)*, in which the antennas are placed at a uniform separation $d$ along a straight line, the antenna manifold is after a change of variables $u = d\sin(\theta)$ given by

$$a(u) = \left[1, e^{2\pi diu}, e^{4\pi diu}, \ldots, e^{2(n-1)\pi diu}\right].$$

The parameter $u$ actually takes on values in the entirety of $[-d, d]$, but let us for now assume that it lies on some grid $\{-\frac{d}{2N}, \ldots, \frac{d}{2N}\}$.

Combining these two models, we see that for a specific user, all transmitted signals result in a collective measurement of the form $\sum_{\ell=1}^{L} \rho_p a(u_p)^* \langle b(\omega_p)^*, c\rangle$ where $(\omega_p, u_p)$ is given by the delay and angle of the $k$th wave-front. The communication is thus characterized by the *channel matrix* [9]

$$H = \sum_{p=1}^{L} \rho_p a(u_p)^* b(\omega_p) \in \mathbb{K}^{N\times n}.$$

Once we know $H$, the base station can easily decode any number of sent messages. Note that as long as the environment and the position of the user do not change drastically, $H$ is expected to stay roughly constant.

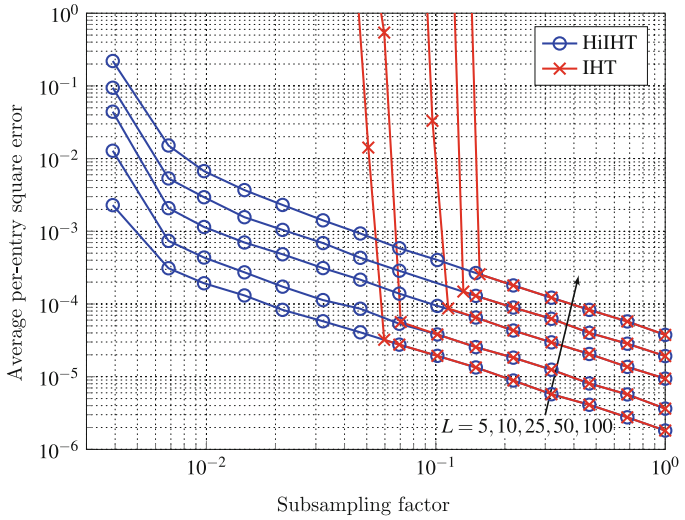



**Fig. 1.5** Comparison of HiIHT and IHT performances for channel matrix reconstruction. ©2019 IEEE. Reprinted, with permission, from Ref. [53]

Now suppose that we are only given a low-dimensional sample of $H$. To be concrete, define sub-sampling operators $P_u \in \mathbb{K}^{M\times N}$, $P_\omega \in \mathbb{K}^{m\times n}$ in angle and delay, and assume that we only observe $P_u H P_\omega^\top$. Can we still recover the entire matrix? To do this, we may utilize that, according to the above discussion, it has a sparse representation in the delay-angle domain. Indeed, defining $A = [e^{2k\pi i u_j}]_{k,j\in[N]} \in \mathbb{K}^{N\times N}$ and $B = [e^{-it_k\omega_\ell}]_{k,\ell\in[n]} \in \mathbb{K}^{n\times n}$, we get

$$P_u H P_\omega^\top = P_u A \left( \sum_{p\in[L]} \rho_p e_{u_{j_p}} \otimes e_{\omega_{\ell_p}} \right) B^* P_\omega^\top = (P_u A \otimes P_\omega B) X,$$

with $X = \sum_{p=1}^{L} \rho_p e_{u_{j_p}} \otimes e_{\omega_{\ell_p}}$. Note that $X$ is not only sparse but also hierarchically sparse: only a few angle blocks are active, and for each such angle, only a few delays $\omega_k$ are utilized and vice versa. In fact, it is a reasonable assumption that the angles for the $L$ paths are distinct, leading to a $(1, L)$-sparse ground truth. We further observe that sampled $H$ is a Kronecker product measurement of $X$, where the terms of the Kronecker product are sub-sampled Fourier matrices. Thus, the results of Sect. 1.4.3 imply that the recovery indeed is possible and provide an explicit sampling complexity.

In Fig. 1.5, the performance of HiIHT and IHT is compared for $m = n = 256$ and $N = 1024$. We generate data synthetically and inject the measurement with Gaussian noise of an SNR of 10dB. The recovery quality is measured in terms of the mean per-entry square error $\frac{1}{nN}\|H - \hat{H}\|^2$ between the actual channel matrix $H$ and the estimate $\hat{H}$. This error is plotted against the sub-sampling factor $M/N$

for different values of $L$. We see that HiIHT handles a small sub-sampling factor considerably better than IHT. Indeed, only accessing one percent of the available antennas is enough to achieve reasonable performance with HiIHT, whereas IHT fails when less than about 10 percent of the antennas are utilized.

The communication setting presented here can be extended in several directions: first, we may drop the assumptions on the delays and angles to be on a grid—in the off-the-grid case, the vector $X$ is arguably still approximately sparse. Second, we can model the case of multiple users by adding a third level to the hierarchical signal. On this level, sparsity naturally emerges assuming a sporadic user activity. We refer to Ref. [53] for details.

### *1.6.2 Secure Massive Access*

With the rise of new communication technologies such as the Internet of Things (IoT) and Tactile Internet (TI), the amount of devices virtually explodes, and with it the amount of sensitive information gathered from various sensors and transmitted over the air. This development poses significant challenges on the security of communication channels and demands for new physical layers of security. In particular, it calls for fast and scalable low-overhead security schemes suitable for the frequent burst of spontaneous communication between low-complexity devices with a base station. Here, we use the hierarchical measurement framework to design a secure massive access procedure based on blind deconvolution, see also the discussion on bisparse structures in Sect. 1.2. More details can be found in Ref. [52].

A base station sends out known pilots to enable all *user equipments (UEs)* to measure the channel between the station and the UE. The channel is here modelled as a filter in $\mathbb{K}^N$, where $N$ is the length of the delay period. For each transmitting UE $p \in [N_d]$ and receiving base station antenna $q \in [N_r]$, there is one filter

$$h_{p,q} = (h_{p,q,1}, \dots, h_{p,q,i}, \dots, h_{p,q,N}) \in \mathbb{K}^N.$$

The concrete appearance of the filters is again determined by delays caused by reflections on random physical features in the environment. Therefore, it is reasonable to assume that each $h_{p,q}$ is sparse, and, for fixed UE $q$, all channels $h_{p,q}$ for $p = 1, \dots, N_t$ share the same sparsity pattern.

As in the previous section, the UE transmits their sequences $c_p \in \mathbb{K}^E$ by first linearly encoding them into signals $x_p = B_p c_p$ using a codebook $B_p \in \mathbb{K}^{N \times E}$ and then sending them over the channel. In an IoT scenario, the messages typically are very short, so that it can be assumed that they can be encrypted as sparse sequences $c_p$. During transmissions, these are convolved with the channel vectors, so that each of the base station's antennas receives a superposition of the UEs' signals,

$$y_q = \sum_{p=1}^{N_r} h_{p,q} \circledast (B_p c_p) + z_q$$

with $q = 1, \dots, N_t$ and $\circledast$ denoting the circular convolution. We may now *lift* [29] the bilinear operation $(c_p, h_{p,q}) \to h_{p,q} \circledast B_p c_p$ to a linear operation $\mathrm{conv}_p : \mathbb{K}^{E \times N} \to \mathbb{K}^N$ on the matrix $b_p h_{p,q}^\top \in \mathbb{K}^{E \times N}$ resulting in

$$y_q = \sum_{p=1}^{N_r} \mathrm{conv}_p(b_p h_{p,q}^\top) + z_q \,. \tag{1.8}$$

We observe that the channel estimation task at the base station becomes the problem of simultaneously performing a blind deconvolution and de-mixing, naturally formalized as the linear reconstruction of a signal

$$X_q = (b_1 h_{1,q}^\top, \dots, b_{N_d} h_{N_d,q}^\top) \in (\mathbb{K}^{E \times N})^{N_d} \sim \mathbb{K}^{N_d \cdot E \cdot N}.$$

The signal further exhibits the following structure: our assumptions of $\sigma$-sparse channels and $s$-sparse messages imply that the matrices $b_p h_{p,q}^\top$ are all $(s, \sigma)$-bisparse. As discussed in Sect. 1.2, we may relax this to simple hierarchical $(s, \sigma)$-sparsity. Additionally assuming a sparse user activity at a given time, i.e. $b_p \neq 0$ only for $\mu$ users, the vector $X_q$ is a three-level $(s, \sigma, \mu)$-sparse vector. Note that the operator $\mathrm{conv}_p$ has a structure that is not covered by our theoretical results. Still, we may try to recover it using the HiHTP algorithm.

We conduct simulations with $N_t = 1$ receive antenna and $N_r = 10$ total UEs. We set $N = 1024$ and $N = E = 128$. For each of the $N_r$ users, a $\sigma$-sparse channel $h_k \in \mathbb{R}^E$ is drawn with the locations of the non-zeros distributed uniformly and entries drawn from the standard normal distribution. The signals are computed as $x_k = Bc_k$, where $B \in \mathbb{R}^{N \times E}$ is a Gaussian random matrix and $c_k \in \mathbb{R}^E$ is $s$-sparse with values in $\{-1, 1\}$ if the user is active and 0 if the user is not active. This results in the data $y_1 \in \mathbb{R}^N$ as defined in (1.8).

We vary the number of active users $\mu$, as well as the sparsities $s$ and $\sigma$. Figures 1.6 and 1.7 show the rate of successful recovery for varying number of active users, averaged over 20 runs per setup. The x- and y-axis show the channel sparsity $\mu$ and the signal sparsity $s$, respectively. As can be seen, the HiHTP algorithm is indeed capable of recovering the ground truth, as long as the sparsity levels are low enough.

An interesting feature of the model is that it can be used to generate a secure communication scheme. To this end, we make use of the *reciprocity* of the channel: the channel $h_{p,q}^{\uparrow}$ for transmission from UE $q$ to base station antenna $p$ is equal to the channel $h_{p,q}^{\downarrow}$ for transmission in the other direction. This reciprocity condition is fulfilled for modern off-the-shelf WiFi devices [48]. Due to the reciprocity,

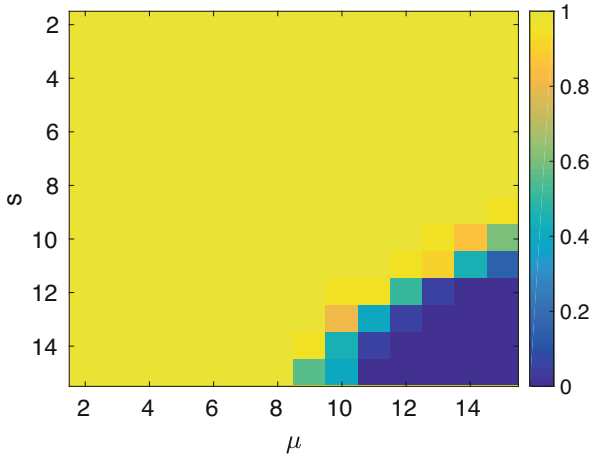



**Fig. 1.6** Recovery rate for 2 of 10 active users. ©2018 IEEE. Reprinted, with permission, from Ref. [52]

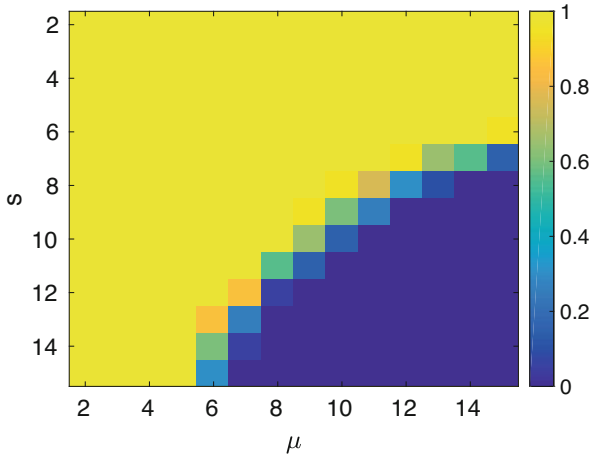



**Fig. 1.7** Recovery rate for 5 of 10 active users. ©2018 IEEE. Reprinted, with permission, from Ref. [52]

the channel itself can serve as a source of shared randomness for the secret key generation. The communication protocol consists of two phases:

**Phase 1**

1. The base station sends a predefined pilot signal to all UEs.
2. Each UE $q$ measures the complex-valued channel gains $h_q^{\downarrow} = (h_{p,q}^{\downarrow})_{p\in[N_r]}$.
3. Each UE encrypts his/her message $m$ to a sequence $c_p = f(m, h_q^{\downarrow})$, using some encryption scheme $f$ and $h_q^{\downarrow}$ as a random encryption key.

**Phase 2**

1. All the UEs $q$ send their encrypted *sequences* $c_q$ to the base station using the scheme discussed above. The encoding operators $B_p$ are left public.
2. The base station receives the superposition of all the convolutions of the cipher text with the respective channels. With a hierarchical thresholding algorithm, the station inverts (1.8) and, thus, gains knowledge of the cipher texts $c_p$ and channels $h^{\uparrow}_{p,q}$.
3. Due to reciprocity $h^{\uparrow}_{p,q} = h^{\downarrow}_{p,q}$, the base station thereby obtains the encryption keys $h^{\downarrow}_{q}$ and decrypts the cipher texts.

The security of the scheme relies on the assumption that the channels of different users are independent of each other and cannot be inferred from another position. Unless a man in the middle has access to the antenna of a UE, the eavesdropper cannot use his/her channel coefficients to recover the message of another user.

We note that small variations between both channels, i.e. small violations of reciprocity, can be tolerated by adjusting the key generation process. One can for example quantize the channel gain sufficiently coarse to equalize the keys. Here, the hierarchical framework is applied to solve a blind deconvolution and de-mixing problem. Refs. [16, 51] present further examples of the hierarchical measurement framework applied to massive random access without a built-in security scheme.

### *1.6.3 Blind Quantum State Tomography*

Quantum communication allows for the transmission of data under unprecedented levels of security [21]. Here, the security proofs are neither based on assumptions on the computational hardness of certain mathematical problems nor on the feasibility of practically reverting or predicting the randomness of physical processes: instead, there are proofs of security available based on the fundamental laws of nature themselves. Under mild assumptions, quantum key distribution can be proven secure under the most general attacks allowed by physics, within a paradigm of closed laboratories. Simultaneously, the advent of novel quantum computing devices promises solving certain tasks with a significantly improved computational complexity compared to classical computing devices. These tasks include NP problems at the heart of established and universally employed cryptographic schemes. It is beyond the scope of the present article to introduce the various applications of the quantum technologies [2]. Instead, we here focus on a particular context in which hierarchical compressed sensing naturally comes into play: this is the task of semi-device dependently identifying the state of a quantum device. Methods for such characterization and certification tasks are important diagnostic tools in the development of quantum technologies. We refer to Refs. [11, 27] for details.

The problem at hand here is the identification of quantum states prepared in some physical prescription. The recovery of unknown quantum states is called *quantum*

*state tomography*. A general quantum state is described by a trace-normalized, positive-definite complex matrix. Of particular interest are unit rank, so-called *pure* quantum states or more generally low-rank quantum states. Ideally, devices in quantum technologies operate or are envisioned to operate in pure quantum states of large dimensions. Quantum states of higher rank encode 'classical' statistical mixtures of pure states typically produced by noisy operations. We denote the set of rank-$r$ quantum states by $\mathcal{D}_r^n \subset \mathbb{C}^{n\times n}$.

An important diagnostic task for quantum devices is, thus, to learn the low-rank quantum state of the device from linear measurements. Exploiting the rank constraint on the quantum states in the recovery task is crucial to devise quantum tomography protocols working in state spaces of sizeable dimension. This renders compressed sensing method of crucial importance for quantum tomography [15, 22, 23, 26, 33, 37, 39, 44].

That said, the apparatus with which one performs the measurements can especially for near-term devices not be reasonably assumed to be fully characterized: commonly, there are calibrating parameters that are not fully known. An important practical problem is, thus, the recovery of a low-rank quantum state $\rho$ by means of measurement devices that are simultaneously themselves characterized by a handful of parameters, giving rise to sparse vectors $\xi$.

In a linear approximation of the measurement device calibration, this leads to the problem of *blind (self-calibrating) quantum state tomography*: let $\mathcal{A} : \mathbb{C}^{nd\times d} \to \mathbb{R}^m$ be a linear map describing the measurement and calibration model. Given data $y = \mathcal{A}(X) \in \mathbb{R}^m$ and the linear map $\mathcal{A}$, recover $X$ under the assumption that

$$X \in \{\xi \otimes \rho \mid \xi \in \mathbb{K}^N \ s\text{-sparse},\ \rho \in \mathcal{D}_r^n\} \subset \mathbb{C}^{Nn\times n}\,. \tag{1.9}$$

The blind quantum state tomography problem can be regarded as a non-commutative analogon of bisparse recovery problems where the data is bilinear in two sparse vectors both to be recovered. Similarly to the vector case, already the projection onto the set of structured signal is an NP-hard problem. In fact, one can encode the *sparse PCA problem* [31] and thereby CLIQUE into the task of finding the closest element of the form $\xi\otimes\rho$ with $\xi \in \mathbb{K}^N$, $\rho \in \mathcal{D}_r^n$ to a given $X \in \mathbb{K}^{Nn\times n}$ in Frobenius norm, Ref. [38, Theorem 3]. For this reason, it is not possible to directly derive an efficient algorithm based on a hard-thresholding operation for the blind quantum tomography problem.

However, the problem of de-mixing a sparse sum of low-rank matrices introduced in Sect. 1.5 can be seen as a relaxation to the closest hierarchically structured signal class that still allows for an efficient projection. The analogy to the relation of bisparsity and hierarchical sparsity is imminent.

Consequently, the SDT algorithm is a natural candidate to efficiently tackle the blind tomography problem. Figure 1.8 shows numerical simulations of the performance of the SDT algorithms in the blind quantum tomography task for a random calibration model motivated by quantum technologies in comparison to a standard low-rank tomography algorithm. The relaxation to the hierarchical structured problem, however, comes at the cost of a sub-optimal scaling in complexity theory. While

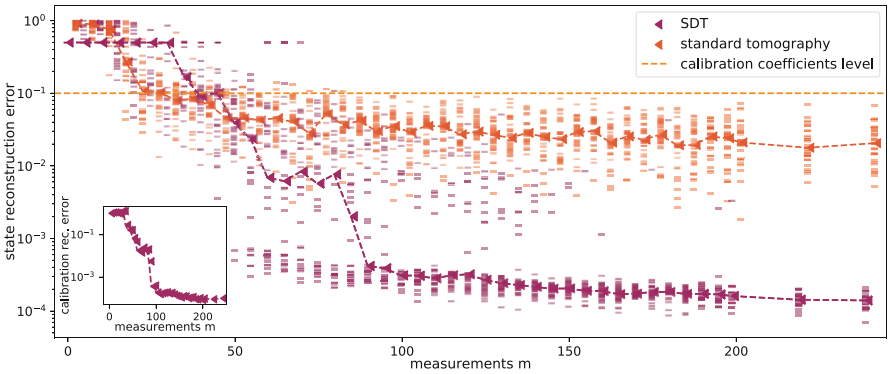



**Fig. 1.8** The figure (taken from Ref. [38]) displays the trace-norm reconstruction error for the SDT algorithm compared to a standard low-rank tomography algorithm for a different number of measurements $m$ of sub-sampled random Pauli measurements. Each point depicts 30 random measurement and signal instances with $r = 1, d = 8, n = 10$, and $s = 3$. The dotted lines indicate the median. The inline figure shows the mean $\ell_2$-norm reconstruction error of the calibration coefficients for the SDT algorithm

a parameter counting of the original blind tomography problem hints at an optimal scaling of $O(\max\{s \log N, nr\})$, the sparse de-mixing problem introduces already in parameter count an additional factor of $s$ to the second term $O(\max\{s \log N, snr\})$. Due to the sparsity assumption on the calibration parameters, the total number of calibration parameter $N$ still only enters logarithmically. For this reason, the scheme remains highly scalable in practically relevant parameter regimes despite the relaxation. At the same time, using the framework of hierarchical compressed sensing outlined above provides a rich toolkit to equip the SDT with flexible guarantees for many ensembles of measurement and calibration models. Another algorithmic approach to bilinear structured problems such as the blind tomography problem is constraint alternating minimization. We refer to Ref. [38] for further details.

## 1.7 Conclusion and Outlook

In this chapter, we have introduced a framework for hierarchically compressed sensing with a focus mostly on the reconstruction of hierarchically sparse signals. In its core, standard approaches of compressed sensing naturally generalize to hierarchically structured signals, giving rise to recovery algorithms equipped with theoretical guarantees. Thereby, the successful recovery of hierarchically sparse signals via hard-thresholding algorithms can be established under a custom-tailored restricted isometry assumption. There are, however, a number of specific features that separate the hierarchical framework from its more generic counterparts.

At the heart of the approach is the fact that the projection operator onto the set of hierarchically structured signals is efficiently calculable via hierarchical hard thresholding. Unlike for, e.g., the bisparse structure, it can be computed in linear time and is highly amenable to parallelization. This in turn renders the simple recovery algorithm interesting in realistic parameter regimes and under practical demands.

Furthermore, within the hierarchical framework, there is a large family of operators that obey the hierarchical, but not the standard restricted isometry property. This makes the framework potentially applicable in settings where standard compressed sensing is infeasible.

On a more theoretical level, the hierarchically sparse structure can be used as a relaxation of the complicated bisparse structure. In particular, we have presented numerical evidence that instances of the sparse blind deconvolution problem can be solved using HiHTP. And we have invoked the same strategy for the quantum tomography problem and other related questions. While in this context theoretical guarantees are expected to be sub-optimal, the simplicity and flexibility of the hierarchical framework might still be of merit in order to analyze complicated measurement settings. We leave further exploring these matters to future research. A particularly interesting question is to analyze the HiRIP properties of the blind deconvolution operator.

Indeed, we have at the end of this chapter seen several exemplary applications where the hierarchical approach facilitates recovery. This brings us to the arguably most important feature of the framework: hierarchically structured signals naturally emerge in many applications. From our own background and past research, we can conclude this with some confidence. But of course, we very much suspect that there are many applications we are unaware of where the hierarchical framework is readily applicable. For the sake of clarity, we have mainly focused our exposition on the set of two-level hierarchically sparse vectors and merely hinted at the generalizations towards multiple levels potentially mixing low-rankness and sparsity and potentially even further structures that for themselves come with an efficient projection. We hope that we have conveyed that the approach, and even most of the results we presented, can be rather straightforwardly generalized to this rich family of hierarchical signal structures, leaving the playing field wide open.

**Acknowledgments** This work is a report of some of the findings of the DFG-funded project EI 519/9-1 within the priority programme ‘Compressed Sensing in Information Processing’ (CoSIP), jointly held by J. Eisert and G. Wunder. We specifically thank our coauthors, in particular, M. Barzegar, G. Caire, R. Fritscheck, S. Haghighatshoar, D. Hangleiter, M. Kliesch, S. Stefanatos, R. Kueng, and J. Wilkens, with which we have explored this research theme over the years.

## References

1. Absil, P.A., Mahony, R., Sepulchre, R.: Optimization Algorithms on Matrix Manifolds. Princeton University Press, Princeton (2009)

2. Acin, A., Bloch, I., Buhrman, H., Calarco, T., Eichler, C., Eisert, J., Esteve, D., Gisin, N., Glaser, S.J., Jelezko, F., Kuhr, S., Lewenstein, M., Riedel, M.F., Schmidt, P.O., Thew, R., Wallraff, A., Walmsley, I., Wilhelm, F.K.: The European quantum technologies roadmap. New J. Phys. **20**, 080201 (2018). https://doi.org/10.1088/1367-2630/aad1ea
3. Adcock, B., Hansen, A.C., Poon, C., Roman, B.: Breaking the coherence barrier: a new theory for compressed sensing. Forum Math. Sigma **5** (2017). https://doi.org/10.1017/fms.2016.32
4. Baraniuk, R.G., Cevher, V., Duarte, M.F., Hegde, C.: Model-based compressive sensing. IEEE Trans. Inf. Theory **56**, 1982–2001 (2010). https://doi.org/10.1109/TIT.2010.2040894
5. Blumensath, T., Davies, M.E.: Iterative thresholding for sparse approximations. J. Four. An. App. **14**, 629–654 (2008). https://doi.org/10.1007/s00041-008-9035-z
6. Blumensath, T., Davies, M.E.: Normalized iterative hard thresholding: guaranteed stability and performance. IEEE J. Sel. Top. Sig. Proc. **4**, 298–309 (2010). https://doi.org/10.1109/JSTSP.2010.2042411
7. Bouchot, J.L., Foucart, S., Hitczenko, P.: Hard thresholding pursuit algorithms: number of iterations. App. Comp. Harm. An. **41**, 412–435 (2016). https://doi.org/10.1016/j.acha.2016.03.002
8. Candes, E.J., Tao, T.: Decoding by linear programming. IEEE Trans. Inf. Theory **51**, 4203–4215 (2005). https://doi.org/10.1109/TIT.2005.858979
9. Chen, Z., Yang, C.: Pilot decontamination in wideband massive MIMO systems by exploiting channel sparsity. IEEE Trans. Wirel. Commun. **15**, 5087–5100 (2016). https://doi.org/10.1109/TWC.2016.2553021
10. Dai, W., Milenkovic, O.: Subspace pursuit for compressive sensing signal reconstruction. IEEE Trans. Inf. Theory **55**, 2230–2249 (2009). https://doi.org/10.1109/TIT.2009.2016006
11. Eisert, J., Hangleiter, D., Walk, N., Roth, I., Markham, D., Parekh, R., Chabaud, U., Kashefi, E.: Quantum certification and benchmarking. Nat. Rev. Phys. **2**, 382–390 (2020). https://doi.org/10.1038/s42254-020-0186-4
12. Eldar, Y.C., Kutyniok, G.: Compressed Sensing: Theory and Applications. Cambridge University Press, Cambridge (2012)
13. Eldar, Y.C., Mishali, M.: Block sparsity and sampling over a union of subspaces. In: Digital Signal Processing, 2009 16th International Conference on, pp. 1–8 (2009). https://doi.org/10.1109/ICDSP.2009.5201211
14. Eldar, Y.C., Mishali, M.: Robust recovery of signals from a structured union of subspaces. IEEE Trans. Inf. Theory **55**, 5302–5316 (2009). https://doi.org/10.1109/TIT.2009.2030471
15. Flammia, S.T., Gross, D., Liu, Y.K., Eisert, J.: Quantum tomography via compressed sensing: error bounds, sample complexity and efficient estimators. New J. Phys. **14**, 095022 (2012). https://doi.org/10.1088/1367-2630/14/9/095022
16. Flinth, A., Groß, B., Roth, I., Eisert, J., Wunder, G.: Hierarchical isometry properties of hierarchical measurements (2021)
17. Foucart, S.: Hard thresholding pursuit: an algorithm for compressive sensing. SIAM J. Num. An. **49**, 2543–2563 (2011). https://doi.org/10.1137/100806278
18. Foucart, S., Gribonval, R., Jacques, L., Rauhut, H.: Jointly low-rank and bisparse recovery: questions and partial answers. Preprint (2019). ArXiv:1902.04731
19. Foucart, S., Rauhut, H.: A Mathematical Introduction to Compressive Sensing. Springer (2013)
20. Friedman, J., Hastie, T., Tibshirani, R.: A note on the group lasso and a sparse group lasso. Preprint (2010). ArXiv: 1001.0736
21. Gisin, N., Ribordy, G., Tittel, W., Zbinden, H.: Quantum cryptography. Rev. Mod. Phys. **74**, 145–195 (2002). https://doi.org/10.1103/RevModPhys.74.145
22. Gluza, M., Schweigler, T., Rauer, B., Krumnow, C., Schmiedmayer, J., Eisert, J.: Quantum read-out for cold atomic quantum simulators. Phys. Commun. **20**, 12 (2020). https://doi.org/10.1038/s42005-019-0273-y
23. Gross, D., Liu, Y.K., Flammia, S.T., Becker, S., Eisert, J.: Quantum state tomography via compressed sensing. Phys. Rev. Lett. **105**, 150401 (2010). https://doi.org/10.1103/PhysRevLett.105.150401

24. Hoare, C.A.R.: Algorithm 65: find. Commun. ACM **4**, 321–322 (1961). https://doi.org/10.1145/366622.366647
25. Jokar, S., Mehrmann, V.: Sparse solutions to underdetermined Kronecker product systems. Linear Algebra Appl. **431**, 2437–2447 (2009). https://doi.org/10.1016/J.LAA.2009.08.005
26. Kalev, A., Kosut, R.L., Deutsch, I.H.: Quantum tomography protocols with positivity are compressed sensing protocols. NJP Quant. Inf. **1**, 15018 (2015). https://doi.org/10.1038/npjqi.2015.18
27. Kliesch, M., Roth, I.: Theory of quantum system certification. PRX Quantum **2**, 010201 (2021). https://doi.org/10.1103/PRXQuantum.2.010201
28. Li, C., Adcock, B.: Compressed sensing with local structure: uniform recovery guarantees for the sparsity in levels class. Appl. Comput. Harm. Anal. **46**, 453–477 (2019). https://doi.org/10.1016/j.acha.2017.05.006
29. Ling, S., Strohmer, T.: Blind deconvolution meets blind demixing: algorithms and performance bounds. IEEE Trans. Inf. Theory **63**, 4497–4520 (2017)
30. Liu, H., Sun, F.: Hierarchical orthogonal matching pursuit for face recognition. In: The First Asian Conference on Pattern Recognition, pp. 278–282 (2011). https://doi.org/10.1109/ACPR.2011.6166530
31. Magdon-Ismail, M.: NP-hardness and inapproximability of sparse PCA. Inf. Proc. Lett. **126**, 35–38 (2017). https://doi.org/10.1016/j.ipl.2017.05.008
32. Needell, D., Tropp, J.A.: CoSaMP: Iterative signal recovery from incomplete and inaccurate samples. Appl. Comp. Harm. Anal. (2008). https://doi.org/10.1016/j.acha.2008.07.002
33. Riofrio, C.A., Gross, D., Flammia, S.T., Monz, T., Nigg, D., Blatt, R., Eisert, J.: Experimental quantum compressed sensing for a seven-qubit system. Nat. Commun. **8**, 15305 (2017). https://doi.org/10.1038/ncomms15305
34. Roth, I., Flinth, A., Kueng, R., Eisert, J., Wunder, G.: Hierarchical restricted isometry property for Kronecker product measurements. In: 2018 56th Annual Allerton Conference on Communication, Control, and Computing (Allerton), pp. 632–638. IEEE (2018). https://doi.org/10.1109/ALLERTON.2018.8635829
35. Roth, I., Kliesch, M., Flinth, A., Wunder, G., Eisert, J.: Reliable recovery of hierarchically sparse signals for Gaussian and Kronecker product measurements. IEEE Trans. Signal Process. **68**, 4002–4016 (2020). https://doi.org/10.1109/tsp.2020.3003453
36. Roth, I., Kliesch, M., Wunder, G., Eisert, J.: Reliable recovery of hierarchically sparse signals. In: Proceedings of the third "International Traveling Workshop on Interactions between Sparse models and Technology" (iTWIST'16), pp. 36–38 (2016)
37. Roth, I., Kueng, R., Kimmel, S., Liu, Y.K., Gross, D., Eisert, J., Kliesch, M.: Recovering quantum gates from few average gate fidelities. Phys. Rev. Lett. **121** (2018). https://doi.org/10.1103/physrevlett.121.170502
38. Roth, I., Wilkens, J., Hangleiter, D., Eisert, J.: Semi-device-dependent blind quantum tomography. Preprint (2020). ArXiv:2006.03069
39. Shabani, A., Kosut, R.L., Mohseni, M., Rabitz, H., Broome, M.A., Almeida, M.P., Fedrizzi, A., White, A.G.: Efficient measurement of quantum dynamics via compressive sensing. Phys. Rev. Lett. **106**, 100401 (2011). https://doi.org/10.1103/PhysRevLett.106.100401
40. Shabara, Y., Koksal, C.E., Ekici, E.: How long to estimate sparse MIMO channels. Preprint. arXiv:2101.07287 (2021)
41. Simon, N., Friedman, J., Hastie, T., Tibshirani, R.: A sparse-group Lasso. J. Comp. Graph. Stat. **22**, 231–245 (2013). https://doi.org/10.1080/10618600.2012.681250
42. Sprechmann, P., Ramirez, I., Sapiro, G., Eldar, Y.: Collaborative hierarchical sparse modeling. In: 2010 44th Annual Conference on Information Sciences and Systems (CISS), pp. 1–6 (2010). https://doi.org/10.1109/CISS.2010.5464845
43. Sprechmann, P., Ramirez, I., Sapiro, G., Eldar, Y.C.: C-HiLasso: a collaborative hierarchical sparse modeling framework. IEEE Trans. Sig. Proc. **59**, 4183–4198 (2011). https://doi.org/10.1109/TSP.2011.2157912

44. Steffens, A., Riofrío, C.A., McCutcheon, W., Roth, I., Bell, B.A., McMillan, A., Tame, M.S., Rarity, J.G., Eisert, J.: Experimentally exploring compressed sensing quantum tomography. Quantum Sci. and Technol. **2**, 025005 (2017). https://doi.org/10.1088/2058-9565/aa6ae2
45. Strohmer, T., Wei, K.: Painless breakups-efficient demixing of low rank matrices. J. Four. Ana. App. **25**, 1–31 (2019). https://doi.org/10.1007/s00041-017-9564-4
46. Tropp, J.A.: Greed is good: algorithmic results for sparse approximation. IEEE Trans. Inf. Theory **50**, 2231–2242 (2004). https://doi.org/10.1109/TIT.2004.834793
47. Vandereycken, B.: Low-rank matrix completion by Riemannian optimization. SIAM J. Opt. **23**, 1214–1236 (2013). https://doi.org/10.1137/110845768
48. Vasisht D. Kumar, S., Katabi, D.: Decimeter-level localization with a single WiFi access point. In: NSDI, pp. 165–178 (2016)
49. Wei, K., Cai, J.F., Chan, T.F., Leung, S.: Guarantees of Riemannian optimization for low rank matrix recovery. SIAM J. Math. Appl. **37**, 1198–1222 (2016). https://doi.org/10.1137/15M1050525
50. Wilkens, J., Hangleiter, D., Roth, I.: (2020). Open source Gitlab repository at https://gitlab.com/wilkensJ/blind-quantum-tomography
51. Wunder, G., Flinth, A., Groß, B.: Measure concentration on the OFDM-based massive random access channel. In: 2021 IEEE Statistical Signal Processing Workshop (SSP), pp. 526–530 (2021)
52. Wunder, G., Roth, I., Fritschek, R., Groß, B., Eisert, J.: Secure massive IoT using hierarchical fast blind deconvolution. In: 2018 IEEE Wireless Communications and Networking Conference Workshops (WCNCW), pp. 119–124. IEEE (2018)
53. Wunder, G., Stefanatos, S., Flinth, A., Roth, I., Caire, G.: Low-overhead hierarchically-sparse channel estimation for multiuser wideband massive MIMO. IEEE Trans. Wirel. Commun. **18**, 2186–2199 (2019)

# Chapter 2
# Proof Methods for Robust Low-Rank Matrix Recovery


**Tim Fuchs, David Gross, Peter Jung, Felix Krahmer, Richard Kueng, and Dominik Stöger**

## 2.1 Introduction

Computationally tractable data acquisition in high dimensions is a fundamental problem in various real-world applications in signal processing, data science, and physics. Nyquist sampling or scanning the data in full is often unfeasible. This motivates the use of compressive observation schemes, which employ regularization methods to recover as much of the signal as possible from seemingly incomplete observations. Thus, quantifying the trade-off between sample complexity and recon-

T. Fuchs (✉)
Department of Mathematics, Technical University of Munich, Garching, Germany
e-mail: tim.fuchs@tum.de

D. Gross
Institute for Theoretical Physics, University of Cologne, Cologne, Germany
e-mail: david.gross@thp.uni-koeln.de

P. Jung
Communications and Information Theory Group, Technische Universität Berlin, Berlin, Germany

Data Science in Earth Observation, Technical University of Munich, Munich, Germany
e-mail: peter.jung@tu-berlin.de

F. Krahmer
Department of Mathematics, Technical University of Munich, Garching, Germany

Munich Data Science Institute, Technical University of Munich, Garching, Germany
e-mail: felix.krahmer@tum.de

R. Kueng
Institute for Integrated Circuits, Johannes Kepler University Linz, Linz, Austria
e-mail: richard.kueng@jku.at

D. Stöger
Department of Mathematics, KU Eichstätt-Ingolstadt, Eichstätt, Germany
e-mail: dominik.stoeger@ku.de



G. Kutyniok et al. (eds.), *Compressed Sensing in Information Processing*, Applied and Numerical Harmonic Analysis, https://doi.org/10.1007/978-3-031-09745-4_2

struction accuracy has become a key task for identification of feasible regimes and the design of efficient approaches for sensing and reconstruction. These questions have been central in the area of inverse problems for many years. However, starting in the early 2000s, a highly successful novel viewpoint has emerged. Namely initiated by the influential works on compressed sensing [15, 17], various authors have studied the problem of what can be gained when the measurements can be optimized over all vectors or within a structural measurement framework [31]. Commonly, the term compressed sensing is used nowadays also for more general sensing scenarios beyond the initial setup that follow this paradigm.

In this generality, compressed sensing is therefore concerned with the recovery of structured data, i.e., data that lives on a low-dimensional subset embedded in a high-dimensional space, from a number of observations that scale with the intrinsic dimension, rather than the ambient dimension. As it turns out, for a large class of different measurement models combined with various structural constraints, choosing the free parameters of the measurement scheme at random leads to near-optimal performance.

Initially, the model of a nontrivial but relevant low-dimensional set was given by *sparse vectors*. For matrices, a natural basis independent notion of sparsity is "sparsity in the eigenbasis," i.e., *low rank*, and we are thus led to studying the *low-rank matrix recovery problem*: estimate an unknown $n_1 \times n_2$-matrix $X_0$ from $m$ observations of the form

$$y = \mathcal{A}(X_0) + e \in \mathbb{C}^m, \tag{2.1}$$

where $\mathcal{A}$ is a known linear measurement operator and $e$ is additive noise. Here and in the following, we will use the notation

$$\mathcal{A}(X_0)(i) := \langle A_i, X_0 \rangle, \quad A_i \in \mathbb{C}^{n_1 \times n_2}, \tag{2.2}$$

which expresses the $i$th component of the measurement as the Frobenius inner product with a matrix $A_i$. The problem is interesting in the regime

$$\mathrm{rk}(X_0) \max\{n_1, n_2\} \leq m \ll n_1 n_2,$$

where $\mathrm{rk}(X_0)$ denotes the matrix rank. Assume for concreteness that we have the bound $\|e\|_2 \leq \tau$ on the noise strength ($\|\cdot\|_2$ denotes the $\ell_2$-norm of a vector). As $X_0$ has low-rank, one could naively try to estimate $X_0$ by solving the following minimization problem:

$$\begin{aligned} &\underset{X \in \mathbb{C}^{n_1 \times n_2}}{\text{minimize}} && \mathrm{rk}(X) \\ &\text{subject to} && \|\mathcal{A}(X) - y\|_2 \leq \tau. \end{aligned}$$

Unfortunately, problems of this type are NP-hard in general, as minimizing the support size of a vector (i.e., finding the sparsest solution) can be considered as a special case [74]. Therefore, in [26], it was proposed to use the nuclear norm $\|\cdot\|_*$

(the sum of singular values) as a proxy for the rank. For this reason, the following approach has been suggested [14, 18, 20, 36, 76] for matrix completion:

$$\begin{aligned} &\underset{X\in\mathbb{C}^{n_1\times n_2}}{\text{minimize}} && \|X\|_* \\ &\text{subject to} && \|\mathcal{A}(X)-y\|_2 \le \tau. \end{aligned} \tag{2.3}$$

It is the analysis of this semi-definite program (SDP) we are concerned with in the present chapter. Before tackling the technical details, we briefly list three important applications of the framework of low-rank matrix recovery.

## 2.1.1 Sample Applications

In this section, we highlight three famous applications of low-rank matrix recovery which have been investigated intensively in the last years.

### 2.1.1.1 Matrix Completion

Maybe the most natural instantiation of the general model (2.2) is the case where the measurements reveal individual matrix elements

$$\mathcal{A}(X)(i) := \sqrt{\tfrac{n_1 n_2}{m}}\langle X, e_{a_i} e_{b_i}^*\rangle_F = \sqrt{\tfrac{n_1 n_2}{m}} X_{a_i,b_i}, \tag{2.4}$$

where $\{e_{a_i}\}$ and $\{e_{b_i}\}$ denote the standard basis of $\mathbb{C}^{n_1}$ and $\mathbb{C}^{n_2}$, respectively. This is the *matrix completion problem*. Since it arises in many different applications such as multiclass learning [3], collaborative filtering [77], and distance matrix completion problem in sensor localization tasks [44], see here also Fig. 2.1, it has become very popular in the last decade and has been studied intensively in the statistics, machine learning, and signal processing literature.

Assume that the matrix elements $(a_i, b_i)$ for $i \in [m] := \{1 \dots m\}$ to be revealed are chosen independently and uniformly among all $n_1 \times n_2$ possibilities. It is clear that not *all* low-rank matrices can be efficiently recovered from few such measurements. For example, if $X$ has a single non-zero entry, then unless $m = O(n_1 n_2)$, the probability that any non-zero information is obtained is small.

To identify a set of well-behaved instances, Ref. [14] introduced the following two *coherence parameters*:

$$\mu(U) := \sqrt{\tfrac{n_1}{r}} \max_{i\in[n_1]} \|U^* e_i\|_2$$

$$\mu(V) := \sqrt{\tfrac{n_2}{r}} \max_{i\in[n_2]} \|V^* e_i\|_2,$$

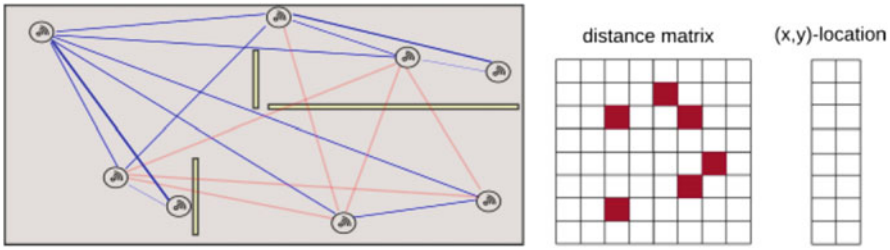



**Fig. 2.1** *Distance matrix completion:* one source of low-rank matrices is *Gram matrices* which encode the Euclidean geometries of point sets. The task of recovering Gram matrices is related to the one of finding *distance matrices* from few pairwise distances. This problem appears, for example, in sensor localization; see, e.g., [44]. Wireless sensors are distributed in an area (indoor room, industry hall, etc.) and measure individual signal strengths, but obstacles like walls block certain path directions (red). The goal is to complete the matrix of pairwise distances and compute the sensor locations. Recall that given $n$ points $\{x_i\}_{i=1}^n \subset \mathbb{R}^d$, the Gram matrix $G_{i,j} = \langle x_i, x_j \rangle$ has rank upper bounded by $d$, independent of $n$

where $X_0 = U \Sigma V^*$ with $U \in \mathbb{C}^{n_1 \times r}$ and $V \in \mathbb{C}^{n_2 \times r}$ denotes the singular value decomposition (SVD). Indeed, it was shown in [18] that

$$m \gtrsim n_1 r \log n_1 \max \left\{ \mu^2(U), \mu^2(V) \right\} \tag{2.5}$$

observations are necessary for a rank-$r$ matrix $X_0$ to be uniquely determined from the revealed entries.

Subsequently, a series of works [36, 76] established that this sampling rate is almost sufficient as well. Compared to Eq. (2.5), an additional $\log(n)$-factor and a third incoherence parameter suffice to ensure exact recovery via nuclear norm minimization (2.3). See Sect. 2.3 for a detailed statement and proof sketch.

#### 2.1.1.2 Blind Deconvolution

Blind deconvolution [41] refers to the problem of recovering a signal $x \in \mathbb{C}^L$ from the noisy convolution $w * x + e \in \mathbb{C}^L$, where $w \in \mathbb{C}^L$ is an unknown kernel and $e \in \mathbb{C}^L$ refers to additive noise. When using appropriate cyclic extensions or considering zero padding, the convolution can rewritten as a circular convolution

$$(w * x)(i) := \sum_{j=1}^{L} w_j x_{i-j} \quad \text{for } i \in [L]. \tag{2.6}$$

The difference $i - j$ is considered modulo $L$. As prototypical example of a bilinear inverse problem, blind deconvolution refers to recovering $(x, w)$ from a noisy version of $w * x$ and the precise role of $x$ and $w$ depends on the underlying

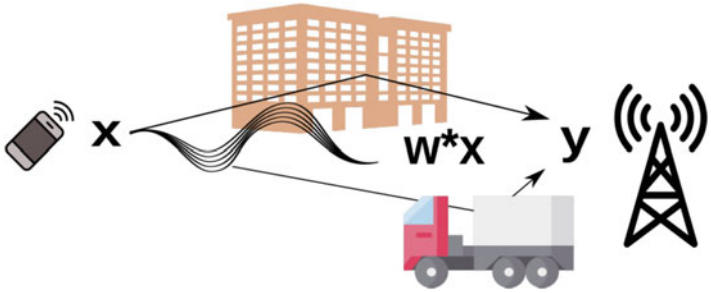



**Fig. 2.2** *Blind deconvolution*: a wireless device transmits a signal $x$ received by the base station. Due to reflections in the environment, the signal experiences an unknown channel distortion, represented by a convolution $w * x$ with the impulse response $w$

application. In imaging, for example, one considers such a problem for the two-dimensional convolution. The signal $x$ typically represents the image and $w$ is an unknown blurring kernel [79]. In communication engineering, the discrete model above describes the effective convolution in complex baseband. Hence, $w$ represents the sampled impulse response of the transmission channel and the task is to demodulate and decode information from the signal vector $x$, only having access to the noisy channel output $w * x + e$, see Fig. 2.2. The conventional coherent approach in this application is to send known pilot signals, to first estimate $w$ and then demodulate later information-bearing signals $x$. However, this approach is not feasible for short signals $x$ and for communication at low latency or high mobility. For communication engineers, the important question is then how much overhead is required for coping with the unknown impulse response $w$ of the communication channel [34] when using non-coherent strategies [87].

Of course, blind deconvolution is a highly underdetermined bilinear inverse problem. Without further assumptions, recovery is only possible up to inherent ambiguities [23, 88]. To avoid nontrivial ambiguities, one has to further constrain the vectors, for example, by assuming that $x$ and $w$ lie in $N$ and $K$-dimensional subspaces, respectively. As we will outline below, this yields to the problem of recovering $N \times K$ matrices from $L$ observations. To be more compliant with existing works in the literature, we will stick to this notation implying that $n_1 = N$, $n_2 = K$, and $m = L$. In formulas, we assume that $w = Bh_0$ and $x = C\overline{m}_0$ for given $B \in \mathbb{C}^{L\times K}$, $C \in \mathbb{C}^{L\times N}$, and unknown $h_0 \in \mathbb{C}^K$, $m_0 \in \mathbb{C}^N$. Then, the measurement operator acting on $h_0$ and $m_0$ is known to be generically injective up to the unavoidable scaling ambiguity if and only if $L \geq 2(N+K-4)$ [49, 68]. That is, one aims for sampling complexities that are near-linear in $N + K$.

Following [1], we consider the case that $B$ is a fixed matrix such that $B^*B = \mathrm{Id}$ and $C$ is a random matrix with i.i.d. complex normal entries $C_{ij} \overset{iid}{\sim} \mathcal{CN}(0, 1/\sqrt{L})$. The choice of the *random* matrix $C$ is motivated by the success of randomization in compressed sensing as well as by applications in wireless communications. Here $m_0$ contains a message to be transmitted and $C$ is a coding matrix. The signal $x = C\overline{m}_0$ gets transmitted through a time-invariant channel, which can be modeled as

a circular convolution with impulse response $w$ when using an appropriate cyclic prefix.

In many applications, it is reasonable to assume that only the first few entries of $w$ are non-zero as the path delays are often much shorter than the length of the signals $x$. In this case, $B$ would be the matrix which extends $h_0 \in \mathbb{C}^K$ by zeros. Hence, the receiver observes $w * x + e$, where $e$ represents additive noise, and the goal is to reconstruct the original message contained in the vector $m_0$.

Now let $F \in \mathbb{C}^{L\times L}$ be the unitary discrete Fourier transformation matrix. It is well known that $F$ diagonalizes the circular convolution, i.e.,

$$\widehat{w * x} := F(w * x) = \sqrt{L}\,\mathrm{diag}(FBh_0)\,FC\overline{m}_0.$$

Let $b_\ell$ denote the $\ell$th row of $\overline{FB}$, and let $c_\ell$ denote the $\ell$th row of $\sqrt{L}FC$. Note that this implies that all the entries of $\{c_\ell\}_{\ell=1}^L$ are jointly independent and have distribution $\mathcal{CN}(0, 1)$. Moreover, we obtain that

$$\left(\widehat{w * x}\right)_\ell = b_\ell^* h_0 m_0^* c_\ell = \langle b_\ell c_\ell^*, h_0 m_0^*\rangle.$$

We observe that $\widehat{w * x}$ is linear in the $K \times N$ matrix $h_0 m_0^*$. This motivates the definition of the linear operator $\mathcal{A} : \mathbb{C}^{K\times N} \to \mathbb{C}^L$ by

$$(\mathcal{A}(X))(\ell) := \langle b_\ell c_\ell^*, X\rangle \quad \text{where } \ell \in [L]. \tag{2.7}$$

Hence, we obtain the model

$$y := \widehat{w * x} + e = \mathcal{A}(X_0) + e,$$

where $X_0 = h_0 m_0^*$ and $e \in \mathbb{C}^L$ represents noise with $\|e\|_2 \le \tau$. Note that $X_0$ is a rank-one matrix. This reformulation effectively reduces blind deconvolution to a low-rank matrix recovery problem, where measurement matrices correspond to outer products $A_\ell = b_\ell c_\ell^*$.

If, in addition, a sparsity constraint is to be imposed, the problem becomes considerably more difficult. In particular, linear combinations of the convex regularizes no longer lead to sample-efficient recovery guarantees [75] even when using optimal tuning [52]. Only under additional structural assumptions, recovery guarantees are available using an alternating minimization approach [33, 67]. This, however, is beyond the scope of this chapter.

#### 2.1.1.3 Phase Retrieval

Another instance of a challenging inverse problem is *phase retrieval*—an important problem with a long history that dates back to the 60s [89]. It occurs naturally in X-ray crystallography [40, 73], astronomy [28], ptychography [42, 78], and quantum tomography [61, 64]. We refer to Fig. 2.3 for a visual illustration. Mathematically

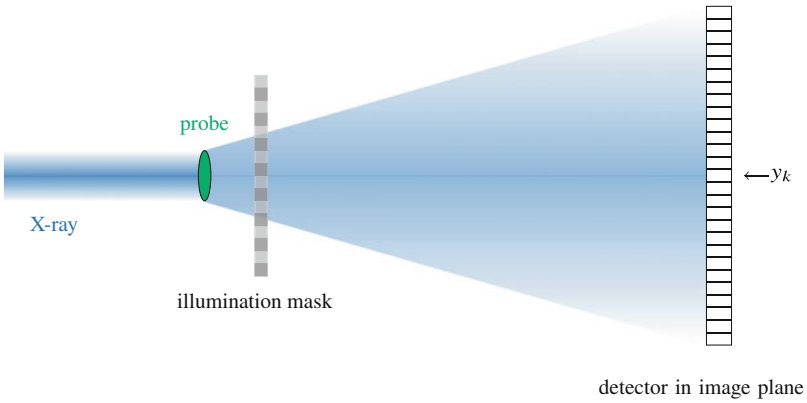



**Fig. 2.3** *Phase retrieval:* in diffraction imaging, a probe is illuminated by coherent X-ray light. The resulting diffraction pattern is first modulated by an illumination mask and recorded at detectors in the 2D image plane. Importantly, these detectors can only record intensities, not phases: $y_k = |\langle f_k, D^* x_0 \rangle|^2$, where $x_0 \in \mathbb{C}^n$ encodes the microscopic structure of the probe, $D = \operatorname{diag}(d_1, \ldots, d_n)$ describes the illumination mask, and $f_k \in \mathbb{C}^n$ is a Fourier vector (Fraunhofer approximation to the diffraction equation)

speaking, the discrete phase retrieval problem asks for inferring a complex signal vector $x_0 \in \mathbb{C}^n$ from $m$ measurements of the form (noiseless for simplicity)

$$\tilde{y}_i = |\langle a_i, x_0 \rangle| , \quad i \in [m]. \tag{2.8}$$

This problem cannot be solved unless the measurement system is overcomplete because all phase information is lost in the measurement process. More precisely, it has been shown that one needs $m \geq 4n - 4$ generic measurements to ensure that there is a unique solution [24].

If, in contrast, one instead had access to the complex phases $\phi_k$ of $\langle a_k, x_0 \rangle$, this problem would reduce to solving a linear system of equations:

$$\Phi \tilde{y} = A x_0, \tag{2.9}$$

where $\Phi = \sum_{k=1}^m \bar{\phi}_k e_k e_k^*$ and $A = \sum_{k=1}^m e_k a_k^*$ subsumes the measurement process. Crucially, for phase retrieval, we do not know $\Phi$ in (2.9). One intuitive approach to recovering $x_0$ is performing a least-squares minimization over both unknowns:

$$\underset{\Phi \in U(m), x \in \mathbb{C}^n}{\text{minimize}} \quad \|\Phi \tilde{y} - Ax\|_2 , \tag{2.10}$$

$\Phi \in U(m)$ is unitary and diagonal in the standard basis. Although NP-hard in general, heuristic approaches exist for solving non-convex problems of this form. One such heuristics is *alternating minimization*, see, e.g., [29, 72]. This is an iterative algorithm, where one alternates between keeping $x$ fixed and minimizing

$\Phi$ and vice versa: fixing $\Phi$ and optimizing over $x$. Very few theoretical guarantees regarding its performance are known.

Given the importance of the problem and the lack of mathematical understanding, obtaining theoretical guarantees for phase retrieval is highly desirable. In order to do so, we will follow a different direction pioneered by Balan, Bodmann, Casazza, and Eddidin [4]: lift the quadratic phase retrieval problem to a linear inverse problem on positive semidefinite $n \times n$ matrices:

$$y_i = |\langle a_i, x_0\rangle|^2 = \mathrm{tr}\langle a_i a_i^* \; X_0\rangle \quad \text{where} \quad X_0 = x_0 x_0^* \in \mathbb{C}^{n\times n} \tag{2.11}$$

is proportional to the orthoprojector onto $\mathrm{span}(x_0) \subset \mathbb{C}^n$. By construction, the desired solution is a Hermitian $n \times n$ matrix with minimal rank ($\mathrm{rk}(X_0) = 1$). Following Refs. [8, 16], we can exploit this intrinsic rank constraint via constrained nuclear norm minimization (2.3). This approach effectively reduces the phase retrieval problem to a Hermitian low-rank matrix recovery problem, where each linear measurement (2.2) must only involve (Hermitian) outer products:

$$y_i = \mathcal{A}\left(x_0 x_0^*\right)(i) = \langle A_i, x_0 x_0^*\rangle, \quad \text{where } A_i = a_i a_i^* \in \mathbb{H}_n \text{ and } i \in [m].$$

The reformulation of phase retrieval as a low-rank matrix recovery problem has led to the establishment of rigorous recovery guarantees. By and large, these apply to randomly selected measurement vectors that are sufficiently "generic." Exemplary is the main result from Ref. [9]: already $m \gtrsim n$ standard complex Gaussian measurements $a_1, \ldots, a_m \overset{iid}{\sim} \mathcal{CN}(0, I)$ suffice to ensure correct recovery. Subsequent research has led to similar recovery guarantees for phaseless measurements that are less generic [10, 38, 48]. We will present two such arguments further below. In Sect. 2.2.3, we partially derandomize the recovery guarantee for Gaussian measurements by executing a descent cone analysis.

We conclude by emphasizing that the phase retrieval problem admits a clean reformulation in terms of low-rank matrix recovery. This is an ideal starting point for developing rigorous convergence guarantees but might come with a considerable algorithmic overhead. After all, we have replaced a non-convex problem over $n$-dimensional vectors by a convex problem over (Hermitian) $n \times n$ matrices (2.3). General purpose solvers, like CVX, quickly run into storage issues as the problem dimension $n$ increases. This motivated the development and rigorous analysis of non-convex phase retrieval algorithms. These include gradient descent-type algorithms on $\mathbb{C}^n$ [7, 11, 21], as well as non-convex approaches based on matrix factorization [6, 42]. In parallel, the development of matrix sketching algorithms led to substantial storage and runtime improvements for solving certain convex optimization problems [86, 91]. Importantly, these also apply to lifted phase retrieval and ensure algorithmic tractability even for moderate to large problem sizes [91]. So, recovery guarantees for lifted phase retrieval—like the ones presented in this book chapter—are also of algorithmic relevance.

### 2.1.2 This Work

In this book chapter, we take a look back at more than a decade of rapid progress concerning randomized inverse problems for matrix recovery. A complete treatment of all interesting developments would go way beyond the scope of a single chapter and we choose to focus on one aspect: mathematically rigorous recovery guarantees for the reconstruction of low-rank matrices from generic as well as structured measurements.

With the benefit of hindsight, we review two versatile proof techniques and put them into context, namely the descent cone analysis, as well as the construction of approximate dual certificates.

Section 2.2 deals with the descent cone analysis. That is, low-rank matrix recovery guarantees are obtained by analyzing the relative geometric orientation of the optimization problem's feasible space with respect to the objective function's descent cone anchored at the signal $X_0$ of interest. Exact and unique recovery happens if and only if the intersection of these two convex objects only contains a single point. Deep results from high-dimensional probability theory show that this desirable event happens with overwhelming probability, provided that the measurements are sampled independently from sufficiently generic ensembles. Prominent example applications include optimal generic low-rank matrix recovery (Sect. 2.2.2), as well as phase retrieval from generic measurement vectors (Sect. 2.2.3). Although geometrically appealing, this proof technique is not without limitations. It struggles to handle less generic problems, where additional structure—like incoherence of the unknown signals—is essential to rule out exceptional problem instances where the reconstruction must necessarily fail. Moreover, this technique does not always give precise insights into the noise robustness of the reconstruction schemes (Sect. 2.2.4).

Section 2.3 introduces an alternative proof technique based on duality of convex optimization. Convex optimization problems—like nuclear norm minimization (2.3)—come in pairs and the two problems have a duality gap: objective function values of the primal problem are always smaller than or equal to objective function values of the dual problem. Equality occurs if and only if both primal and dual solutions are optimal. This, in turn, implies that optimality of a certain feasible point, say $X_0$, can be certified by constructing a dual feasible point that achieves the same objective function value. What is more, exact feasibility is not required to certify optimality of $X_0$ for constrained nuclear norm minimization. An *approximate dual certificate* suffices, provided that the measurement operator fulfills certain additional properties (Sect. 2.3.1).

We will then describe how to construct approximate dual certificates via a probabilistic method—the so-called golfing scheme (Sect. 2.3.2). A key advantage of the golfing scheme is that it can be applied to problems with incoherence constraints, where it is not immediately clear how to apply the methods described in Sect. 2.2. Concrete example applications are matrix completion (Sect. 2.3.3), blind deconvolution and demixing (Sect. 2.3.4), and phase retrieval with incoherence (Sect. 2.3.5).

Approximate dual certificates do also have their downsides, however. Chief among them is noise robustness. In Sect. 2.4, we refine the descent cone arguments introduced in Sect. 2.2. This leads to near-optimal blind deconvolution guarantees in the high-noise regime (Sect. 2.4.1), as well as novel insights into the phase retrieval problem (Sect. 2.4.2).

## 2.2 Recovery Guarantees via a Descent Cone Analysis

### *2.2.1 Descent Cone Analysis*

Recalling the linear inverse problem (2.1) $y = \mathcal{A}(X_0) + e \in \mathbb{C}^m$, there is usually a large set of possible solutions for which $\mathcal{A}$ does not deviate too much from $y$. Further properties, such as low rank, can be obtained by minimizing an appropriate function $f : \mathbb{C}^{n_1 \times n_2} \to \mathbb{R}$ over this set. If $f$ yields low values only for a small subset of $\{X \in \mathbb{C}^{n_1 \times n_2} : \|\mathcal{A}(X) - y\|_2 \leq \tau\}$ (or $\{X \in \mathbb{C}^{n_1 \times n_2} : \mathcal{A}(X) = y\}$ in the noiseless case), recovery guarantees can be obtained. This motivates descent cone analysis.

The descent cone $\mathcal{D}(f, X_0)$ of a proper convex function $f : \mathbb{C}^{n_1 \times n_2} \to \mathbb{R}$ at a point $X_0 \in \mathbb{C}^{n_1 \times n_2}$ is the conic hull of directions in which $f$ decreases near $X_0$:

$$\mathcal{D}(f, X_0) := \{Z \in \mathbb{C}^{n_1 \times n_2} : f(X_0 + \epsilon Z) \leq f(X_0) \text{ for some } \epsilon > 0\}.$$

Descent cone analysis can facilitate the estimation of probability of success for solving linear inverse problems with optimization. Consider the following two convex optimization problems (left: noiseless and right: noisy measurements)

$$\begin{aligned} &\text{minimize} && f(X) \\ &\text{subject to} && \mathcal{A}(X) = y. \end{aligned} \tag{2.12}$$

$$\begin{aligned} &\text{minimize} && f(X) \\ &\text{subject to} && \|\mathcal{A}(X) - y\|_2 \leq \tau. \end{aligned} \tag{2.13}$$

Let us first discuss the noiseless case. If $X_0$ is the ground truth of the measurements $\mathcal{A}(X_0) = y$, any minimizer $\hat{X}$ of (2.12) has to fulfill $f(\hat{X}) \leq f(X_0)$ and $\mathcal{A}\left(\hat{X}\right) = y$ and therefore can be decomposed as the sum of $X_0$ and a perturbation $Z \in \mathcal{D}(f, X_0) \cap \ker(\mathcal{A})$. If the intersection between the nullspace $\ker(\mathcal{A})$ and the descent cone $\mathcal{D}(f, X_0)$ only contains the zero element, $X_0$ is the unique optimal solution of (2.12). This is illustrated in Fig. 2.4 (left).

This clean geometric picture can be extended to the noisy case. In this setting, exact recovery cannot be expected. Therefore, we will bound the reconstruction error $\|\hat{X} - X_0\|_F = \|Z\|_F$ between a feasible minimizer $\hat{X} = X_0 + Z$ of (2.13) and the ground truth $X_0$. Since $\|\mathcal{A}(X_0 + Z) - y\|_2 \leq \tau$ implies that $\|\mathcal{A}(Z)\|_2 \leq 2\tau$, the intersection of $\mathcal{D}(f, X_0)$ and $\{Z : \|\mathcal{A}(Z)\|_2 \leq 2\tau\}$ has to be analyzed, see

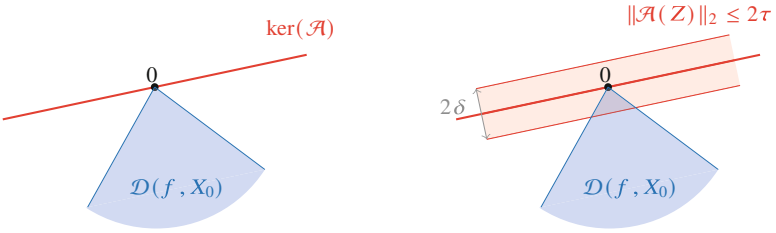



**Fig. 2.4** *Illustration of a descent cone analysis:* the intersection between the nullspace of $\mathcal{A}$ (resp., the set for which $\|\mathcal{A}(Z)\|_2$ is low) with the descent cone $\mathcal{D}(f, X_0)$, i.e., the set of directions $Z$ in which $f$ is decreasing at $X_0$, contains all perturbations $Z$ such that $X_0 + Z$ is a minimizer of the noiseless (resp., noisy) convex optimization problem (left: noiseless and right: noisy)

Fig. 2.4 (right). In order to control the size of this intersection, we will need the following quantity, which we refer to as *smallest conic singular value*:

$$\lambda_{\min}(\mathcal{A}, \mathcal{D}(f, X_0)) := \inf_{Z \in \mathcal{D}(f, X_0) \setminus \{0\}} \frac{\|\mathcal{A}(Z)\|_2}{\|Z\|_F}.$$

If the conic singular value is larger, we expect the intersection to be smaller, and, hence, we should obtain stronger noise bound. This intuition is made precise by the following lemma by [19], see also [85].

**Lemma 2.1** *[19, Proposition 2.2] Let $\mathcal{A} : \mathbb{C}^{n_1 \times n_2} \to \mathbb{R}^m$ be a linear operator and assume that $y = \mathcal{A}(X_0) + e$ with $\|e\|_2 \leq \tau$. Then, any minimizer $\hat{X}$ of the convex optimization problem (2.13) satisfies*

$$\|\hat{X} - X_0\|_F \leq \frac{2\tau}{\lambda_{\min}(\mathcal{A}, \mathcal{D}(f, X_0))}.$$

***Proof Sketch*** By definition, $\lambda_{\min}(\mathcal{A}, \mathcal{D}(f, X_0)) \leq \frac{\|\mathcal{A}(Z)\|_2}{\|Z\|_F} \leq \frac{2\tau}{\|Z\|_F}$ for any feasible $Z$. The first inequality follows from the definition of $\lambda_{\min}(\mathcal{A}, \mathcal{D}(f, X_0))$ and $Z \in \mathcal{D}(f, X_0)$, and the second inequality follows from $\|\mathcal{A}(Z)\|_2 \leq 2\tau$, which concludes the proof. □

In the following, we will discuss applications with various underlying random operators $\mathcal{A}$. We will show how one can obtain lower bounds for the minimum conic singular value, which by Lemma 2.1 will yield recovery guarantees, both in the noise-free and in the noisy case.

### 2.2.2 Application 1: Generic Low-Rank Matrix Recovery

Low-rank matrix recovery describes the problem of recovering a low-rank matrix $X_0 \in \mathbb{R}^{n_1 \times n_2}$ from measurements of the form

$$y_i = \langle A_i, X_0 \rangle \quad \text{where } A_i \in \mathbb{R}^{n_1 \times n_2} \text{ and } i \in [m].$$

It is useful to introduce the measurement operator $\mathcal{A} : \mathbb{R}^{n_1 \times n_2} \to \mathbb{R}^m$ by

$$\mathcal{A}(X)(i) := \langle A_i, X \rangle \quad \text{where } A_i \in \mathbb{R}^{n_1 \times n_2} \text{ and } i \in [m] \tag{2.14}$$

for $X \in \mathbb{R}^{n_1 \times n_2}$.

(This setting can be extended to the complex-valued setting. However, for simplicity of the exposition, we will only discuss the real-valued setting in this section.) In this subsection, we will focus on independent random measurement matrices $A_i$ with independent standard normal entries. In order to recover a low-rank matrix $X_0$, we will consider the convex optimization problems (2.12) and (2.13) with $f = \|\cdot\|_*$.

Recall from the last section that by setting $E := \{Z \in \mathcal{D}(\|\cdot\|_*, X_0) : \|Z\|_F = 1\}$ and bounding $\inf_{Z \in E} \|\mathcal{A}(Z)\|_2$, the smallest conic singular value from below would guarantee that $\mathcal{D}(\|\cdot\|_*, X_0) \cap \ker(\mathcal{A})$ only contains the zero element and, therefore, exact recovery in the noiseless scenario.

Adjusting Fourcart's and Rauhut's formulation of Gordon's escape through a mesh [31, Theorem 9.21] (originally due to Gordon [35]) to the real-valued vector space $\mathbb{R}^{n_1 \times n_2}$, one obtains a powerful lower bound that can exploit the randomness of $\mathcal{A}$.

**Theorem 2.1 (Gordon's Escape Through a Mesh)** *Let $\mathcal{A} : \mathbb{R}^{n_1 \times n_2} \to \mathbb{R}^m$ be a Gaussian measurement operator as defined in (2.14), and let $E$ be a subset of the Frobenius unit sphere $S_F(\mathbb{R}^{n_1 \times n_2}) := \{Z \in \mathbb{R}^{n_1 \times n_2} : \|Z\|_F = 1\}$. Furthermore, define the Gaussian width of $E$ as*

$$\ell(E) := \mathbb{E} \sup_{Z \in E} \langle A, Z \rangle, \tag{2.15}$$

*where $A \in \mathbb{R}^{n_1 \times n_2}$ is a standard normal matrix ($A_{ij} \overset{iid}{\sim} \mathcal{N}(0,1)$). Then, for $t>0$,*

$$\inf_{Z \in E} \|\mathcal{A}(Z)\|_2 \geq \sqrt{m-1} - \ell(E) - t$$

*with probability at least $1 - e^{-t^2/2}$.*

The Gaussian width is actually a reasonable summary parameter for the size of a convex cone. It is also closely related to the statistical dimension [2]. If $\ell(E)$ does not exceed $\sqrt{m-1}$, recovery guarantees can be obtained.

Theorem 2.1 only requires $E$ to be a subset of the Frobenius unit sphere, and, therefore, one is not restricted to a specific descent cone, but one can instead choose the union over all possible descent cones corresponding to rank-$r$ matrices in order to obtain uniform recovery guarantees:

$$E_r = S_F(\mathbb{R}^{n_1 \times n_2}) \cap K_r \quad \text{and} \quad K_r = \bigcup_{X \in \mathbb{R}^{n_1 \times n_2} : \mathrm{rk}(X) = r} \mathcal{D}(\|\cdot\|_*, X)\,.$$

Hölder's inequality yields $\sup_{Z\in E_r}\langle A, Z\rangle \leq \|A\| \sup_{Z\in E_r}\|Z\|_*$. Tight bounds on the operator norm of a standard Gaussian matrix are readily available (more on that later), but it seems plausible that the largest nuclear norm of $Z \in E_r$ could scale unfavorably with the ambient dimension ($\|Z\|_* \leq \sqrt{\min\{n_1, n_2\}}\|Z\|_F$ which is sharp). The geometry of descent cones, however, excludes such worst-case instances. The following lemma highlights that the effective rank of descent cone elements is proportional to the rank of the anchor point. It is a generalization of [64, Lemma 10] to rectangular matrices. To increase accessibility, we write $x \lesssim y$ if there is a positive constant $C > 0$ such that $x \leq Cy$.

**Lemma 2.2** *Suppose that $Z \in \mathbb{C}^{n_1\times n_2}$ is contained in the nuclear norm descent cone of a rank-r matrix $X \in \mathbb{C}^{n_1\times n_2}$. Then,*

$$\|Z\|_* \lesssim \sqrt{r}\|Z\|_F.$$

The suppressed proportionality constant is small ($C \leq 1+\sqrt{2}$), but probably not optimal. The proof is novel and uses ideas from dual certificates (Sect. 2.3), as well as pinching, see, e.g., [5, Problem II.5.4]. We refer to appendix for details. With this lemma at hand, we can bound the Gaussian width of $E_r$.

**Corollary 2.1** *The Gaussian width of $E_r$, the union over all possible descent cones with an anchor point of rank-r, can be bounded by*

$$\ell(E_r) \lesssim \sqrt{r}\left(\sqrt{n_1}+\sqrt{n_2}\right).$$

*Furthermore, let $\mathcal{A} : \mathbb{R}^{n_1\times n_2} \to \mathbb{R}^m$ be a Gaussian measurement operator as defined in (2.14). Then, $\lambda_{\min}(\mathcal{A}, \mathcal{D}(f, X))$ is bounded away from zero for any rank-r matrix X w.h.p. if*

$$m \gtrsim r(n_1+n_2).$$

***Proof Sketch*** Using Hölder's inequality and Lemma 2.2, the Gaussian width $\ell(E_r)$ can be bounded in terms of the expected operator norm of a standard Gaussian matrix:

$$\ell(E_r) = \mathbb{E}\sup_{Z\in E_r}\langle A, Z\rangle \leq \sup_{Z\in E_r}\|Z\|_*\,\mathbb{E}\|A\| \lesssim \sqrt{r}\mathbb{E}\|A\|.$$

A tight upper bound $\mathbb{E}\|A\| \leq (\sqrt{n_1}+\sqrt{n_2})$ can be found, e.g., in [31, p.292]. By Theorem 2.1,

$$\inf_{X\in E_r}\|\mathcal{A}(Z)\|_2 = \inf_{X\in\mathbb{R}^{n_1\times n_2}:\mathrm{rk}(X)=r}\lambda_{\min}(\mathcal{A}, \mathcal{D}(\|\cdot\|_*, X_0)) \geq \sqrt{m-1}-\ell(E)-t$$

with probability at least $1-e^{-t^2/2}$. Therefore, if $m \gtrsim r(n_1+n_2)$, we can pick $t > 0$, such that $\inf_{X\in E_r}\|\mathcal{A}(Z)\|_2$ is positive w.h.p. □

Even when measuring multiple matrices of rank-$r$ via the same measurement operator $\mathcal{A}$, Corollary 2.1 uniformly bounds $\lambda_{\min}(\mathcal{A}, \mathcal{D}(\|\cdot\|_*, X))$ from below and, therefore, gives a uniform recovery guarantee for recovering not only one but all possible rank-$r$ matrices.

### 2.2.3 Application 2: Phase Retrieval

Recall that $\mathbb{H}_n \subset \mathbb{C}^{n\times n}$ denotes the (real-valued) vector space of Hermitian $n \times n$ matrices. The lifted reformulation of the phase retrieval problem is based on the measurement operator

$$\mathcal{A}(X_0)(i) = \langle A_i, X_0\rangle \quad A_i = a_i a_i^* \in \mathbb{H}_n,\ X_0 = x_0 x_0^* \in \mathbb{H}_n,\ i \in [m]\,.$$

This bears strong similarities with the measurement operator for generic low-rank matrix recovery (2.14), but there is one crucial distinction. Each measurement matrix $A_i = a_i a_i^*$ is itself a rank-one orthoprojector. These are everything but generic random matrices (cf. a matrix with standard normal entries is almost surely *not* rank-deficient), and a clean descent cone analysis based on Gordon's escape through a mesh (Theorem 2.1) seems out of reach. Fortunately, Mendelson and co-authors [55, 71] developed a weaker variant of Theorem 2.1. Known as Mendelson's small ball method, this result only requires i.i.d. measurement matrices that also obey a small ball property. We refer to Tropp [85] for a user-friendly exposition and proof and state it directly in terms of measurement operators on Hermitian $n \times n$ matrices.

**Theorem 2.2 (Mendelson's Small Ball Method)** *Suppose that $\mathcal{A} : \mathbb{H}_n \to \mathbb{R}^m$ is a measurement operator (2.2) whose measurements correspond to independent realizations of a Hermitian random matrix $A \in \mathbb{H}_n$. Fix a subset $E \subset \mathbb{H}_n$, and for $\xi > 0$, define*

$$Q_\xi(E; A) = \inf_{Y\in E} \Pr\left[|\langle A, Y\rangle| \geq \xi\right],$$

$$W_m(E; A) = \mathbb{E}\sup_{Y\in E} \langle Y, H\rangle \quad H = \frac{1}{\sqrt{m}}\sum_{i=1}^m \epsilon_i A_i,$$

*where $\epsilon_1, \ldots, \epsilon_m \overset{iid}{\sim} \{\pm 1\}$ is a Rademacher sequence. Then, for any $\xi > 0$ and $t > 0$,*

$$\inf_{Y\in E} \|\mathcal{A}(Y)\|_2 \geq \xi\sqrt{m}\, Q_{2\xi}(E; \Phi) - 2W_m(E; \Phi) - \xi t \tag{2.16}$$

*with probability at least $1 - \mathrm{e}^{-2t^2}$.*

In fact, this statement is valid for all real-valued[1] inner product spaces with finite dimensions. It is worthwhile to point out that for standard normal random matrices $\Phi_1, \dots, \Phi_m \in \mathbb{R}^{n_1 \times n_2}$ and subsets $E$ of the Frobenius unit sphere, this result recovers Theorem 2.1 up to constants. Fix $\xi > 0$ of appropriate size. Then, $E \subset \{Y \in \mathbb{H}_n : \|Y\|_2 = 1\}$ ensures that $\xi Q_{2\xi}(A; E)$ is constant. What is more, $W_m(A, E)$ reduces to the usual Gaussian width (2.15).

We obtain a recovery guarantee for phase retrieval by appropriately analyzing both contributions to Eq. (2.16). Similar to before, we can actually obtain a uniform recovery guarantee by taking into account all possible descent cones in one go:

$$E_1 = \{Y \in \mathbb{H}_n : \|Y\|_F = 1\} \cap K_1, \quad \text{where} \quad K_1 = \bigcup_{x \in \mathbb{C}^n} \mathcal{D}\big(\|\cdot\|_*, xx^*\big). \tag{2.17}$$

Let us start with controlling the empirical width.

**Lemma 2.3 (Empirical Width for Non-generic Phase Retrieval)** *Let $E_1 \subset \mathbb{H}_n$ be the union of descent cones defined in Eq. (2.17) and suppose that $a \in \mathbb{C}^n$ is an isotropic, sub-normalized random vector, i.e., $\mathbb{E}aa^* = Id$, $\|a\|_2 \le \sqrt{2n}$. Then,*

$$W_m(E_1) \lesssim \sqrt{n \log(n)} \quad \textit{provided that } m \lesssim n \log(n). \tag{2.18}$$

The assumption $m \lesssim n \log(n)$ is not essential but will simplify exposition later on. Similar arguments apply to standard complex Gaussian measurement vectors $g \in \mathbb{C}^n$ (which are not sub-normalized) and produce tighter bounds [64]: $W_m(E_1, aa^*) \lesssim \sqrt{n}$ (no $\log(n)$-factor), provided that $m \lesssim n$. The following proof sketch summarizes arguments presented in Ref. [64].

***Proof Sketch (Lemma 2.3)*** We will show the slightly more general bound

$$\mathbb{E}\|H\| \lesssim \sqrt{\max\{m, n \log(n)\}}.$$

Apply Lemma 2.2 to obtain

$$W_m(E_1, A) = \mathbb{E} \sup_{Y \in E_1} \langle Y, H \rangle \lesssim \sup_{Y \in E_1} \|Y\|_* \mathbb{E}\|H\| \le \sqrt{r}\mathbb{E}\|H\|.$$

The remaining expression is an operator norm of a random matrix $H = \frac{1}{\sqrt{m}} \sum_{i=1}^m \epsilon_i a_i a_i^*$ that features two types of randomness. The matrix Khintchine inequality, see, e.g., [31, Exercise 8.6(d)], allows us to trade the Rademacher randomness against an additional square root. More precisely,

$$\mathbb{E}\|H\| = \mathbb{E}_a \mathbb{E}_\epsilon \|H\| \lesssim \mathbb{E}_a \sqrt{\tfrac{\log(n)}{m}} \Big\| \sum_{j=1}^m (a_j a_j^*)^2 \Big\|^{1/2} \lesssim \sqrt{\tfrac{n \log(n)}{m}} \mathbb{E}_a \Big\| \sum_{j=1}^m a_j a_j^* \Big\|^{1/2},$$

[1] Extensions to complex-valued inner product spaces are also possible, see, e.g., [46].

where the last inequality follows from $(a_j a_j^*)(a_j a_j^*) = \|a_j\|_2\, a_j a_j^* \lesssim \sqrt{n}\, a_j a_j^*$. We now face an operator norm of a sum of random matrices $X_j = a_j a_j^*$ that are positive semidefinite and obey $\|X_i\| = \|a_j\|_2^2 \leq 2n$ each. Isotropy also asserts $\left\| \sum_{j=1}^m \mathbb{E} X_j \right\| = \|m\mathrm{Id}\| = m$, and we can apply the matrix Chernoff inequality [84] to obtain for any $\tau > 0$

$$\mathbb{E}_a \left\| \sum_{j=1}^m a_j a_j^* \right\| \leq \tfrac{\mathrm{e}^\tau - 1}{\tau} m + \tfrac{\sqrt{2}}{\tau} n \log(n) \lesssim \max\{m, n\log(n)\}.$$

□

The empirical width bound (2.18) suggests that an order of $n\log(n)$ non-generic phaseless measurements may suffice to establish strong uniform recovery guarantees for phase retrieval via low-rank matrix reconstruction. However, this is only true if the measurement matrices $a_i a_i^*$ are not too spikey. More precisely, we need that $Q_{2\xi}(aa^*, E_1)$—the second quantity in Mendelson's small ball method (2.16)—is lower bounded by a constant.

**Lemma 2.4 (Marginal Tail Function for Non-generic Phase Retrieval)** *Suppose $a \in \mathbb{C}^n$ is a random vector that obeys $\mathbb{E}\langle a, Ya\rangle^2 \gtrsim \langle Y, Y\rangle$ and $\mathbb{E}\langle a, Ya\rangle^4 \lesssim \left(\mathbb{E}\langle a, Ya\rangle^2\right)^2$ for all $Y \in E_1$. Then,*

$$Q_{2\xi}(E_1; aa^*) \gtrsim \left(1 - \tfrac{4\xi^2}{\mathrm{const}}\right)^2 \quad \textit{for all} \quad 0 < \xi < \sqrt{\mathrm{const}/4}.$$

***Proof*** Fix $Y \in E_1$ and use $\mathbb{E}\langle a, Ya\rangle^2 \gtrsim \langle Y, Y\rangle = \mathrm{const}$ to apply a Paley–Zygmund type argument:

$$\Pr\left[\left|\langle aa^*, Y\rangle\right| \geq 2\xi\right] \geq \Pr\left[\langle a, Ya\rangle^2 \geq \tfrac{4\xi^2}{\mathrm{const}} \mathbb{E}\langle a, Ya\rangle^2\right] \geq \left(1 - \tfrac{4\xi^2}{\mathrm{const}}\right)^2 \tfrac{\left(\mathbb{E}\langle a, Ya\rangle^2\right)^2}{\mathbb{E}\langle a, Y, a\rangle^4}.$$

The moment assumption $\mathbb{E}\langle a, Ya\rangle^4 \lesssim \left(\mathbb{E}\langle a, Ya\rangle^2\right)^2$ ensures that the final ratio is lower bounded by a constant. Such a lower bound is valid, regardless of $Y \in E_r$. Hence, it also applies to the infimum $Q_{2\xi} = \inf_{Y \in E_1} \Pr\left[|\langle aa^*, Y\rangle| \geq 2\xi\right]$. □

We now have gathered all the auxiliary statements we need to carry out a descent cone analysis for phase retrieval with non-generic measurements.

**Theorem 2.3 (Phase Retrieval from Non-generic Measurements)** *Let $a \in \mathbb{C}^n$ be a random vector that is isotropic ($\mathbb{E}aa^* = Id$) and sub-normalized ($\|a\|_2 \leq \sqrt{2n}$) and also obeys*

$$\mathbb{E}\langle a, Ya\rangle^2 \gtrsim \langle Y, Y\rangle, \quad \textit{as well as} \quad \left(\mathbb{E}\langle a, Ya\rangle^2\right)^2 \lesssim \mathbb{E}\langle a, YA\rangle^4, \tag{2.19}$$

*for every $Y \in K_1$. Then, with high probability, a total of*

$$m \gtrsim n \log(n)$$

*randomly selected phaseless measurements* $a_1, \ldots, a_m \sim a \in \mathbb{C}^n$ *suffice to reconstruct signals* $x_0 \in \mathbb{C}^n$ *via constrained nuclear norm minimization* (2.3).

In fact, this recovery guarantee is actually *uniform*. That is, with high probability, a single collection of randomly sampled phaseless measurements allows for reconstructing *all* phaseless signals via nuclear norm minimization (2.3). Conditioned on this event, the actual reconstruction is also stable with respect to noise corruption. Suppose that $y = \mathcal{A}(xx^*) + e$, where $\|e\|_{\ell_2} \leq \tau$ and the noise bound is known. Then, the solution $\hat{X}$ of the convex optimization problem (2.3) is guaranteed to obey $\|\hat{X} - x_0 x_0^*\|_F \lesssim \tau/\sqrt{m}$. Up to constants, this assertion is on par with some of the strongest stability guarantees for low-rank matrix reconstruction in general [9, 13, 47].

***Proof Sketch (Theorem 2.3)*** Let us start by reformulating phase retrieval as a low-rank matrix recovery problem ($r = 1$). The general descent cone analysis presented in Sect. 2.2.1 identifies the minimum conic singular value as an important summary parameter. If it is positive, the current set of measurements allows to recover $X_0 = x_0 x_0^*$ via nuclear norm minimization under idealized circumstances (no noise). The size of the minimum conic singular value also captures noise robustness (the larger the better). Theorem 2.2 (Mendelson's small ball method) achieves just that. Fix $\xi =$ const sufficiently small and insert the bounds from Lemma 2.4 and Lemma 2.3 into the assertion of Theorem 2.2:

$$\begin{aligned}\inf_{Y \in E_1} \|\mathcal{A}(Y)\|_2 \geq & \xi \sqrt{m} Q_{2\xi}\left(E_1; aa^*\right) - 2W_m\left(E_1; aa^*\right) - \xi t \\ \gtrsim & \sqrt{m} - \text{const}\left(\sqrt{n \log(n)} + t\right),\end{aligned}$$

with probability at least $1 - \mathrm{e}^{-2t^2}$. Assigning $m = Cn \log(n)$ and $t = \gamma/2\sqrt{m}$, where $C > 0$ ($\gamma > 0$) is a sufficiently large (small) constant, allows us to conclude $\inf_{Y \in E_1} \|\mathcal{A}(Y)\|_2 \gtrsim \sqrt{m}$ with probability at least $1 - \mathrm{e}^{-\gamma\sqrt{m}}$. This ensures that the minimum conic singular value is of (optimal) order $\sqrt{m}$.

There is one additional twist. In Eq. (2.17), we have defined the set $E_1$ as the union of all possible descent cones anchored at all possible lifted signals $X = xx^*$. Consequently, Theorem 2.2 produces a lower bound of $\sqrt{m}$ on the infimum over all possible descent cones, not just a single one. This allows us to effectively treat all possible signals at once and establish a uniform recovery guarantee. □

Let us conclude this section with discussing the extra assumptions (2.19). They formulate conditions on the second and fourth moments of the measurement matrices $A = aa^*$. The second moment condition ensures that the expected measurement operator is non-singular on the union $K_1$ of all descent cones:

$$\tfrac{1}{m}\mathbb{E}\langle Y, \mathbb{E}\mathcal{A}^*\mathcal{A}(Y)\rangle = \mathbb{E}\langle aa^*, Y\rangle_F = \mathbb{E}\langle a, Ya\rangle^2 \gtrsim \langle Y, Y\rangle \quad \text{for all } Y \in K_1. \tag{2.20}$$

Viewed from this angle, it actually captures (sub-)isotropy on the relevant parts of $\mathbb{H}_n$—a natural requirement for any low-rank matrix recovery procedure. Alas, by itself, it is not sufficient to derive nontrivial recovery guarantees [37, 62] and extra assumptions are required. Theorem 2.3, for instance, requires that (certain) fourth moments of $A = aa^*$ are comparable to their second moment squared. It should be viewed as a relaxation of (sub-)Gaussian moment growth conditions, but only up to order four. Suitable measurement ensembles only need to mimic (outer products of) Gaussian measurement vectors up to fourth moments. This condition is much weaker than sub-Gaussianity, and vector distributions that satisfy Eq. (2.19) can admit a lot of structure. A concrete example is orbits of certain symplectic symmetry groups that arise naturally in quantum information (Clifford group) and time–frequency analysis (oscillator group) [65]. A more refined analysis also allows for replacing constrained nuclear norm minimization (2.3) by a simple least-squares or $\ell_p$-fit over the cone of positive semidefinite matrices [47], such as the convex optimization problem

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{m} \left| \operatorname{tr}\left(\xi^{(i)}(\xi^{(i)})^* X\right) - y_i \right| \\ \text{subject to} \quad & X \in \mathcal{S}_+^n, \end{aligned} \tag{2.21}$$

where $\mathcal{S}^n \subset \mathbb{R}^{n\times n}$ denotes the set of real-valued symmetric matrices and $\mathcal{S}_+^n \subseteq \mathcal{S}^n$ its positive definite subset. Such reformulations have the added benefit of being tuning-free. In particular, no a priori noise bound $\tau$ is required, see [63] for related arguments addressing sparse vector recovery and [27] sparse covariance matching.

### 2.2.4 Limitations

As we have seen, a descent cone analysis combined with probabilistic tools such as Mendelson's small ball method yields essentially near-optimal uniform recovery results for low-rank matrix recovery from Gaussian measurement matrices or phase retrieval measurements with Gaussian measurement vectors. A key observation of the proof is that the union of all descent sets is contained in a suitably large nuclear norm ball, so it suffices to estimate the Gaussian width of this ball.

This approach, however, has significant limitations when it comes to problems with more structure such as matrix completion and blind deconvolution. The reason is that in these problems, as explained in Sect. 2.1.1.1, recovery guarantees will necessarily fail for some exceptional signals that violate certain incoherence conditions. Thus it will necessarily be impossible to bound the minimum conic singular values for the descent cones anchored at these signals and estimating a general superset cannot be sufficient.

However, one may wonder whether it is possible to obtain a comparable result to Corollary 2.1 and Theorem 2.3 by considering *the union of all descent cones of all incoherent rank-r matrices* instead. However, this turns out to be more delicate. In particular, it is unclear how to mathematically formulate a property that captures the fact that matrices in the descent cone anchored at incoherent signals are better conditioned with respect to the measurements. A direct connection to the notion of incoherence is difficult, as matrices in the descent cone anchored at incoherent signals will not necessarily be incoherent. As a consequence, also the minimum conic singular values can become provably very small [58, 60], which makes it difficult to bound them from below, which would be necessary for recovery guarantees based on the strategy explained above even for the noiseless case.

In the next section, we present an alternative analysis strategy that is better suited to deal with incoherence conditions, as it is based on (approximate) dual certificates rather than the descent cone and relies on the signal alone rather than differences to alternative solutions. In certain cases, however, as we will see in Sect. 2.4, it will also be possible to adapt the descent cone analysis to such scenarios.

## 2.3 Recovery Guarantees via the Golfing Scheme

### *2.3.1 Recovery Guarantees via Dual Certificates*

Maybe the most natural way of proving that a convex optimization attains its optimal value at a given argument is by exhibiting a *dual certificate* – the generalization to possibly non-smooth convex functions of the familiar gradient condition for optimality. Let us start by considering the noiseless nuclear norm problem ($\tau = 0$)

$$\begin{aligned} \underset{X \in \mathbb{C}^{n_1 \times n_2}}{\text{minimize}} \quad & \|X\|_* \\ \text{subject to} \quad & \mathcal{A}(X) = y, \end{aligned} \tag{2.22}$$

see also Eq. (2.12) with $f(X) = \|X\|_*$. Let $X \in \mathbb{C}^{n_1 \times n_2}$ be a rank-$r$ matrix with singular value decomposition (SVD) $X = U \Sigma V^*$. That is, $\Sigma \in \mathbb{R}^{r \times r}$ is a diagonal matrix with nonnegative entries and $U \in \mathbb{C}^{n_1 \times r}$ and $V \in \mathbb{C}^{n_2 \times r}$ are isometries, i.e., $U^*U = V^*V = \mathrm{Id}_r$. The tangent space of the variety of rank-$r$ matrices at the point $X$ can be checked to be given by

$$T_X := \left\{ UA^* + BV^* : \ A \in \mathbb{C}^{n_2 \times r}, B \in \mathbb{C}^{n_1 \times r} \right\}. \tag{2.23}$$

Denote by $\mathcal{P}_{T_X}$ the (Hilbert-Schmidt) orthogonal projection onto the tangent space and by $\mathcal{P}_{T_X^\perp}$ the projection onto its orthocomplement. The *subdifferential* $\partial \|\cdot\|_* (X)$

of the nuclear norm at $X$ is the set of affine lower bounds to the nuclear norm that coincide with the norm at $X$. A simple application of the matrix Hölder inequality [5] shows that [90]

$$\partial \|\cdot\|_* (X) = \left\{ W \in \mathbb{C}^{n_1 \times n_2} : \mathcal{P}_{T_X} W = UV^*, \ \|\mathcal{P}_{T_X^\perp} W\| \leq 1 \right\}. \tag{2.24}$$

With these notions, it is straightforward to see that a sufficient condition for $X_0$ being the minimizer of (2.22) is given by the following lemma, first formulated in Ref. [14].

**Lemma 2.5 ([14])** *Let $X_0 \in \mathbb{C}^{n_1 \times n_2}$ be such that $\mathcal{A}(X_0) = y \in \mathbb{C}^m$. Suppose that the following two conditions hold:*

1. *There exists a vector $z \in \mathbb{C}^m$ such that $Y = \mathcal{A}^*(z)$ satisfies*

$$\mathcal{P}_{T_{X_0}} Y = UV^* \quad \text{and} \quad \|\mathcal{P}_{T_{X_0}^\perp} Y\| < 1.$$

2. *The linear operator $\mathcal{A}$ is injective when restricted to the tangent space $T_{X_0}$.*

*Then, $X_0$ is the unique minimizer of* (2.22).

In Ref. [14], it was shown in the context of low-rank matrix completion from a sufficient number of uniformly sampled matrix elements that such a dual certificate exists with high probability. A refined (and fairly involved) analysis in Ref. [18] showed that the number of measurements can be reduced to the order of the information-theoretic limit, up to logarithmic factors.

Reference [36] introduced a new approach—the *golfing scheme*—for constructing dual certificates. In the original paper, and commonly in works referring to it, the result is presented as being based on the observation that the conditions in Lemma 2.5 can be relaxed and that the existence of an *approximate dual certificate* suffices to establish uniqueness. Approximate dual certificates are easier to construct using randomized processes, which in their natural formulations will give results that are correct only approximately and up to a small probability of failure.

In this chapter, we aim to present the story from a different point of view. Namely, we will show that a minor tweak of the golfing scheme actually gives an explicit randomized construction for an *exact* dual certificate, using no more measurements than the original argument. In this sense, it is inaccurate to say that constructing exact certificates is harder than constructing approximate ones. While this point of view does not seem to impact the headline result on recovery guarantees, we feel that it represents a conceptually clearer way of thinking about the argument. To the best of our knowledge, this approach has not appeared elsewhere in the literature before.

### 2.3.2 Golfing with Precision

Here, we recall the basic logic behind the golfing scheme, in preparation of presenting the *putting proposition*, Proposition 2.2. We start with two definitions.

The analysis uses the fact that the measurement operator $\mathcal{A}$ is an approximate isometry when restricted to the subspace $T_X$. The precise notion employed is this:

**Definition 2.1** Let $X \in \mathbb{C}^{n_1 \times n_2}$. We say that $\mathcal{A}$ fulfills the $\delta$-restricted isometry property ($\delta$-RIP) on $T_X$, if for all matrices $Z \in T_X$, it holds that

$$(1-\delta)\,\|Z\|_F^2 \leq \|\mathcal{A}(Z)\|_2^2 \leq (1+\delta)\,\|Z\|_F^2.$$

As the name suggests, approximate dual certificates obey condition 1 in Lemma 2.5 approximately. This is captured by the following formal definition.

**Definition 2.2** Given a measurement operator $\mathcal{A} : \mathbb{C}^{n_1 \times n_2} \to \mathbb{C}^m$, a vector $z \in \mathbb{C}^m$, giving rise to a matrix $Y = \mathcal{A}^*(z)$, is an *approximate dual certificate* at $X_0 = U\Sigma V^*$ if it satisfies the following properties:

$$\|z\|_2 \leq 2, \tag{2.25}$$

$$\alpha = \|UV^* - \mathcal{P}_{T_{X_0}}\mathcal{A}^*(z)\|_F \leq \tfrac{1}{8\|\mathcal{A}\|}, \tag{2.26}$$

$$\left\|\mathcal{P}_{T_{X_0}^{\perp}}\left(\mathcal{A}^*(z)\right)\right\| < \tfrac{1}{2}. \tag{2.27}$$

With these definitions, the central result reads as follows:

**Proposition 2.1** *[12, 36] Let $X_0 \in \mathbb{C}^{n_1 \times n_2}$ with SVD $X_0 = U\Sigma V^*$, and suppose that $y = \mathcal{A}(X_0) + e$ with $\|e\|_2 \leq \tau$. Suppose that the following two conditions hold:*

1. *There exists an approximate dual certificate $Y = \mathcal{A}^*(z)$*
2. *The measurement operator $\mathcal{A}$ satisfies the $\delta$-restricted isometry property on $T_{X_0}$ with constant $\delta = 3/4$*

*Then, every minimizer $\hat{X}$ of (2.22) satisfies*

$$\|X_0 - \hat{X}\|_F \lesssim \|\mathcal{A}\|\tau. \tag{2.28}$$

Here, $\|\mathcal{A}\| = \sup_{\|z\|_2=1}\|\mathcal{A}(z)\|_F$ is the operator norm of the measurement operator. The requirement (2.25) on the norm of $z$ is only necessary in the noisy case $\|e\|_2 > 0$.

The bound (2.28) is not always tight. For example, let $\mathcal{A}$ be the Gaussian measurement operator defined in Sect. 2.2.2. For $m \ll n_1 n_2$, $\|\mathcal{A}\| \asymp \sqrt{n_1 n_2}$ with high probability. This is larger than the optimal error scaling $\|X_0 - \hat{X}\|_F \propto \sqrt{m}$ for this regime and measurement model.

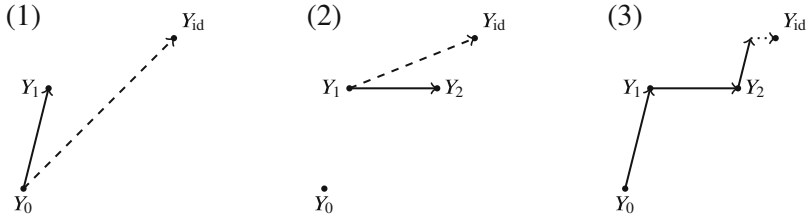



**Fig. 2.5** *Construction of an approximate dual certificate via golfing:* for the $p$th leg, we start with a current best guess $Y_p$ (cf. panels (1) and (2) above). On the tangent space $T_{X_0}$, we aim to express the difference $\Delta_p := Y_{\mathrm{id}} - \mathcal{P}_{T_{X_o}} Y_p$ (dotted line) in terms of rows from the partial measurement matrix $\mathcal{A}^p$. Here, $Y_{\mathrm{id}} = UV^*$ is the "ideal" dual certificate, which is an element of the tangent space. If the rows of $\mathcal{A}^p$ were an orthonormal basis, then $\Delta_p = (\mathcal{A}^p)^* \mathcal{A}^p (Y_{\mathrm{id}})$ would give an exact solution. If $\mathcal{A}^p$ is subsampled from an orthonormal basis, standard measure concentration results imply that on the tangent space, we will obtain a relatively decent approximation for $\Delta_p$ (solid line). In fact, if the number of rows in $\mathcal{A}^p$ is sufficient, one can easily show that the distance to the ideal certificate will be reduced by a constant factor with high probability. It is then natural to just iterate the scheme (panel (3)). This results in a random process which converges in Frobenius norm to the ideal certificate (on the tangent space) exponentially quickly. At the same time, on the space orthogonal to the tangent space, we have that $\mathbb{E}\big(\mathcal{P}_{T_{X_0}^\perp}(\mathcal{A}^*(z))\big) = 0$. Again using concentration of measure results, one can show that, during the logarithmically many legs of the golfing procedure, the spectral norm of these terms remains small

Before proving this statement, we sketch the idea behind the *golfing scheme* [36] for the construction of an approximate dual certificate (cf. Fig. 2.5).

The ensemble of measurement vectors will often be *isotropic* in the sense that $\mathbb{E}\left[\mathcal{A}^*\mathcal{A}\right] = \mathrm{Id}$. This motivates the choice $\tilde{z}_1 = \mathcal{A}(UV^*)$ and $\tilde{Y}_1 = \mathcal{A}^*(z) = \mathcal{A}^*\mathcal{A}(UV^*)$ for $z$ and $Y$, as it leads to the correct result $\mathbb{E}\left[\mathcal{A}^*(\tilde{z}_1)\right] = UV^*$ *in expectation*. Consequently, one could then hope to show properties (2.25), (2.26), and (2.27) using measure concentration around the mean. Unfortunately, this approach does not usually work directly. One problem is that the operator norm $\|\mathcal{A}\|$ can be quite large (for blind deconvolution $\|\mathcal{A}\|$, it is of the order $\sqrt{KN/L}$). This, in turn, means that $\left\| UV^* - \mathcal{P}_{T_{X_0}} \mathcal{A}^*(z) \right\|_F$ needs to be small, smaller than typical fluctuations. The idea behind the golfing scheme is to iteratively refine this initial guess until condition (2.26) is satisfied:

- **Step 1:** Choose a partition of $[m]$ into $Q$ disjoint sets $\left\{\Gamma_1, \ldots, \Gamma_Q\right\}$ of size roughly $|\Gamma_q| \approx m/Q$, such that $Q\mathbb{E}\left[(\mathcal{A}^q)^*\mathcal{A}^q\right] \approx \mathrm{Id}$, where $\mathcal{A}^q := Q_{\Gamma_q}\mathcal{A}$. (Here, $Q_{\Gamma_q} : \mathbb{C}^m \to \mathbb{C}^m$ denotes the coordinate projection onto $\Gamma_q$.)
- **Step 2:** Set

$$Y_0 = 0 \quad \text{and}$$

$$Y_q = Y_{q-1} + Q(\mathcal{A}^q)^*\mathcal{A}^q \left(UV^* - \mathcal{P}_{T_{X_0}} Y_{q-1}\right) \qquad \text{where } 1 \le q \le Q.$$

The corresponding $z \in \mathbb{C}^m$ is then given by

$$z := Q \sum_{q=1}^{Q} \mathcal{A}^q \left( UV^* - \mathcal{P}_{T_{X_0}} Y_{q-1} \right).$$

Note that a consequence of the sample splitting in Step 1 is that the golfing scheme is set up in such a way that the distribution of $\mathcal{A}^q$ is independent of $Y_{q-1}$. This simplifies the analysis but is not essential [39].

The precise convergence properties of this random process depend on the parameters (partition size, incoherence, etc. [36]) and is, in any case, beyond the scope of this article. Instead, we want to make precise the following new observation—which, in keeping with the theme, we call the *putting proposition*. For more context, see the discussion at the end of Sect. 2.3.1.

**Proposition 2.2 ("Putting Proposition")** *Assume that the approximate dual certificate properties (2.25)-(2.27) hold and that $\mathcal{A}$ fulfills the $\delta$-restricted isometry property on $T_{X_0}$ for $\delta < 3/4$. Then, there exists an exact dual certificate for $X_0$.*

***Proof*** Using the variational characterization of the operator norm of a Hermitian linear map, as well as the definition of the $\delta$-RIP, we get

$$\begin{aligned} \left\| P_{T_{X_0}} \mathcal{A}^* \mathcal{A} P_{T_{X_0}} - P_{T_{X_0}} \right\| &= \sup_{Z \in T_{X_0}, \|Z\|_F = 1} \left| (Z, \mathcal{A}^* \mathcal{A} Z) - 1 \right| \\ &= \sup_{Z \in T_{X_0}, \|Z\|_F = 1} \left| \|\mathcal{A} Z\|_F^2 - 1 \right| \leq \delta. \end{aligned}$$

Hence, as a linear map on the tangent space, $P_{T_{X_0}} \mathcal{A}^* \mathcal{A} P_{T_{X_0}}$ is invertible and satisfies

$$\left\| \left( P_{T_{X_0}} \mathcal{A}^* \mathcal{A} P_{T_{X_0}} \right)^{-1} \right\| \leq \tfrac{1}{1-\delta}.$$

Set

$$x = \mathcal{A} P_{T_{X_0}} \left( P_{T_{X_0}} \mathcal{A}^* \mathcal{A} P_{T_{X_0}} \right)^{-1} \left( UV^* - \mathcal{P}_{T_{X_0}} \mathcal{A}^*(z) \right).$$

Together with (2.26), this gives

$$\|x\|_2 \leq \tfrac{1}{\sqrt{1-\delta}} \left\| UV^* - \mathcal{P}_{T_{X_0}} \mathcal{A}^*(z) \right\|_F \leq \tfrac{1}{8\sqrt{1-\delta}\|\mathcal{A}\|}.$$

But then, with $Y' = \mathcal{A}^*(z + x) = Y + \mathcal{A}^*(x)$, we have that

$$\mathcal{P}_{T_{X_0}}(Y') = UV^*$$

and

$$\begin{aligned}\left\|\mathcal{P}_{T_{X_0}^{\perp}} Y'\right\| &\leq \left\|\mathcal{P}_{T_{X_0}^{\perp}} Y\right\| + \left\|\mathcal{P}_{T_{X_0}^{\perp}} \mathcal{A}^*(x)\right\| \\ &\leq \frac{1}{2} + \left\|\mathcal{P}_{T_{X_0}^{\perp}} \mathcal{A}^*(x)\right\|_F \\ &\leq \frac{1}{2} + \|\mathcal{A}\| \|x\|_2 \\ &\leq \frac{1}{2} + \frac{1}{8\sqrt{1-\delta}} < 1.\end{aligned}$$

□

### 2.3.3 Application 3: Matrix Completion

Using dual certificate-based proof techniques, the nuclear norm minimization approach to matrix completion has been studied extensively [14, 18, 20, 36, 76]. A typical result for the noiseless case ($\tau = 0$) reads as follows:

**Theorem 2.4 ([20])** *Assume that $n_1 \geq n_2$. Consider measurements of the form $y = \mathcal{A}(X_0)$, where $X_0 \in \mathbb{R}^{n_1 \times n_2}$ is a rank-$r$ matrix and $\mathcal{A}$ is given by (2.4). Assume that*

$$m \geq C \max\left\{\mu^2(U), \mu^2(V)\right\} r n_1 \log^2 n_1.$$

*Then, with high probability, the matrix $X_0$ is the unique minimizer of SDP (2.22) (see also SDP (2.3) with $\tau = 0$).*

Further variants have been studied in the literature. For example, if the noise term is drawn randomly instead of adversarially, improved results can be given, see Refs. [53, 54] for subexponential and Ref. [22] for sub-Gaussian noise. Non-convex algorithms with rigorous performance guarantees can be found in Refs. [30, 32, 43, 50, 51, 66, 70, 83]).

### 2.3.4 Application 4: Simultaneous Demixing and Blind Deconvolution

Simultaneous blind deconvolution and demixing is a generalization of the blind deconvolution problem introduced in Sect. 2.1.1.2. It is motivated by wireless communication scenarios that involve multiple senders, but only one receiver. Each sender wants to transmit a signal $m_i$ using a linear encoder $C_i$. The encoded signal

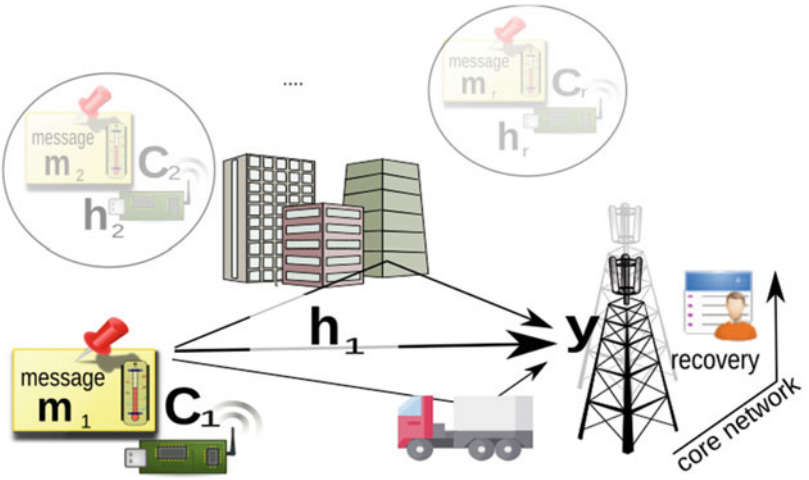



**Fig. 2.6** *A multi-user wireless (uplink) communication scenario:* wireless devices $i = 1, \dots, r$ simultaneously transmit messages $m_i$ to the base station which are individually encoded with a linear code $C_i$ and experience individual convolutional channels $h_i$

$x_i = C_i m_i$ is sent through an unknown convolution channel $w_i$ to the receiver. Because there are multiple senders, the receiver obtains the superposition of $r$ convolutions, where the goal is to reconstruct all messages $\{m_i\}_{i=1}^r$.

In mathematical terms, this leads to an inverse problem of the form

$$y = \sum_{i=1}^{r} w_i * x_i + e \in \mathbb{C}^L, \tag{2.29}$$

where $*$ denotes the (circular) convolution introduced in Eq. (2.6). The goal is to simultaneously reconstruct all signals $x_i$, as well as all channel descriptions $w_i$. As in the randomized blind deconvolution framework, we have to use some prior knowledge on $w_i$ and $x_i$ in order to be able to reconstruct these signals. We are going to adopt the framework introduced in Ref. [69]. Assume that $w_i$ and $x_i$ are elements of known subspaces. Hence, we can write $w_i = Bh_i$ and $x_i = C_i m_i$ for all $i \in [r]$, where $B \in \mathbb{C}^{L \times K}$ and $C_i \in \mathbb{C}^{L \times N}$, see Fig. 2.6. We assume that $B^* B = \mathrm{Id}$ and, moreover, that for each $i \in [r]$, the entries of the matrix $C_i$ are i.i.d. samples from the complex normal distribution $\mathcal{CN}(0, 1)$.

Similar to the randomized blind deconvolution setting, we note that for each $i \in [r]$ there is a unique linear operator $\mathcal{A}_i : \mathbb{C}^{K \times N} \to \mathbb{C}^L$ such that for all $u \in \mathbb{C}^K$ and $v \in \mathbb{C}^N$ it holds that

$$\mathcal{A}_i \left( uv^* \right) = Bu * C_i \overline{v}.$$

Hence, we can rewrite Eq. (2.29) as

$$y = \sum_{i=1}^{r} \mathcal{A}_i \left( h_i m_i^* \right) + e.$$

We emphasize that each outer product $h_i m_i^*$ comes with a unique linear operator $\mathcal{A}_i$. This allows us to recast Eq. (2.29) as a low-rank matrix recovery problem on a larger space. Our goal is to recover the block-diagonal rank-$r$ matrix

$$X_0 = h_1 m_1^* \oplus h_2 m_2^* \oplus \cdots \oplus h_r m_r^*$$

from a linear measurement operator that decomposes accordingly ($\mathcal{A}(Z_1 \oplus \cdots \oplus Z_r) = \sum_{i=1}^{r} \mathcal{A}_i(Z_i)$). Adapting SDP (2.3) to this problem structure yields

$$\begin{aligned} \underset{X_1,\ldots,X_r \in \mathbb{C}^{K\times N}}{\text{minimize}} \quad & \sum_{i=1}^{r} \|X_i\|_* \\ \text{subject to} \quad & \|y - \sum_{i=1}^{r} \mathcal{A}_i\left(X_i\right)\|_2 \leq \tau, \end{aligned} \tag{2.30}$$

see [69]. Furthermore, denote by $\mu_{\max}^2$ and $\mu_h^2$ the coherence parameters, which are similar to the ones defined in Sect. 2.1.1.2. (For a precise definition, we refer to [45].) In [69], it has been shown that if

$$L \gtrsim r^2 \left( K \mu_{\max}^2 + N \mu_h^2 \right) \log^3 L \tag{2.31}$$

holds, then in the noiseless scenario, i.e., $e = 0$, the convex relaxation (2.30) recovers the ground truth matrix $X_0$ with high probability.

However, we observe that the number of degrees of freedom in this problem is $r\,(K + N - 1)$, which raises the question, whether the quadratic dependence on $r$ in (2.31) is necessary. Indeed, numerical experiments in [69] indicate that the true dependence of the sample complexity in $r$ should rather be linear (see [69, Section IV] as well as [45, Section III]).

The main result in [45] shows that the required simple complexity is indeed linear in $r$. Hence, nuclear norm minimization can recover the ground truth signal $X_0$ at near-optimal sample complexity.

**Theorem 2.5 ([45] and see also [80–82])** *Let $y \in \mathbb{C}^L$ be given by (2.29) with $\|e\|_2 \leq \tau$. Assume that*

$$L / \log^3 L \gtrsim r \left( K \mu_{\max}^2 \log \left( K \mu_{\max}^2 \right) + N \mu_h^2 \right).$$

*Then, with high probability, every minimizer $\hat{\mathbf{X}} = \hat{X}_1 \oplus \ldots \oplus \hat{X}_r$ of SDP (2.30) satisfies*

$$\sqrt{\sum_{i=1}^{r} \left\| \hat{X}_i - h_i m_i^* \right\|_F^2} \lesssim \tau \sqrt{rN}.$$

In the following, we are going to describe the main technical ingredient, which allowed for linear scaling in $r$. In both [69] and [45], the proofs establish the existence of an approximate dual certificate with high probability. One ingredient is to show that the measurement operator acts as an approximate isometry operator on the tangent space of $X_0$, see Definition 2.1. To make this precise in the blind demixing scenario, define, for $i \in [r]$, the tangent space $T_i$ of rank-1 matrices at $h_i m_i^*$:

$$T_i = \left\{ h_i u_i^* + v_i m_i^* : \ u_i \in \mathbb{C}^K, v_i \in \mathbb{C}^N \right\}.$$

Then, we can define the tangent space at $X_0$ by

$$\tilde{T} := \{ X_1 \oplus \ldots \oplus X_r : \ X_i \in T_i \quad \text{for all } i \in [r] \}.$$

In both [69] and [45], one part of the proof consists in showing that, with high probability, the collection of measurement operators $\{\mathcal{A}_i\}_{i=1}^r$ fulfills a local isometry property on $\tilde{T}$. That is, for a sufficiently small $\delta > 0$,

$$(1-\delta) \|X\|_F^2 \leq \left\| \sum_{i=1}^{r} \mathcal{A}_i (X_i) \right\|_2^2 \leq (1+\delta) \|X\|_F^2 \quad \text{for all } X = X_1 \oplus \ldots \oplus X_r \in \tilde{T}. \tag{2.32}$$

In [69], the restricted isometry property is first shown individually on each $T_i$ and after that is shown that the images of the subspaces $T_i$ under the operator $\mathcal{A}_i$ are sufficiently near-orthogonal to each other. Combining these two properties yields (2.32). However, the second step requires that $L$ scales quadratically in $r$.

In contrast, our analysis establishes the restricted isometry property directly on $\hat{T}$. For that, we define

$$\hat{T} := \left\{ X_1 \oplus \ldots \oplus X_r \in \tilde{T} : \sum_{i=1}^{r} \|X_i\|_F^2 = 1 \right\}.$$

Next, we observe that (2.32) is equivalent to

$$\begin{aligned} \delta &\geq \sup_{X_1 \oplus \ldots \oplus X_r \in \hat{T}} \left| \left\| \sum_{i=1}^{r} \mathcal{A}_i (X_i) \right\|_2^2 - \sum_{i=1}^{r} \|X_i\|_F^2 \right| \\ &= \sup_{X_1 \oplus \ldots \oplus X_r \in \hat{T}} \left| \left\| \sum_{i=1}^{r} \mathcal{A}_i (X_i) \right\|_2^2 - \mathbb{E}\left[ \left\| \sum_{i=1}^{r} \mathcal{A}_i (X_i) \right\|_2^2 \right] \right|. \end{aligned}$$

The key idea is that the last expression can be interpreted as a suprema of chaos processes, and we can use deep results from empirical process theory [57] to bound this expression with high probability.

### 2.3.5 Phase Retrieval with Incoherence

Recall that in the phase retrieval problem we are interested in reconstructing a signal $x_0$ from measurements of the form

$$y_k = |\langle a_k, x_0 \rangle|^2 + e_k. \tag{2.33}$$

We have seen in Sect. 2.2.3 that this problem can be solved not only for Gaussian measurement vectors $\{a_i\}$ but also for measurement vectors that are less generic. Nevertheless, the required assumptions are somewhat more restrictive than, for example, in compressive sensing. In particular, for measurement vectors with unimodular entries, the problem does not even have a unique solution.

To see that, assume that for all $k$, the entries of the vector $a_k$ have all the same modulus, i.e.,

$$|\,(a_k)_1\,| = |\,(a_k)_2\,| = \ldots = |\,(a_k)_n\,|. \tag{2.34}$$

In this case, both the vectors

$$\begin{aligned} x_1 &:= (1, 0, \ldots, 0) \in \mathbb{R}^n \\ x_2 &:= (0, 1, \ldots, 0) \in \mathbb{R}^n \end{aligned} \tag{2.35}$$

lead to the same measurements, i.e.,

$$|\langle a_i, x_1 \rangle|^2 = |\langle a_i, x_2 \rangle|^2 \quad \text{for all } i \in [m]\,.$$

Hence, $x_1$ and $x_2$ cannot be distinguished based on phaseless measurements alone. We want to stress that condition (2.34) holds for several interesting classes of measurement vectors. For example, this condition is fulfilled if the entries $a_k$ are Rademacher random variables, i.e., $(a_k)_i$ is either 1 or $-1$, each with probability $1/2$. Moreover, if each entry $(a_k)_i$ is a random variable with uniform distribution over $S^1 \subset \mathbb{C}$, this condition would also be fulfilled.

Another example, which is important for certain applications, is given by random masks [10, 38]. That is, the measurement vector $a_k$ is of the form

$$a_k = \operatorname{diag}(\epsilon_k)\, f_{l_k},$$

where $\epsilon_k \in \{-1, 1\}^n$ is a Rademacher vector and $f_{l_k}$ is the $l_k$th column of the DFT matrix $F \in \mathbb{C}^{n \times n}$.

A first step to address these issues was taken in [56]. The key idea is to impose an incoherence condition of the form

$$\frac{\|x_0\|_\infty}{\|x_0\|_2} \leq \mu < 1, \tag{2.36}$$

which prevents counterexamples of the form (2.35). Under such incoherence condition, one can obtain recovery guarantees for all centered random vectors with i.i.d. real-valued sub-Gaussian entries of unit variance, including the case of Rademacher random vectors that was previously excluded. More precisely, [56, Theorem V.1] yields that with high probability, all signals satisfying (2.36) for $\mu = \frac{1}{\sqrt{8}}$ can be recovered via (2.21) from an order-optimal number of measurements. The proof combines the golfing scheme with stability bounds of [25], confirming that the golfing scheme is well suited to deal with incoherence.

We note that the incoherence condition (2.36) is much weaker than the incoherence conditions in matrix completion and blind deconvolution because it is dimension-free. At the same time, this approach is limited to the real case, as the underlying stability results from [25] exploit that the phase factors to be recovered are actually signs and hence belong to a finite candidate set. Thus for the complex case, one needs different tools, which will be discussed in Sect. 2.4.2 below.

## 2.4 More Refined Descent Cone Analysis

### *2.4.1 Application 5: Blind Deconvolution*

In Sect. 2.2.4, we have discussed why the descent cone analysis framework described in Sect. 2.2.1 cannot be directly applied to the matrix completion and blind deconvolution scenario. In the following, we want to outline how one can refine those methods to obtain novel insights into low-rank matrix recovery problems. For that, we are going to revisit the blind deconvolution setting, see Sect. 2.1.1.2, and demonstrate how to combine a descent cone analysis with incoherence constraints to prove near-optimal bounds in settings which are relevant in practice. This improves over existing error bounds (see, e.g., [1]), which depend polynomially on $K$ and $N$ and hence are quite pessimistic.

More precisely, recall from Sect. 2.2.4 that we only expect to obtain reasonable bounds for matrices with low incoherence, as described by the set

$$\mathcal{H}_\mu := \left\{ h_0 \in \mathbb{C}^K : \ \sqrt{L}|\langle b_\ell, h_0\rangle| \leq \mu \|h_0\|_2 \text{ for all } \ell \in [L] \right\}.$$

Even if the signal is contained in the set, not all principal components of a descent direction need to be incoherent as well. The key observation underlying the following theorem is that these "coherent" descent directions only allow for very small decrements and will hence only play a significant role for very small

noise levels. Thus even under mild lower bounds on the noise level, one obtains near-optimal recovery guarantees.

**Theorem 2.6 ([60, Theorem 3.7])** *Let $\alpha > 0$ and $B \in \mathbb{C}^{L\times K}$ such that $B^*B = Id$. Assume that*

$$L \gtrsim \frac{\mu^2}{\alpha^2} (K + N) \log^2 L.$$

*Then, with high probability, the following statement holds for all $h_0 \in \mathcal{H}_\mu \setminus \{0\}$, all $m_0 \in \mathbb{C}^N \setminus \{0\}$, all $\tau > 0$, and all $e \in \mathbb{C}^L$ with $\|e\|_2 \leq \tau$ :*

*Any $\hat{X}$ minimizing the nuclear norm subject to a data fidelity term of at most $\tau$ satisfies*

$$\|\hat{X} - h_0 m_0^*\|_F \lesssim \frac{\mu^{2/3} \log^{2/3} L}{\alpha^{2/3}} \max \left\{\tau, \alpha \|h_0 m_0^*\|_F\right\}. \tag{2.37}$$

Note that the error estimate in (2.37) depends only logarithmically on $L$. To illustrate this result, assume that the noise level $\tau = \epsilon \mu^{-2} \log^{-2} L$ for some $\epsilon > \epsilon_0$. Then, by setting $\alpha \asymp \epsilon_0 \mu^{-2} \log^{-2} L$, we obtain near-linear error bounds with a required sample complexity at the order of

$$L \geq C_1 \frac{\mu^6}{\epsilon_0^2} (K + N) \log^6 L.$$

This improves over existing noise bounds as in [1] and shows that for large enough noise near-optimal recovery bounds are possible.

***Proof Sketch*** As discussed in Sect. 2.2.4, the minimum conic singular value of the descent cone at the point $h_0 m_0^*$ is ill-conditioned, i.e., there exists a matrix $Z \in \mathbb{C}^{K\times N}$ such that $\frac{\|\mathcal{A}(Z)\|_2}{\|Z\|_F}$ is small. The key observation in the proof is that only matrices $Z$, which are near-orthogonal to the ground truth, can be poorly conditioned. This observation gives rise to the following proof strategy. Namely, we partition the descent cone of the nuclear norm at the point $h_0 m_0^*$ into two cones $\mathcal{K}_1$ and $\mathcal{K}_2$, where the cone $\mathcal{K}_1$ contains all the directions, which are almost orthogonal to the ground truth matrix $h_0 m_0^*$. The cone $\mathcal{K}_2$ contains all the remaining directions. It turns out that matrices in the descent cone $\mathcal{K}_2$ inherit certain coherence properties from the matrices $h_0 m_0^*$, which allows us to apply Mendelson's small ball method to obtain a lower bound for the minimum conic singular value $\lambda_{\min}(\mathcal{A}, \mathcal{K}_2)$, which is at the order of a constant (up to log-factors and ignoring the $\mu$-dependence). Then, using Lemma 2.1, we can control the error, which arises from the directions contained in the cone $\mathcal{K}_2$. In order to control the error, which can arise from directions in $\mathcal{K}_1$, we use the observation that for those directions, the nuclear norm ball around $h_0 m_0^*$ behaves locally like a Euclidean ball. In particular, if the noise level $\tau$ is small, only a short segment in this direction will have smaller nuclear norm than $h_0 m_0^*$. Hence, only a small error can occur from these near-orthogonal directions $Z \in \mathcal{K}_1$. □

### 2.4.2 Application 6: Phase Retrieval with Incoherence

The strategy of splitting the descent cone into two parts can also be applied to the phase retrieval problem with measurements consisting of arbitrary i.i.d. sub-Gaussian entries as introduced in Sect. 2.3.5, allowing to generalize the results of [56] to complex-valued measurements. This is a key step toward understanding real-world applications such as ptychography, which typically do not give rise to real-valued measurements. For complex-valued measurements of real-valued signals, one obtains recovery guarantees exactly analogous to those discussed in Sect. 2.3.5, where this time one requires the incoherence constraint (2.36) with parameter $\mu = \frac{1}{81}$, see [59, Theorem 2].

The proof of these guarantees proceeds via a descent cone analysis of the cone of all admissible directions

$$\mathcal{M}_\mu := \operatorname{cone}\left\{Z \in \mathcal{S}^n : \exists x_0 \in \mathcal{X}_\mu \text{ such that } x_0 x_0^* + Z \in \mathcal{S}_+^n\right\},$$

where

$$\mathcal{X}_\mu := \left\{x_0 \in \mathbb{R}^n \setminus \{0\} : \|x_0\|_\infty \leq \mu \|x_0\|_2\right\}.$$

In order to observe how incoherence is useful, it is instructive to consider the signal $x_0 = e_1 = (1, 0, \ldots, 0) \in \mathbb{R}^n$. Note that the matrix $Z = e_2 e_2^T - e_1 e_1^T$ is an admissible direction, that is, $x_0 x_0^* + tZ \in \mathcal{S}_+^n$ for a sufficiently small $t > 0$. However, if the measurement vector $a_k$ satisfies

$$|(a_k)_1| = |(a_k)_2| = \ldots = |(a_k)_n|.$$

we have $\operatorname{tr}(a_k^* Z a_k) = 0$. The problem here is that all the mass of $Z$ is concentrated on its diagonal. The proof in [59] shows that this cannot be the case, if $x_0$ is incoherent.

To extend the recovery guarantees to complex-valued signals, one needs to address an additional difficulty. Namely, the phases of the entries of the measurement vector must be well distributed on the unit circle in $\mathbb{C}$. To see this, consider real measurements of a complex signal $x$. Then, $x$ and $\bar{x}$ give rise to the same phaseless measurements and hence cannot be distinguished. Such ambiguities can be addressed by an additional constraints on the measurements; then, the proof techniques sketched above carries over. We refer the interested reader to [59] for details.

## 2.5 Conclusion

Although many inverse problems admit a reformulation as a low-rank matrix recovery problem, as we have seen, even for the benchmark reconstruction approach via nuclear norm minimization, the structure imposed by the applications can

make a significant difference. This is true both in terms of how to analyze the reconstruction performance and in terms of the robustness results that can be expected. A key concept in this context is the role of incoherence that distinguishes problems with comparable performance for different signals from problems where for some signals the solution is not even unique. The golfing scheme has proven to be a useful tool to derive signal-dependent recovery guarantees for incoherent signals but has several shortcomings such as limited geometric interpretations. Some of these shortcomings can be addressed by a refined descent cone analysis that partitions the descent cone into multiple parts that can be analyzed separately. To date, however, this approach has only been applied to very few scenarios, in all of which the underlying signal is of rank one. Generalizing this analysis to higher rank and also precisely analyzing the performance in the small noise regime would be of great importance for generating a more comprehensive understanding of the potential and limitation of low-rank matrix recovery via nuclear norm minimization.

**Acknowledgments** This work was prepared as part of the Priority Programme Compressed Sensing in Information Processing (SPP 1798) of the German Research Foundation (DFG). The authors would like to thank Julia Kostina for finding a minor mistake in the first version of the manuscript.

## Appendix: Descent Cone Elements Are Effectively Low Rank

**Lemma 2.2** *Suppose that $Z \in \mathbb{C}^{n_1 \times n_2}$ is contained in the nuclear norm descent cone of a rank-r matrix $X \in \mathbb{C}^{n_1 \times n_2}$. Then,*

$$\|Z\|_* \leq \left(1 + \sqrt{2}\right) \sqrt{r} \|Z\|_F .$$

The constant $1 + \sqrt{2}$ is not optimal and could be further improved by a more refined analysis. The argument presented here is novel and inspired by dual certificate arguments reviewed in Sect. 2.3. It also requires a rectangular generalization of the pinching inequality for Hermitian matrices, see, e.g., [5, Problem II.5.4]

**Theorem 2.13 ((Hermitian) Pinching Inequality)** *Let $P_1, \ldots, P_L \subset \mathbb{H}_n$ be a resolution of the identity ($P_l^2 = P_l$ and $\sum_l P_l = Id$). Then,*

$$\|X\|_* \geq \sum_{l=1}^{L} \|P_l X P_l\|_* \quad \text{for every } X \in \mathbb{H}_n .$$

We can extend pinching to general rectangular matrices by embedding them within a larger block matrix. The *self-adjoint dilation* of $Z \in \mathbb{C}^{n_1 \times n_2}$ is

$$\mathcal{T}(Z) = \begin{pmatrix} 0 & Z \\ Z^* & 0 \end{pmatrix} \in \mathbb{H}_{n_1+n_2} .$$

Dilations preserve spectral information. In particular,

$$\begin{aligned}\|\mathcal{T}(Z)\|_* =& \operatorname{tr}\left(\sqrt{\mathcal{T}(Z)^*\mathcal{T}(Z)}\right) = \operatorname{tr}\begin{pmatrix}\sqrt{ZZ^*} & 0\\ 0 & \sqrt{Z^*Z}\end{pmatrix}\\ =& \operatorname{tr}(\sqrt{ZZ^*}) + \operatorname{tr}(\sqrt{Z^*Z}) = 2\|Z\|_*. \end{aligned} \tag{2.38}$$

For simplicity, we only formulate and prove our generalization of the Hermitian pinching inequality for identity resolutions with two elements each. Statement and proof do, however, readily extend to more general resolutions with compatible dimensions.

**Corollary 2.3 (Pinching for Non-symmetric Matrices)** *Let $P, P^\perp \in \mathbb{H}_{n_1}$ and $Q, Q^\perp \in \mathbb{H}_{n_2}$ be two resolutions of the identity. Then,*

$$\|X\|_* \geq \|PXQ\|_* + \left\|P^\perp X Q^\perp\right\|_* \quad \text{for all } X \in \mathbb{C}^{n_1\times n_2}.$$

***Proof (Corollary 2.3)*** Use Eq. (2.38) to relate the nuclear norm of $X$ to the nuclear norm of its self-adjoint dilation:

$$2\|X\|_* = \|\mathcal{T}(X)\|_* = \left\|\begin{pmatrix}0 & X\\ X^* & 0\end{pmatrix}\right\|_*.$$

Next, we combine $P, P^\perp \in \mathbb{H}_{n_1}$ and $Q, Q^\perp \in \mathbb{H}_{n_2}$ to obtain a resolution of the identity with compatible dimension:

$$\begin{pmatrix}P & 0\\ 0 & Q\end{pmatrix}, \begin{pmatrix}P^\perp & 0\\ 0 & Q^\perp\end{pmatrix} \in \mathbb{H}_{n_1+n_2}.$$

Since everything is Hermitian, we can apply Theorem 2.13 (original pinching) with respect to this resolution of the identity to the nuclear norm of the s.a. dilation:

$$\begin{aligned}\left\|\begin{pmatrix}0 & X\\ X^* & 0\end{pmatrix}\right\|_* &\geq \left\|\begin{pmatrix}P & 0\\ 0 & Q\end{pmatrix}\begin{pmatrix}0 & X\\ X^* & 0\end{pmatrix}\begin{pmatrix}P & 0\\ 0 & Q\end{pmatrix}\right\|_* + \left\|\begin{pmatrix}P^\perp & 0\\ 0 & Q^\perp\end{pmatrix}\begin{pmatrix}0 & X\\ X^* & 0\end{pmatrix}\begin{pmatrix}P^\perp & 0\\ 0 & Q^\perp\end{pmatrix}\right\|_*\\ &= \left\|\begin{pmatrix}0 & PXQ\\ QX^*P & 0\end{pmatrix}\right\|_* + \left\|\begin{pmatrix}0 & P^\perp X Q^\perp\\ Q^\perp X^* P^\perp & 0\end{pmatrix}\right\|_*.\end{aligned}$$

We can now recognize self-adjoint dilations of two rectangular matrices. Using Eq. (2.38) implies

$$\|\mathcal{T}(X)\|_* \geq \|\mathcal{T}(PXQ)\|_* + \|\mathcal{T}(P^\perp X Q^\perp)\|_* = 2\|PXQ\|_* + 2\|P^\perp X Q^\perp\|_*.$$

□

Next, the concept of sign functions of real numbers is extendable to non-Hermitian matrices. Let $X \in \mathbb{C}^{n_1 \times n_2}$ be a rectangular matrix with SVD $X = U\Sigma V^*$. We define its sign matrix to be $\operatorname{sign}(X) = UV^* \in \mathbb{C}^{n_1 \times n_2}$. Note that this sign matrix is unitary and obeys

$$\langle \operatorname{sign}(X), X \rangle_F = \operatorname{tr}\left((UV^*)^* U\Sigma V^*\right) = \operatorname{tr}(\Sigma) = \|X\|_*.$$

The last ingredient is the dual formulation of the nuclear norm:

$$\|X\|_* = \max_{\|U\| \leq 1} |\langle U, X \rangle| = \max_{U \text{unitary}} |\langle U, X \rangle| .$$

***Proof (Lemma 2.2)*** By assumption, $Z \in \mathbb{C}^{n_1 \times n_2}$ is contained in the descent cone of a rank-$r$ matrix $X$. This implies that there exists $\tau > 0$ such that $\|X\|_* \geq \|X + \tau Z\|_*$. Apply an SVD $X = U\Sigma V^*$ and use it to define $r$-dimensional orthoprojectors $P = UU^* \in \mathbb{H}_{n_1}$, $Q = VV^* \in \mathbb{H}_{n_2}$, as well as their orthocomplements $P^\perp = \mathrm{Id} - P$ and $Q^\perp = \mathrm{Id} - Q$. Use them to define the matrix-valued projections

$$\mathcal{P}_{T_X}^\perp : \; Z \mapsto P^\perp Z Q^\perp \quad \text{and} \quad \mathcal{P}_{T_X} : \mapsto Z - \mathcal{P}_{T_X}^\perp(Z) = PZ + ZQ - PZQ$$

such that $Z = \mathcal{P}_{T_X}^\perp(Z) + \mathcal{P}_{T_X}(Z) = Z_{T_X}^\perp + Z_{T_X}$ and, in particular, $X_{T_X}^\perp = 0$ and $X_{T_X} = X$. In words, $\mathcal{P}_{T_X}$ projects $\mathbb{C}^{n_1 \times n_2}$ onto a subspace whose compression to the kernel of $X$ vanishes identically, namely the tangent space of $X$ (as defined in (2.23)). Moreover, for every $Z \in \mathbb{C}^{n_1 \times n_2}$,

$$\begin{aligned} \operatorname{rk}\left(Z_{T_X}\right) &= \operatorname{rk}\left(PZ + (P + P^\perp)ZQ - PZQ\right) = \operatorname{rk}\left(PZ + P^\perp ZQ\right) \\ &\leq \operatorname{rk}(PZ) + \operatorname{rk}\left(P^\perp ZQ\right) \leq \operatorname{rk}(P) + \operatorname{rk}(Q) = 2r, \end{aligned} \tag{2.39}$$

because matrix rank is subadditive and cannot increase under matrix products. Corollary 2.3 (pinching)—with respect to $P$ and $Q$—and the descent cone property of $Z$ together imply

$$\begin{aligned} \|X\|_* \geq \|X + \tau Z\|_* &\geq \|P(X + \tau Z)Q\|_* + \left\| P^\perp (X + \tau Z) Q^\perp \right\|_* \\ &= \|X + \tau PZQ\|_* + \tau \left\| P^\perp Z Q^\perp \right\|_* \\ &= |\langle \operatorname{sign}(X + \tau PZQ), X + \tau PZQ \rangle_F| + \tau \left\| P^\perp Z Q^\perp \right\|_* \\ &\geq |\langle \operatorname{sign}(X), X \rangle_F + \tau \langle \operatorname{sign}(X), PZQ \rangle_F| + \tau \|P^\perp Z Q^\perp\|_* \\ &\geq \|X\|_* + \tau \left( - |\langle \operatorname{sign}(X), PZQ \rangle_F| + \left\| P^\perp Z Q^\perp \right\|_* \right). \end{aligned}$$

Since $\tau > 0$, this chain of inequalities can only be valid if

$$\left\| Z_{T_X}^{\perp} \right\|_* = \left\| P^{\perp} Z Q^{\perp} \right\|_* \leq |\langle \operatorname{sign}(X), PZQ \rangle_F| \leq \|\operatorname{sign}(X)\| \|PZQ\|_* \leq \sqrt{r} \|PZQ\|_F$$

because both $P$ and $Q$ are rank-$r$ projectors. We can combine this with a decomposition $Z = Z_{T_X}^{\perp} + Z_{T_X}$ and Eq. (2.39) to conclude

$$\begin{aligned} \|Z\|_* &\leq \left\| Z_{T_X}^{\perp} \right\|_* + \left\| Z_{T_X} \right\|_* \leq \sqrt{r} \|PZQ\|_F + \sqrt{\operatorname{rank}(Z_{T_X})} \|Z_{T_X}\|_F \\ &\leq \sqrt{r} \|Z\|_F + \sqrt{2r} \|Z\|_F = \left(1 + \sqrt{2}\right) \sqrt{r} \|Z\|_F \end{aligned}$$

because both $Z \mapsto PZQ$ and $Z \mapsto Z_{T_X}$ are contractions with respect to the Frobenius norm. □

## References


1. Ahmed, A., Recht, B., Romberg, J.: Blind deconvolution using convex programming. IEEE Trans. Inform. Theory **60**(3), 1711–1732 (2014)
2. Amelunxen, D., Lotz, M., McCoy, M.B., Tropp, J.A.: Living on the edge: phase transitions in convex programs with random data. Inf. Inference **3**(3), 224–294 (2014)
3. Argyriou, A., Evgeniou, T., Pontil, M.: Convex multi-task feature learning. Mach. Learn. **73**(3), 243–272 (2008)
4. Balan, R., Bodmann, B.G., Casazza, P.G., Edidin, D.: Painless reconstruction from magnitudes of frame coefficients. J. Fourier Anal. Appl. **15**(4), 488–501 (2009)
5. Bhatia, R.: Matrix Analysis. Springer, New York (2013)
6. Burer, S., Monteiro, R.D.C.: A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. Math. Program. **95**(2), 329–357 (2003). https://doi.org/10.1007/s10107-002-0352-8
7. Cai, T.T., Li, X., Ma, Z., et al.: Optimal rates of convergence for noisy sparse phase retrieval via thresholded wirtinger flow. Ann. Stat. **44**(5), 2221–2251 (2016)
8. Candès, E.J., Eldar, Y.C., Strohmer, T., Voroninski, V.: Phase retrieval via matrix completion. SIAM Rev. **57**(2), 225–251 (2015)
9. Candès, E.J., Li, X.: Solving quadratic equations via phaselift when there are about as many equations as unknowns. Found. Comput. Math. **14**(5), 1017–1026 (2014)
10. Candès, E.J., Li, X., Soltanolkotabi, M.: Phase retrieval from coded diffraction patterns. Appl. Comput. Harmon. Anal. **39**(2), 277–299 (2015)
11. Candès, E.J., Li, X., Soltanolkotabi, M.: Phase retrieval via wirtinger flow: theory and algorithms. IEEE Trans. Inform. Theory **61**(4), 1985–2007 (2015)
12. Candès, E.J., Plan, Y.: Matrix completion with noise. Proc. IEEE **98**(6), 925–936 (2010)
13. Candès, E.J., Plan, Y.: A probabilistic and ripless theory of compressed sensing. IEEE Trans. Inform. Theory **57**(11), 7235–7254 (2011)
14. Candès, E.J., Recht, B.: Exact matrix completion via convex optimization. Found. Comput. Math. **9**(6), 717 (2009)
15. Candès, E.J., Romberg, J., Tao, T.: Stable signal recovery from incomplete and inaccurate measurements. Commun. Pure Appl. Math. **59**, 1207–1223 (2005)

16. Candès, E.J., Strohmer, T., Voroninski, V.: Phaselift: exact and stable signal recovery from magnitude measurements via convex programming. Commun. Pure Appl. Math. **66**(8), 1241–1274 (2013)
17. Candès, E.J., Tao, T.: Decoding by linear programming. IEEE Trans. Inform. Theory **51**(12), 4203–4215 (2005). https://doi.org/10.1109/TIT.2005.858979
18. Candès, E.J., Tao, T.: The power of convex relaxation: near-optimal matrix completion. IEEE Trans. Inform. Theory **56**(5), 2053–2080 (2010)
19. Chandrasekaran, V., Recht, B., Parrilo, P.A., Willsky, A.S.: The convex geometry of linear inverse problems. Found. Comput. Math. **12**(6), 805–849 (2012)
20. Chen, Y.: Incoherence-optimal matrix completion. IEEE Trans. Inf. Theory **61**(5), 2909–2923 (2015)
21. Chen, Y., Candès, E.J.: Solving random quadratic systems of equations is nearly as easy as solving linear systems. Commun. Pure Appl. Math. **70**(5), 822–883 (2017)
22. Chen, Y., Chi, Y., Fan, J., Ma, C., Yan, Y.: Noisy matrix completion: understanding statistical guarantees for convex relaxation via nonconvex optimization. arXiv preprint arXiv:1902.07698 (2019)
23. Choudhary, S., Mitra, U.: On identifiability in bilinear inverse problems. In: IEEE International Conference on Accoustics, Speech and Signal Processing, vol. 1(1) (2013)
24. Conca, A., Edidin, D., Hering, M., Vinzant, C.: An algebraic characterization of injectivity in phase retrieval. Appl. Comp. Harmonic Anal. **38**(2), 346–356 (2015)
25. Eldar, Y.C., Mendelson, S.: Phase retrieval: stability and recovery guarantees. Appl. Comput. Harmon. Anal. **36**(3), 473–494 (2014)
26. Fazel, M., Hindi, H., Boyd, S.P., et al.: A rank minimization heuristic with application to minimum order system approximation. In: Proceedings of the American Control Conference, vol. 6, pp. 4734–4739. Citeseer (2001)
27. Fengler, A., Haghighatshoar, S., Jung, P., Caire, G.: Non-Bayesian activity detection, large-scale fading coefficient estimation, and unsourced random access with a massive MIMO receiver. IEEE Trans. Inform. Theory 1–1 (2021). https://doi.org/10.1109/TIT.2021.3065291
28. Fienup, C., Dainty, J.: Phase retrieval and image reconstruction for astronomy. Image Recovery Theory Appl. **231**, 275 (1987)
29. Fienup, J.R.: Phase retrieval algorithms: a comparison. Appl. Opt. **21**(15), 2758–2769 (1982)
30. Fornasier, M., Rauhut, H., Ward, R.: Low-rank matrix recovery via iteratively reweighted least squares minimization. SIAM J. Optim. **21**(4), 1614–1640 (2011)
31. Foucart, S., Rauhut, H.: A Mathematical Introduction to Compressive Sensing, vol. 1. Birkhäuser, Basel (2013)
32. Ge, R., Lee, J.D., Ma, T.: Matrix completion has no spurious local minimum. In: Advances in Neural Information Processing Systems, pp. 2973–2981 (2016)
33. Geppert, J., Krahmer, F., Stöger, D.: Sparse power factorization: balancing peakiness and sample complexity. Adv. Comput. Math. 45, 1711–1728 (2019)
34. Godard, G.H.: Self-recovering equalization and carrier tracking in two dimensional data communication systems. IEEE Trans. Commun. **28**(11), 1867–1875 (1980). https://doi.org/10.1109/TCOM.1980.1094608
35. Gordon, Y.: On Milman's inequality and random subspaces which escape through a mesh in Rn. In: Lindenstrauss, J., Milman, V.D. (eds.) Geometric Aspects of Functional Analysis, pp. 84–106. Springer Berlin Heidelberg, Berlin, Heidelberg (1988)
36. Gross, D.: Recovering low-rank matrices from few coefficients in any basis. IEEE Trans. Inform. Theory **57**(3), 1548–1566 (2011)
37. Gross, D., Krahmer, F., Kueng, R.: A partial derandomization of phaselift using spherical designs. J. Fourier Anal. Appl. **21**(2), 229–266 (2015)
38. Gross, D., Krahmer, F., Kueng, R.: Improved recovery guarantees for phase retrieval from coded diffraction patterns. Appl. Comput. Harmon. Anal. **42**(1), 37–64 (2017)
39. Gross, D., Nesme, V.: Note on sampling without replacing from a finite collection of matrices. arXiv preprint arXiv:1001.2738 (2010)
40. Harrison, R.W.: Phase problem in crystallography. JOSA A **10**(5), 1046–1055 (1993)

41. Haykin: Blind Deconvolution. Prentice Hall, New Jersey (1994). http://www.getcited.org/pub/103095818
42. Horstmeyer, R., Chen, R.Y., Ou, X., Ames, B., Tropp, J.A., Yang, C.: Solving ptychography with a convex relaxation. New J. of Phys. **17**(5), 053044 (2015). https://doi.org/10.1088/1367-2630/17/5/053044
43. Jain, P., Netrapalli, P., Sanghavi, S.: Low-rank matrix completion using alternating minimization. In: Proceedings of the Forty-fifth Annual ACM Symposium on Theory of Computing, STOC '13, pp. 665–674. ACM, New York (2013). https://doi.org/10.1145/2488608.2488693
44. Javanmard, A., Montanari, A.: Localization from incomplete noisy distance measurements. Found. Comput. Math. **13**(3), 297–345 (2013). https://doi.org/10.1007/s10208-012-9129-5
45. Jung, P., Krahmer, F., Stöger, D.: Blind demixing and deconvolution at near-optimal rate. IEEE Trans. Inform. Theory **64**(2), 704–727 (2018)
46. Jung, P., Kueng, R., Mixon, D.G.: Derandomizing compressed sensing with combinatorial design. Front. Appl. Math. Stat. **5**, 26 (2019). https://doi.org/10.3389/fams.2019.00026
47. Kabanava, M., Kueng, R., Rauhut, H., Terstiege, U.: Stable low-rank matrix recovery via null space properties. Inf. Inference **5**(4), 405–441 (2016)
48. Kech, M.: Explicit frames for deterministic phase retrieval via phaselift. CoRR abs/1508.00522 (2015). http://arxiv.org/abs/1508.00522
49. Kech, M., Krahmer, F.: Optimal injectivity conditions for bilinear inverse problems with applications to identifiability of deconvolution problems. SIAM J. Appl. Alg. Geom. **1**(1), 20–37 (2017). https://doi.org/10.1137/16M1067469
50. Keshavan, R.H., Montanari, A., Oh, S.: Matrix completion from a few entries. IEEE Trans. Inf. Theory **56**(6), 2980–2998 (2010)
51. Keshavan, R.H., Montanari, A., Oh, S.: Matrix completion from noisy entries. J. Mach. Learn. Res. **11**, 2057–2078 (2010)
52. Kliesch, M., Szarek, S.J., Jung, P.: Simultaneous structures in convex signal recovery—revisiting the convex combination of norms. Front. Appl. Math. Stat. **5** (2019). https://doi.org/10.3389/fams.2019.00023
53. Klopp, O.: Noisy low-rank matrix completion with general sampling distribution. Bernoulli **20**(1), 282–303 (2014)
54. Koltchinskii, V., Lounici, K., Tsybakov, A.B., et al.: Nuclear-norm penalization and optimal rates for noisy low-rank matrix completion. Ann. Stat. **39**(5), 2302–2329 (2011)
55. Koltchinskii, V., Mendelson, S.: Bounding the smallest singular value of a random matrix without concentration. Int. Math. Res. Not. IMRN **2015**(23), 12991–13008 (2015)
56. Krahmer, F., Liu, Y.K.: Phase retrieval without small-ball probability assumptions. IEEE Trans. Inform. Theory **64**(1), 485–500 (2018)
57. Krahmer, F., Mendelson, S., Rauhut, H.: Suprema of chaos processes and the restricted isometry property. Commun. Pure Appl. Math. **67**(11), 1877–1904 (2014)
58. Krahmer, F., Stöger, D.: Blind deconvolution: Convex geometry and noise robustness. In: 52nd Annual Asilomar Conference on Signals, Systems, and Computers (2018)
59. Krahmer, F., Stöger, D.: Complex phase retrieval from Subgaussian measurements. J. Fourier Anal. Appl. **26**(6), 27 (2020). Id/No 89
60. Krahmer, F., Stöger, D.: On the convex geometry of blind deconvolution and matrix completion. Commun. Pure Appl. Math. (2020)
61. Kueng, R.: Low rank matrix recovery from few orthonormal basis measurements. In: 2015 International Conference on Sampling Theory and Applications (SampTA), pp. 402–406 (2015)
62. Kueng, R., Gross, D., Krahmer, F.: Spherical designs as a tool for derandomization: The case of phaselift. In: 2015 International Conference on Sampling Theory and Applications (SampTA), pp. 192–196 (2015). https://doi.org/10.1109/SAMPTA.2015.7148878
63. Kueng, R., Jung, P.: Robust nonnegative sparse recovery and the nullspace property of 0/1 measurements. IEEE Trans. Inf. Theory **64**(2), 689–703 (2018). https://doi.org/10.1109/TIT.2017.2746620

64. Kueng, R., Rauhut, H., Terstiege, U.: Low rank matrix recovery from rank one measurements. Appl. Comput. Harmon. Anal. **42**(1), 88–116 (2017)
65. Kueng, R., Zhu, H., Gross, D.: Low rank matrix recovery from Clifford orbits. arXiv preprint arXiv:1610.08070 (2016)
66. Kümmerle, C., Sigl, J.: Harmonic mean iteratively reweighted least squares for low-rank matrix recovery. J. Mach. Learn. Res. **19**, 49 (2018)
67. Lee, K., Li, Y., Junge, M., Bresler, Y.: Blind recovery of sparse signals from subsampled convolution. IEEE Trans. Inform. Theory **63**(2), 802–821 (2017)
68. Li, Y., Lee, K., Bresler, Y.: A unified framework for identifiability analysis in bilinear inverse problems with applications to subspace and sparsity models. IEEE Trans. Inf. Theory **63**(2), 822–842 (2017)
69. Ling, S., Strohmer, T.: Blind deconvolution meets blind demixing: algorithms and performance bounds. IEEE Trans. Inform. Theory **63**(7), 4497–4520 (2017)
70. Ma, C., Wang, K., Chi, Y., Chen, Y.: Implicit regularization in nonconvex statistical estimation: Gradient descent converges linearly for phase retrieval, matrix completion and blind deconvolution. arXiv preprint arXiv:1711.10467 (2017)
71. Mendelson, S.: Learning without concentration. In: Conference on Learning Theory, pp. 25–39 (2014)
72. Miao, J., Charalambous, P., Kirz, J., Sayre, D.: Extending the methodology of x-ray crystallography to allow imaging of micrometre-sized non-crystalline specimens. Nature **400**(6742), 342–344 (1999)
73. Millane, R.P.: Phase retrieval in crystallography and optics. JOSA A **7**(3), 394–411 (1990)
74. Natarajan, B.K.: Sparse approximate solutions to linear systems. SIAM J. Comput. **24**(2), 227–234 (1995)
75. Oymak, S., Jalali, A., Fazel, M., Eldar, Y.C., Hassibi, B.: Simultaneously structured models with application to sparse and low-rank matrices. IEEE Trans. Inform. Theory **61**(5), 2886–2908 (2015)
76. Recht, B.: A simpler approach to matrix completion. J. Mach. Learn. Res. **12**(Dec), 3413–3430 (2011)
77. Rennie, J.D.M., Srebro, N.: Fast maximum margin matrix factorization for collaborative prediction. In: Proceedings of the 22Nd International Conference on Machine Learning, ICML '05, pp. 713–719. ACM, New York (2005). https://doi.org/10.1145/1102351.1102441
78. Rodenburg, J.M.: Ptychography and related diffractive imaging methods. Adv. Imaging Electron Phys. **150**, 87–184 (2008)
79. Stockham, T., Cannon, T., Ingebretsen, R.: Blind deconvolution through digital signal processing. Proc. IEEE **63**(4), 678–692 (1975). https://doi.org/10.1109/PROC.1975.9800
80. Stöger, D., Jung, P., Krahmer, F.: Blind deconvolution and compressed sensing. In: 4th International Workshop on Compressed Sensing Theory and Its Applications to Radar, Sonar and Remote Sensing (CoSeRa), pp. 24–27. IEEE (2016)
81. Stöger, D., Jung, P., Krahmer, F.: Blind demixing and deconvolution with noisy data at near optimal rate. In: Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, vol. 10394 (2017)
82. Stöger, D., Jung, P., Krahmer, F.: Blind demixing and deconvolution with noisy data: near-optimal rate. In: WSA 2017; 21th International ITG Workshop on Smart Antennas, pp. 1–5. VDE (2017)
83. Sun, R., Luo, Z.Q.: Guaranteed matrix completion via non-convex factorization. IEEE Trans. Inform. Theory **62**(11), 6535–6579 (2016)
84. Tropp, J.A.: User-friendly tail bounds for sums of random matrices. Found. Comput. Math. **12**(4), 389–434 (2012). https://doi.org/10.1007/s10208-011-9099-z
85. Tropp, J.A.: Convex recovery of a structured signal from independent random linear measurements. In: Sampling Theory, A Renaissance. Compressive Sensing and Other Developments, pp. 67–101. Birkhäuser/Springer, Cham (2015)

86. Tropp, J.A., Yurtsever, A., Udell, M., Cevher, V.: Practical sketching algorithms for low-rank matrix approximation. SIAM J. Matrix Anal. Appl. **38**(4), 1454–1485 (2017). https://doi.org/10.1137/17M1111590
87. Walk, P., Jung, P., Hassibi, B.: MOCZ for blind short-packet communication: basic principles. IEEE Trans. Wirel. Commun. **18**(11), 5080–5097 (2019). https://doi.org/10.1109/TWC.2019.2932668
88. Walk, P., Jung, P., Pfander, G.E., Hassibi, B.: Ambiguities on convolutions with applications to phase retrieval. In: Matthews, M.B. (ed.) 50th Asilomar Conference on Signals, Systems and Computers, ACSSC 2016, Pacific Grove, CA, USA, November 6–9, 2016, pp. 1228–1234. IEEE (2016). https://doi.org/10.1109/ACSSC.2016.7869569
89. Walther, A.: The question of phase retrieval in optics. J. Mod. Opt. **10**(1), 41–49 (1963)
90. Watson, G.: Characterization of the subdifferential of some matrix norms. Linear Algebra Appl. **170**, 33–45 (1992). https://doi.org/10.1016/0024-3795(92)90407-2
91. Yurtsever, A., Udell, M., Tropp, J.A., Cevher, V.: Sketchy decisions: convex low-rank matrix optimization with optimal storage. In: Singh, A., Zhu, X.J. (eds.) Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20–22 April 2017, Fort Lauderdale, FL, USA, Proceedings of Machine Learning Research, vol. 54, pp. 1188–1196. PMLR (2017). http://proceedings.mlr.press/v54/yurtsever17a.html

# Chapter 3
# New Challenges in Covariance Estimation: Multiple Structures and Coarse Quantization

**Johannes Maly, Tianyu Yang, Sjoerd Dirksen, Holger Rauhut, and Giuseppe Caire**

## 3.1 Introduction

The key objective in covariance estimation is simple to state: given $n \in \mathbb{N}$ i.i.d. samples $\mathbf{X}^1, ..., \mathbf{X}^n \overset{\mathrm{d}}{\sim} \mathbf{X}$ of a random vector $\mathbf{X} \in \mathbb{R}^p$, compute a reliable estimate of the covariance matrix $\mathbb{E}[\mathbf{X}\mathbf{X}^\top] = \mathbf{\Sigma} \in \mathbb{R}^{p\times p}$ (without loss of generality, we restrict ourselves here to mean-zero distributions, i.e., $\mathbb{E}[\mathbf{X}] = \mathbf{0}$). For this purpose, a natural estimator is the *sample covariance matrix*

$$\hat{\mathbf{\Sigma}}_n = \frac{1}{n}\sum_{k=1}^{n} \mathbf{X}^k (\mathbf{X}^k)^\top \tag{3.1}$$

as it converges to $\mathbf{\Sigma}$, for $n \to \infty$, by the law of large numbers. Nevertheless, an asymptotic result is of limited use from practical perspective. Given $n \in \mathbb{N}$, it provides no information on the reconstruction error $\|\hat{\mathbf{\Sigma}}_n - \mathbf{\Sigma}\|$ measured in the operator norm $\|\cdot\|$. (Although other norms or error metrics might be considered

J. Maly (✉)
Catholic University of Eichstaett-Ingolstadt, Eichstätt, Germany
e-mail: johannes.maly@ku.de

T. Yang · G. Caire
Technical University of Berlin, Berlin, Germany
e-mail: tianyu.yang@tu-berlin.de; caire@tu-berlin.de

S. Dirksen
Utrecht University, Utrecht, Netherlands
e-mail: s.dirksen@uu.nl

H. Rauhut
RWTH Aachen University, Aachen, Germany
e-mail: rauhut@mathc.rwth-aachen.de



G. Kutyniok et al. (eds.), *Compressed Sensing in Information Processing*, Applied and Numerical Harmonic Analysis, https://doi.org/10.1007/978-3-031-09745-4_3

as well, e.g., the Frobenius norm, we mainly restrict ourselves in this chapter on operator norm bounds as the most common representative.)

In the last two decades, numerous works on non-asymptotic analysis of covariance estimation showed that reliable approximation of $\mathbf{\Sigma}$ by $\hat{\mathbf{\Sigma}}_n$ becomes feasible for sub-Gaussian distributions if $n \gtrsim p$, where $a \lesssim b$ denotes $a \leq Cb$ for some absolute constant $C > 0$. For instance, if $\mathbf{X}$ has a Gaussian distribution, then it is well known [61] that with probability at least $1 - 2e^{-t}$

$$\|\hat{\mathbf{\Sigma}}_n - \mathbf{\Sigma}\| \lesssim \|\mathbf{\Sigma}\| \left( \sqrt{\frac{p+t}{n}} + \frac{p+t}{n} \right). \tag{3.2}$$

This classical result exhibits various weaknesses. For instance, it requires strong concentration of the distribution of $\mathbf{X}$ around its mean. The estimator in (3.1) is sensitive to outliers and not reliable if concentration fails [12, 34]. Furthermore, in applications the ambient dimension can easily exceed the number of accessible samples such that even if concentration may be assumed, the estimate in (3.2) is void.

### *3.1.1 Outline and Notation*

In Sect. 3.2, we briefly discuss massive MIMO as one specific modern application of covariance estimation. The massive MIMO setting originates from wireless communications research and will serve as a motivation for investigating multiple structures and quantized samples in a mathematical framework. Section 3.3 then surveys recent theoretical advances on estimation of structured covariance matrices, and Sect. 3.4 shows the impact of coarse sample quantization on estimation guarantees. Having the theoretical results from Sects. 3.3 and 3.4 in mind, in Sect. 3.5, we finally return to the details of massive MIMO and present our recent approach in engineering literature. We conclude in Sect. 3.6 by discussing the gap between existing theoretical guarantees and practical solutions. Some technical details of Sect. 3.3 are deferred to the Appendix.

We denote $[n] = \{1, ..., n\}$. For any absolute constant $C > 0$, we abbreviate $a \leq Cb$ (resp., $\geq$) as $a \lesssim b$ (resp., $\gtrsim$). We furthermore write $a \lesssim_L b$ (resp., $\gtrsim_L$) if $C$ only depends on the quantity $L$. Whenever we use absolute constants $c, C > 0$, their values may vary from line to line. Scalar-valued functions act component-wise on vectors and matrices. For a set $S$, the *indicator function* $\chi_S$ is 1 on $S$ and 0 on its complement $S^c$. We denote the all ones-matrix by $\mathbf{1} \in \mathbb{R}^{p \times p}$ and the identity by $\mathbf{I} \in \mathbb{R}^{p \times p}$. In particular,

$$[\mathrm{sign}(\mathbf{x})]_i = \begin{cases} 1 & \text{if } x_i \geq 0 \\ -1 & \text{if } x_i < 0, \end{cases}$$

for all $\mathbf{x} \in \mathbb{R}^p$ and $i \in [p]$. For $\mathbf{Z} \in \mathbb{R}^{p\times p}$, we denote the *operator norm* (the maximum singular value) by $\|\mathbf{Z}\| = \sup_{\mathbf{u}\in\mathbb{S}^{p-1}} \|\mathbf{Z}\mathbf{u}\|_2$, the *nuclear norm* (the sum of singular values) by $\|\mathbf{Z}\|_* = \mathrm{tr}(\sqrt{\mathbf{Z}^\top\mathbf{Z}})$, the *Frobenius norm* (trace norm) by $\|\mathbf{Z}\|_F^2 = \mathrm{tr}(\mathbf{Z}^\top\mathbf{Z}) = \sum_{i,j=1}^p Z_{i,j}^2$, the *max norm* by $\|\mathbf{Z}\|_\infty = \max_{i,j} |Z_{i,j}|$, and the *maximum column norm* $\|\mathbf{Z}\|_{1\to 2} = \max_{j\in[p]} \|\mathbf{z}_j\|_2$, where $\mathbf{z}_j$ denotes the $j$-th column of $\mathbf{Z}$. We use $\odot$ for the Hadamard (i.e., entry-wise) product of two matrices. The *uniform distribution* on a set $S$ is denoted by $\mathrm{Unif}(S)$. The *multivariate Gaussian distribution* with mean $\boldsymbol{\mu} \in \mathbb{R}^p$ and covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{p\times p}$ is denoted by $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. The *sub-Gaussian* ($\psi_2$-) *and subexponential* ($\psi_1$-) *norms* of a random variable $X$ are defined by

$$\|X\|_{\psi_\alpha} = \inf\left\{t > 0 \colon \mathbb{E}\left[\exp\left(\frac{|X|^\alpha}{t^\alpha}\right)\right] \le 2\right\}$$

A mean-zero random vector $\mathbf{X}$ on $\mathbb{R}^n$ is called *$K$-sub-Gaussian* if

$$\|\langle \mathbf{X}, \mathbf{x}\rangle\|_{\psi_2} \le K\ \mathbb{E}[\langle \mathbf{X}, \mathbf{x}\rangle^2]^{1/2} \quad \text{for all } \mathbf{x} \in \mathbb{R}^n.$$

## 3.2 Motivation: Massive MIMO

*Multiple-input multiple-output (MIMO)* is a method in wireless communication to enhance the capacity of a radio link by using multiple transmission and multiple receiving antennas. It has become an essential element of wireless communication standards for Wi-Fi and mobile devices [24, 50]. *Massive MIMO* equips the *base station (BS)* with a large number of antennas to further increase bandwidth and potential number of users [44, 45].

In a classical massive MIMO communication system, the BS is equipped with a *uniform linear array (ULA)* of $M$ antennas and communicates with multiple users through a scattering channel, e.g., wave reflection on buildings or objects. See Fig. 3.1 for an exemplary setup. During *uplink (UL)*, the BS receives user pilots and aims at estimating the respective channel covariance matrices, which characterize

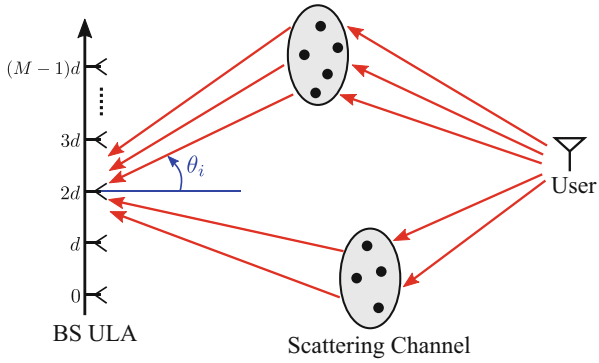



**Fig. 3.1** An exemplary multipath propagation channel, where the user signal is received at the BS through two scattering clusters

each transmission channel. By assuming mutual orthogonality of all UL pilots, it suffices to focus on a single user channel. We denote the corresponding UL channel vector at time–frequency resource $s$ by $\mathbf{h}(s) \in \mathbb{C}^M$ (standard block-fading model, e.g., [59]). Furthermore, we assume that the user transmits a single pilot per channel coherence block such that the channel vectors $\mathbf{h}(s)$ are i.i.d. complex Gaussian vectors, for $s \in [N]$ [27, 28].

The received pilot signal at the BS at resource block $s$ is then given as

$$\mathbf{y}(s) = \mathbf{h}(s)x(s) + \mathbf{z}(s), \tag{3.3}$$

for $s \in [N]$, where $x(s) \in \mathbb{C}$ is the known pilot symbol and $\mathbf{z}(s) \sim \mathcal{CN}(\mathbf{0}, N_0\mathbf{I}) = \mathcal{N}(\mathbf{0}, \frac{N_0}{2}\mathbf{I}) + j\mathcal{N}(\mathbf{0}, \frac{N_0}{2}\mathbf{I})$ models *additive white Gaussian noise (AWGN)*. Without loss of generality, one may assume that the pilot symbols are normalized, i.e., $x(s) = 1$. The core problem of massive MIMO channel estimation is now to estimate the channel covariance matrix

$$\mathbf{\Sigma_h} = \mathbb{E}[\mathbf{h}(s)\mathbf{h}(s)^{\mathsf{H}}] \tag{3.4}$$

from $N$ noisy samples $\mathbf{y}(s)$, $s \in [N]$. Since the number of samples $N$ is limited due to time constraints of the UL phase, one expects for massive MIMO that $N \approx M$. Translating this into our initial theoretical setting, i.e., identifying the ambient dimension $p$ with the number of antennas $M$, the number of samples $n$ with the number of independent time–frequency resources $N$, and the sample vectors $\mathbf{X}^k$ with the channel vectors $\mathbf{h}(s)$, we see that the sample covariance matrix will not provide a reliable estimate of $\mathbf{\Sigma_h}$ in this case, cf. Eq. (3.2) for $n \approx p$. Nevertheless, a closer look into the channel model reveals that $\mathbf{\Sigma_h}$ naturally exhibits intrinsic structures such as low-rankness and Toeplitz structure, cf. Sect. 3.5.

**Structure and Quantization** Let us highlight two crucial points. First, whereas engineers are successful in boosting the sample covariance matrix by using special features of their problem setting, cf. Sect. 3.5, it might simplify existing approaches if alternatives to the sample covariance matrix are used that automatically leverage intrinsic structure(s) of the covariance matrix. As Sect. 3.3 will show, the last decade substantially improved our theoretical understanding in this regard. Second, if the above methods are used in real applications, one has to take into account that the sample vectors $\mathbf{y}(s)$ have to be quantized to finite alphabets before digital processing. Especially, in massive MIMO, the information loss due to quantization can be significant since fine quantization at a multitude of antennas leads to enormous energy consumption. The results presented in Sect. 3.4 can be seen as a first theoretical step into understanding the non-asymptotic behavior of covariance estimators under coarse quantization of the samples. Since we concentrate on memoryless quantization schemes (each vector entry is quantized independently of all others), our model should be applicable to massive MIMO in a straightforward way.

## 3.3 Estimation of Structured Covariance Matrices and Robustness Against Outliers

As we have already seen in Sect. 3.2, there are several structures of interest that $\mathbf{\Sigma}$ might exhibit in applications. We concentrate here on three important instances—sparsity, low-rankness, and Toeplitz structure—that naturally emerge in engineering, biology, and data science, e.g., [42, 53]. Parts of the results we review below are not restricted to Gaussian random vectors but allow to treat heavy-tailed distributions that only satisfy assumptions on their lower moments. Techniques for robust covariance estimation include median of means [31, 49], element- and spectrum-wise truncation [12, 47], and $M$-estimators [47, 48]. The recent work [46] even constructs a "sub-Gaussian" estimator that only requires a finite kurtosis assumption ($L_4$–$L_2$-norm equivalence). In this context, an estimator is called sub-Gaussian if it performs on non-Gaussian distributions as well as the sample covariance matrix applied to Gaussian distributions, for further discussion see [46]. Although the proposed construction is computationally intractable, it illustrates the potential of robust estimation. For further information on early and recent approaches to robust covariance estimation, we refer the reader to [29, 34].

### *3.3.1 Sparse Covariance Matrices*

We begin with the assumption that $\mathbf{\Sigma}$ is a *sparse* matrix, i.e., only few entries of $\mathbf{\Sigma}$ are relevant and hence non-zero. If $\mathbf{X}$ models ordered variables, the non-zero entries of $\mathbf{\Sigma}$, for instance, might cluster around the diagonal such that $\mathbf{\Sigma}$ is a banded or tapered matrix. A straightforward way to estimate such covariance matrices is to band/taper the sample covariance matrix $\hat{\mathbf{\Sigma}}_n$ [6, 11, 23]. If the variables are not ordered and the non-zero entries of $\mathbf{\Sigma}$ do not cluster, thresholding of $\hat{\mathbf{\Sigma}}_n$ is a viable alternative [5, 19]. As remarked in [40], the aforementioned approaches can be treated in a unified way by introducing a *mask* $\mathbf{M} \in [0,1]^{p\times p}$ and considering the *masked sample covariance matrix* $\mathbf{M} \odot \hat{\mathbf{\Sigma}}_n$. The masked formulation allows to decompose the estimation error

$$\|\mathbf{M} \odot \hat{\mathbf{\Sigma}}_n - \mathbf{\Sigma}\| \leq \|\mathbf{M} \odot \hat{\mathbf{\Sigma}}_n - \mathbf{M} \odot \mathbf{\Sigma}\| + \|\mathbf{M} \odot \mathbf{\Sigma} - \mathbf{\Sigma}\|$$

into a variance term that behaves well if $\mathbf{M}$ is (close to) sparse and a bias term that is small whenever $\mathbf{M}$ encodes the support of $\mathbf{\Sigma}$. The bias term is deterministic and solely depends on a proper choice of $\mathbf{M}$. For understanding the influence of sparsity on the required sample size, it thus suffices to control the variance term. The corresponding state-of-the-art result can be found in [13] which extends [40] from Gaussian distributions to general distributions of finite fourth moment and strengthens [40] if applied to Gaussian distributions. To facilitate the comparison with (3.2), we present the result only in the Gaussian case.

**Theorem 3.1 ([13, Theorem 1.1])** *Let* $\mathbf{M} \in [0,1]^{p\times p}$*, for* $p \geq 3$*, be fixed and* $\mathbf{X} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$ *with* $\mathbf{\Sigma} \in \mathbb{R}^{p\times p}$*. Then,*

$$\left(\mathbb{E}\|\mathbf{M}\odot\hat{\mathbf{\Sigma}}_n - \mathbf{M}\odot\mathbf{\Sigma}\|^2\right)^{\frac{1}{2}}$$
$$\lesssim \|\mathbf{\Sigma}\|\left(\sqrt{\frac{\|\mathbf{\Sigma}\|_\infty}{\|\mathbf{\Sigma}\|}\cdot\frac{\|\boldsymbol{M}\|_{1\to 2}^2\log(p)}{n}} + \frac{\|\mathbf{\Sigma}\|_\infty}{\|\mathbf{\Sigma}\|}\cdot\frac{\|\boldsymbol{M}\|\log(p)\log(np)}{n}\right).$$

Theorem 3.1 only bounds the second moment of the variance term, which yields high-probability estimates via Markov's inequality. However, the same proof techniques apply to higher moments of the variance term as well such that exponential tail bounds can be achieved for Gaussian $\mathbf{X}$, cf. [13, Section 3.3].

Let us compare Theorem 3.1 with (3.2). For general covariance estimation, i.e., $\mathbf{M} = \mathbf{1}$, we have $\|\boldsymbol{M}\|_{1\to 2}^2 = \|\boldsymbol{M}\| = p$, which implies that up to log-factors both results are of the same order $O(\sqrt{\frac{p}{n}} + \frac{p}{n})$. If $\mathbf{M}$ encodes sparsity, however, meaning that only up to $s \ll p$ columns and rows are non-zero and $\|\boldsymbol{M}\|_{1\to 2}^2 = \|\boldsymbol{M}\| = s$, the estimation error is considerably reduced when applying Theorem 3.1. A similar error reduction occurs if $\mathbf{M}\odot\hat{\mathbf{\Sigma}}_n$ is a banded estimator of bandwidth $B$.

**Estimation via Thresholding** While the masked framework provides a unified understanding of the intrinsic complexity of sparse covariance estimation, in practice the mask $\mathbf{M}$ is unknown. A more realistic approach to the problem is hence thresholding procedures as, e.g., [5]. To allow for non-ordered covariance matrices, i.e., general sparsity and not only limited bandwidth of the matrix, the authors of [5] introduce the set of bounded and (effectively) sparse covariance matrices

$$\mathcal{U}(q,s,M) := \left\{\mathbf{\Sigma}\colon \Sigma_{i,i} \leq M \text{ and } \sum_{j=1}^{p}|\Sigma_{i,j}|^q \leq s, \text{ for all } i\in[p]\right\},$$

for $q \in [0,1)$ and $s, M > 0$. If $q = 0$, the matrices in $\mathcal{U}(q,s,M)$ have at most $s$ non-zero entries per row; if $q > 0$, the rows are close to $s$-sparse vectors. To estimate $\mathbf{\Sigma} \in \mathcal{U}(q,s,M)$, the thresholded estimator $\mathbb{T}_\tau(\hat{\mathbf{\Sigma}}_n)$ is considered, where

$$[\mathbb{T}_\tau(\mathbf{A})]_{i,j} = \begin{cases} A_{i,j} & \text{if } |A_{i,j}| \geq \tau, \\ 0 & \text{else,}\end{cases} \tag{3.5}$$

for any $\tau > 0$ and $\mathbf{A} \in \mathbb{R}^{p\times p}$.

**Theorem 3.2 ([5, Theorem 1])** *Let* $\mathbf{X} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$*, for* $\mathbf{\Sigma} \in \mathcal{U}(q,s,M)$*, and* $M' > 0$ *be sufficiently large (depending on* $M$*). If*

$$\tau = M' \sqrt{\frac{\log(p)}{n}},$$

*for* $n \gtrsim \log(p)$*, then with probability at least* $1 - e^{-cn\tau^2}$

$$\|\mathbb{T}_\tau(\hat{\mathbf{\Sigma}}_n) - \mathbf{\Sigma}\| = O\left(s\left(\frac{\log(p)}{n}\right)^{\frac{1-q}{2}}\right).$$

Theorem 3.2 does not require knowledge on the support of $\mathbf{\Sigma}$ and respects sparsity defects. However, if we once more consider the case $q = 0$, we see that the estimate in Theorem 3.2 is suboptimal since the error behaves (up to log-factors) like $O\left(\sqrt{\frac{s^2}{n}}\right)$ and not like $O(\sqrt{\frac{s}{n}})$ as one would expect.

### 3.3.2 Low-Rank Covariance Matrices

When working with high-dimensional random vectors, another commonly considered structural prior is to assume that the distribution concentrates around a low-dimensional manifold. This may manifest itself in $\mathbf{\Sigma}$ being a low-rank matrix. Interestingly, the sample covariance matrix in (3.1) intrinsically leverages low-rankness of $\mathbf{\Sigma}$. To understand this phenomenon, we consider the *effective rank* of $\mathbf{\Sigma}$ defined as

$$\mathbf{r}(\mathbf{\Sigma}) = \frac{\|\mathbf{\Sigma}\|_*}{\|\mathbf{\Sigma}\|}.$$

It is straightforward to verify that $1 \leq \mathbf{r}(\mathbf{\Sigma}) \leq \operatorname{rank}(\mathbf{\Sigma})$. In contrast to the rank of $\mathbf{\Sigma}$, the quantity $\mathbf{r}(\mathbf{\Sigma})$ is small even if $\mathbf{\Sigma}$ is only close to a low-rank matrix, e.g., consider $\mathbf{\Sigma}$ to be a full rank matrix with exponentially decaying spectrum.

**Theorem 3.3 ([37, Corollary 2])** *Let* $\mathbf{X} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$*, for* $\mathbf{\Sigma} \in \mathbb{R}^{p\times p}$*, and* $n \gtrsim \mathbf{r}(\mathbf{\Sigma})$*. Then with probability at least* $1 - e^{-t}$ *the sample covariance matrix satisfies*

$$\|\hat{\mathbf{\Sigma}}_n - \mathbf{\Sigma}\| \lesssim \|\mathbf{\Sigma}\| \left(\sqrt{\frac{\mathbf{r}(\mathbf{\Sigma})}{n}} + \frac{\mathbf{r}(\mathbf{\Sigma})}{n} + \sqrt{\frac{t}{n}} + \frac{t}{n}\right).$$

The authors of [37] further show that the bound in Theorem 3.3 is tight up to constants. If we compare the result to (3.2), we see that both estimates agree for (effectively) full rank matrices like $\mathbf{\Sigma} = \mathbf{I}$. If $\mathbf{\Sigma}$ is of low rank, however, Theorem 3.3 controls the estimation error even in the case $n < p$.

**Low-Rank Estimators** We could stop at this point since $\hat{\mathbf{\Sigma}}_n$ apparently meets our requirements. Nevertheless, two questions remain. First, if one assumes $\mathbf{\Sigma}$ to be

low rank, one would wish for an estimator that is low rank itself, and, second, Theorem 3.3 fails if $\mathbf{X}$ does not exhibit strong concentration around its mean. The first point can be addressed by using the LASSO estimator

$$\hat{\mathbf{\Sigma}}_n^{\lambda} = \arg\min_{\mathbf{S} \succcurlyeq \mathbf{0}} \|\mathbf{S} - \hat{\mathbf{\Sigma}}_n\|_F^2 + \lambda \|\mathbf{S}\|_* \,, \tag{3.6}$$

where $\lambda > 0$ is a tunable parameter. Initially introduced in [43] to estimate covariance matrices from incomplete observations, the result reads in our setting as follows.

**Theorem 3.4 ([43, Corollary 1])** *Let* $\mathbf{X} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$*, for* $\mathbf{\Sigma} \in \mathbb{R}^{p \times p}$*, and* $n \gtrsim \mathbf{r}(\mathbf{\Sigma}) \log(2p+n)^2$*. If*

$$\lambda = C \sqrt{\mathrm{tr}(\hat{\mathbf{\Sigma}}_n) \|\hat{\mathbf{\Sigma}}_n\|} \sqrt{\frac{\log(2p)}{n}},$$

*for a sufficiently large absolute constant* $C > 0$*, then with probability at least* $1 - \frac{1}{2p}$ *the estimator in* (3.6) *satisfies*

$$\|\hat{\mathbf{\Sigma}}_n^{\lambda} - \mathbf{\Sigma}\| \lesssim \|\mathbf{\Sigma}\| \sqrt{\frac{\mathbf{r}(\mathbf{\Sigma}) \log(2p)}{n}}.$$

The nuclear norm regularization in (3.6) induces (effective) low-rankness on $\hat{\mathbf{\Sigma}}_n^{\lambda}$ [21, 51] and the order of estimation error reflects up to log-factors the one in Theorem 3.3. Furthermore, the construction of $\hat{\mathbf{\Sigma}}_n^{\lambda}$ can easily be adapted to heavy-tailed distributions by replacing $\hat{\mathbf{\Sigma}}_n$ with an appropriate robust counterpart, e.g., the spectrum-wise truncated sample covariance matrix [34]. A corresponding version of Theorem 3.4 that is not restricted to (sub)-Gaussian distributions is [34, Theorem 5.2].

### 3.3.3 Toeplitz Covariance Matrices and Combined Structures

The third structure we discuss here in detail naturally arises in various engineering problems. If the entries of $\mathbf{X}$ resemble measurements on a temporal or spatial grid whose covariances only depend on the distances of measurements (in time or space) but not their location, $\mathbf{\Sigma}$ is a *symmetric Toeplitz matrix*, i.e.,

$$\mathbf{\Sigma} = \begin{pmatrix} \sigma_1 & \sigma_2 & \cdots & \sigma_p \\ \sigma_2 & \ddots & \ddots & \vdots \\ \vdots & \ddots & & \sigma_2 \\ \sigma_p & \cdots & \sigma_2 & \sigma_1 \end{pmatrix},$$

and the first column $\boldsymbol{\sigma} \in \mathbb{R}^p$ determines $\mathbf{\Sigma}$ via $\Sigma_{i,j} = \sigma_{|i-j|+1}$. (For simplicity, we identify Toeplitz matrices with their first column in the following.) Such a structure appears, for instance, in Direction-Of-Arrival (DOA) estimation [38] and medical/radar imaging processing [9, 56]. For further examples, we refer the reader to [53]. Since Toeplitz structure reduces the degrees of freedom in $\mathbf{\Sigma}$ from $p^2$ to $p$, leveraging this structure can lead to a notable reduction in sample complexity.

The authors of [10] propose to average, the sample covariance matrix along its diagonals to obtain the Toeplitz estimator $\hat{\mathbf{\Sigma}}_n^{\text{Toep}}$ defined as

$$[\hat{\sigma}_n^{\text{Toep}}]_r = \frac{1}{(p+1)-r} \sum_{i-j=r-1} [\hat{\Sigma}_n]_{i,j}, \quad \text{for } r \in [p]. \tag{3.7}$$

They derive error estimates for Gaussian distributions with banded Toeplitz covariance matrices.

The more recent work [33] extends these results to non-Gaussian distributions and general masks as introduced in Sect. 3.3.1. To be more precise, the authors of [33] assume that the distribution of $\mathbf{X}$ has the so-called convex concentration property.

**Definition 3.1** A random vector $\mathbf{X} \in \mathbb{R}^p$ has the *convex concentration property* with constant $K$ if for any 1-Lipschitz function $\phi\colon \mathbb{R}^p \to \mathbb{R}$, one has $\mathbb{E}[\phi(\mathbf{X})] < \infty$ and

$$\Pr\left[|\phi(\mathbf{X}) - \mathbb{E}[\phi(\mathbf{X})]| \geq t\right] \leq 2e^{-\frac{t^2}{K^2}}, \quad \text{for all } t > 0.$$

By setting $\phi(\cdot) = \langle \cdot, \mathbf{x} \rangle$, for $\mathbf{x} \in \mathbb{R}^p$, one easily sees that all distributions that have the convex concentration property are sub-Gaussian. For the sake of consistency, we therefore restrict ourselves here to Gaussian distributions as their most prominent representative. For a symmetric Toeplitz mask $\mathbf{M} \in [0,1]^{p \times p}$ characterized by its first column $\mathbf{m} \in [0,1]^p$, we furthermore define the weighted $\ell_1$- and $\ell_2$-norms of $\mathbf{m}$ as

$$\|\mathbf{m}\|_{1,*} = \sum_{r=1}^{p} \frac{m_r}{(p+1)-r} \quad \text{and} \quad \|\mathbf{m}\|_{2,*} = \left( \sum_{r=1}^{p} \frac{m_r^2}{(p+1)-r} \right)^{\frac{1}{2}}.$$

**Theorem 3.5 ([33, Theorem 3])** *Let* $\mathbf{M} \in [0,1]^{p\times p}$ *be a symmetric Toeplitz mask and* $\mathbf{X} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$ *with* $\mathbf{\Sigma} \in \mathbb{R}^{p\times p}$ *symmetric and Toeplitz. Then,*

$$\mathbb{E}\|\mathbf{M} \odot \hat{\mathbf{\Sigma}}_n^{\mathrm{Toep}} - \mathbf{M} \odot \mathbf{\Sigma}\| \lesssim \|\mathbf{\Sigma}\| \left( \sqrt{\frac{\|\mathbf{m}\|_{2,*} \log(p)}{n}} + \frac{\|\mathbf{m}\|_{1,*} \log(p)}{n} \right).$$

As Theorem 3.1, the result is not restricted to an estimate of the expected error but includes respective high probability bounds with exponential tail decay. Let us compare Theorem 3.5 to Theorem 3.1. If we ignore log-factors and assume that $\mathbf{M}$ is a banding or tapering mask with support bandwidth $B \leq \frac{p}{2}$, i.e., only the $B$ innermost diagonals of $\mathbf{M}$ are non-zero, Theorem 3.5 guarantees an estimation error of order $O(\sqrt{\frac{B}{pn}} + \frac{B}{pn})$, cf. [33, Corollary 2], which improves the estimate $O(\sqrt{\frac{B}{n}} + \frac{B}{n})$ of Theorem 3.1 by a factor $p$. This improvement corresponds to the reduction in degrees of freedom when comparing Toeplitz to general matrices. Note, however, that the additional assumption $B \leq \alpha p$, for $\alpha \in (0,1)$, is required for such a reduction since estimation of the outermost diagonals of $\mathbf{\Sigma}$ is hardly enhanced by averaging over the Toeplitz structure. This is expressed by Theorem 3.5 since $\|\mathbf{m}\|_{1,*}$ and $\|\mathbf{m}\|_{2,*}$ are $O(1)$ and not $O(\frac{1}{p})$ if the tail entries of $\mathbf{m}$ are not of vanishing magnitude.

**Estimation via Thresholding** Theorem 3.5 differs from the previously discussed results in the sense that it allows to simultaneously leverage two structures of $\mathbf{\Sigma}$, sparsity and Toeplitz structure. Nevertheless, as in Sect. 3.3.1, the masked framework leaves open the question of how to choose $\mathbf{M}$ in practice. By combining the thresholded approach in Theorem 3.2 with the techniques of Theorem 3.5, one can obtain a thresholded Toeplitz estimator which profits from both structural priors. To state a corresponding estimate, let us define the set of bounded Toeplitz covariance matrices with (effectively) sparse first column $\boldsymbol{\sigma}$ by

$$\mathcal{U}^{\mathrm{Toep}}(q,s,M) := \left\{ \mathbf{\Sigma} \colon \Sigma_{i,j} = \sigma_{|i-j|+1} \leq M, \text{ for } \boldsymbol{\sigma} \in \mathbb{R}^p \text{ with } \sum_{r=1}^{p} |\sigma_r|^q \leq s \right\}.$$

We furthermore denote by $\mathbb{B}_{\alpha p}(\mathbf{\Sigma})$ the matrix $\mathbf{\Sigma}$ restricted to bandwidth $\alpha p$, i.e., $[\mathbb{B}_{\alpha p}(\mathbf{\Sigma})]_{i,j} = \Sigma_{i,j}$ if $|i-j|+1 \leq \alpha p$ and $[\mathbb{B}_{\alpha p}(\mathbf{\Sigma})]_{i,j} = 0$ else.

**Theorem 3.6** *There exists an absolute constant* $C > 0$ *such that the following holds. Let* $\mathbf{X}$ *have the convex concentration property with constant* $K$*. Let* $\mathbb{E}[\mathbf{X}] = \mathbf{0}$ *and* $\mathbb{E}[\mathbf{X}\mathbf{X}^\top] = \mathbf{\Sigma}$*, for* $\mathbf{\Sigma} \in \mathcal{U}^{\mathrm{Toep}}(q,s,M)$*. For all* $\alpha \in (0,1)$ *and* $c > 1$*, we have with probability at least* $1 - (2\alpha p)^{-(c-1)}$ *that if*

$$\tau = \sqrt{\frac{2c}{(1-\alpha)}} \max\{CK^2, \sqrt{C}K\} \sqrt{\frac{\log(p)}{np}}, \tag{3.8}$$

*then*

$$\left\| \mathbb{T}_\tau(\mathbb{B}_{\alpha p}(\hat{\mathbf{\Sigma}}_n^{\text{Toep}})) - \mathbf{\Sigma} \right\| \lesssim s \left( \max\{C^2K^4, CK^2\} \frac{c}{1-\alpha} \frac{\log(p)}{np} \right)^{\frac{1-q}{2}} + \|\mathbb{B}_{\alpha p}(\mathbf{\Sigma}) - \mathbf{\Sigma}\|,$$

*where* $\mathbb{T}_\tau$ *is the thresholding operator from* (3.5).

Two comments are in order here. To gain from the Toeplitz structure, Theorem 3.6 requires $\mathbf{\Sigma}$ to be close to a banded matrix. This is as in Theorem 3.5 before and has been discussed previously. Moreover, by adapting the proof strategy of Theorem 3.2, the result inherits the slightly suboptimal error decay in the sparsity level $s$, cf. the discussion of Theorem 3.2 for the case $q = 0$. The proof, which combines ideas from [5] and [33], can be found in the Appendix.

**Combining Toeplitz Structure and Low-Rankness** Sparsity is not the only structure that can be imposed on Toeplitz matrices. For instance, in massive MIMO, see Sect. 3.2, low-rankness of $\mathbf{\Sigma}$ may naturally be assumed in addition to Toeplitz structure [28]. The recent works [20, 39] propose several algorithms to estimate low-rank Toeplitz covariance matrices from partial observations by a technique called "sparse ruler." In particular, the authors can show that the sufficient number of samples to approximate $\mathbf{\Sigma}$ scales (up to log-factors) polynomial in the (effective) rank of $\mathbf{\Sigma}$.

*Remark 3.1* Before closing this section, let us briefly comment on the three types of structures discussed above and their mutual relation:

*Sparsity:* The concept of sparsity is the maybe most fundamental way of theoretically describing intrinsic "low-complexity" of points in a vector space. Whereas we only introduced sparsity of vectors in $\mathbb{R}^n$ with respect to the canonical basis, it is straightforward to generalize the definition to arbitrary vector spaces and other bases (or even frames). Note, however, that sparsity strongly depends on the chosen representation of objects in space, i.e., a point that is sparse in one basis need not be sparse in another.

*Low-rankness:* One can view low-rankness as a special case of sparsity since a matrix is low rank if and only if the vector of its singular values is sparse in the canonical basis. Stated differently, a matrix is low rank if its induced linear mapping only acts on low-dimensional subspaces of the ambient input and output space. This second characterization shows that, in contrast to sparsity, low-rankness is not representation dependent. Furthermore, one can generalize the concept to higher dimensional linear operators as well, e.g., tensor spaces.

*Toeplitz structure:* Just as low-rankness, Toeplitz structure is a special type of sparsity that requires matrix structure of the points in space. Its low-dimensional structure lies in the fact that only $(2n-1)$ parameters are necessary to characterize an $\mathbb{R}^{n\times n}$ Toeplitz matrix. In contrast to low-rankness, Toeplitz structure is representation dependent. Nevertheless, Toeplitz matrices naturally appear as covariance matrices of stationary random processes, i.e., if the covariance of two

events does not depend on their localization in time but only their distance in time.

For further discussion and literature on the subject, we refer the interested reader to [22].

## 3.4 Estimation from Quantized Samples

All results stated above assume that the sample vectors $\mathbf{X}^k$ are real-valued, i.e., one has access to infinite precision representations of the samples. In applications, this assumption is not always fulfilled. Especially in signal processing, samples are collected via sensors and, hence, need to be quantized to finitely many bits before they can be digitally transmitted and processed. Engineers have been examining the influence of coarse quantization on correlation and covariance estimation for decades, e.g., [2, 14, 30, 41, 54]. However, in contrast to classical covariance estimation from unquantized samples, so far only asymptotic estimation guarantees have been derived in the quantized setting. To improve our understanding on the effect of quantization on covariance estimation, we analyzed two memoryless one-bit quantization schemes in our recent work [16]. We call a quantizer memoryless if it quantizes each entry of $\mathbf{X}^k$ independently of all remaining entries. This is fundamentally different from feedback systems, e.g., $\Sigma\Delta$-quantization [4, 55], and of particular interest for large-scale applications like massive MIMO where the entries of $\mathbf{X}^k$ correspond to inputs from different antennas, cf. Sect. 3.2. We conclude by providing a detailed discussion of the models and results in [16].

### *3.4.1 Sign Quantization*

In the first setting, we assume to receive one-bit quantized samples

$$\operatorname{sign}(\mathbf{X}^k) \in \{-1, 1\}^p, \tag{3.9}$$

for $k \in [n]$, instead of $\mathbf{X}^k$ itself. (Recall that we apply scalar functions like sign entry-wise to vectors and matrices.) Since the quantizer sign is scale-invariant, i.e., $\operatorname{sign}(\mathbf{z}) = \operatorname{sign}(\mathbf{Dz})$ for any diagonal matrix $\mathbf{D} \in \mathbb{R}^{p\times p}$ with strictly positive entries and $\mathbf{z} \in \mathbb{R}^p$, we can only hope to recover the correlation matrix of the distribution, i.e., a normalized version of $\mathbf{\Sigma}$ with entries $\left[\frac{\Sigma_{i,j}}{\sqrt{\Sigma_{i,i}}\sqrt{\Sigma_{j,j}}}\right]_{i,j}$. We thus assume that $\mathbf{X} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$, where $\mathbf{\Sigma}$ has ones on its diagonal.

It has been known for decades that

$$\tilde{\mathbf{\Sigma}}_n = \sin\left(\frac{\pi}{2n}\sum_{k=1}^{n}\operatorname{sign}(\mathbf{X}^k)\operatorname{sign}(\mathbf{X}^k)^\top\right) \tag{3.10}$$

is well suited to approximate $\mathbf{\Sigma}$ from the quantized samples, cf. [30]. Note that the specific form of $\tilde{\mathbf{\Sigma}}_n$ is motivated by Grothendieck's identity (see, e.g., [61, Lemma 3.6.6]), also known as "arcsin-law" in the engineering literature [30, 60], which implies that

$$\mathbf{\Gamma} := \mathbb{E}[\operatorname{sign}(\mathbf{X}^k)\operatorname{sign}(\mathbf{X}^k)^\top] = \frac{2}{\pi}\arcsin(\mathbf{\Sigma}) \tag{3.11}$$

if $\mathbf{X} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$. Applying the strong law of large numbers and the continuity of the sine function to (3.10), one easily obtains with (3.11) that $\tilde{\mathbf{\Sigma}}_n$ is a consistent estimator of $\mathbf{\Sigma}$.

The two key quantities for understanding the non-asymptotic performance of $\tilde{\mathbf{\Sigma}}_n$ are $\mathbf{\Gamma}$ and

$$\mathbf{A} := \cos(\arcsin(\mathbf{\Sigma})) = \cos(\tfrac{\pi}{2}\mathbf{\Gamma}).$$

Furthermore, we define

$$\sigma(\mathbf{Z})^2 := \mathbf{Z}^2 \odot \mathbf{\Gamma} - (\mathbf{Z} \odot \mathbf{\Gamma})^2 = \frac{2}{\pi}\mathbf{Z}^2 \odot \arcsin(\mathbf{\Sigma}) - \frac{4}{\pi^2}\big(\mathbf{Z} \odot \arcsin(\mathbf{\Sigma})\big)^2,$$

for symmetric $\mathbf{Z} \in \mathbb{R}^{p\times p}$.

**Theorem 3.7 ([16, Theorem 1])** *There exist constants $c_1, c_2 > 0$ such that the following holds. Let $\mathbf{X} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$ with $\Sigma_{i,i} = 1$, for $i \in [p]$, and $\mathbf{X}^1, \ldots, \mathbf{X}^n \overset{\mathrm{d}}{\sim} \mathbf{X}$ be i.i.d. samples of $\mathbf{X}$. Let $\mathbf{M} \in [0,1]^{p\times p}$ be a fixed symmetric mask. Then, for all $t \geq 0$ with $n \geq c_1 \log^2(p)(\log(p) + t)$, the biased sign estimator $\tilde{\mathbf{\Sigma}}_n$ fulfills with probability at least $1 - 2e^{-c_2 t}$*

$$\begin{aligned}\|\mathbf{M} \odot \tilde{\mathbf{\Sigma}}_n - \mathbf{M} \odot \mathbf{\Sigma}\| \lesssim \|\sigma(\mathbf{M} \odot \mathbf{A})\| \sqrt{\frac{\log(p)+t}{n}} \\ + \left(\max\{\|\mathbf{M} \odot \mathbf{A}\|, \|\mathbf{M} \odot \mathbf{\Sigma}\|\}\right) \frac{\log(p)+t}{n}.\end{aligned} \tag{3.12}$$

The right-hand side in Theorem 3.7 (for convenience, we only consider the case $\mathbf{M} = \mathbf{1}$ here) can be trivially estimated to get

$$\|\tilde{\mathbf{\Sigma}}_n - \mathbf{\Sigma}\| \lesssim \max\{\|\cos(\arcsin(\mathbf{\Sigma}))\|, \|\mathbf{\Sigma}\|\}\left(\sqrt{\frac{\log(p)+t}{n}} + \frac{\log(p)+t}{n}\right),$$

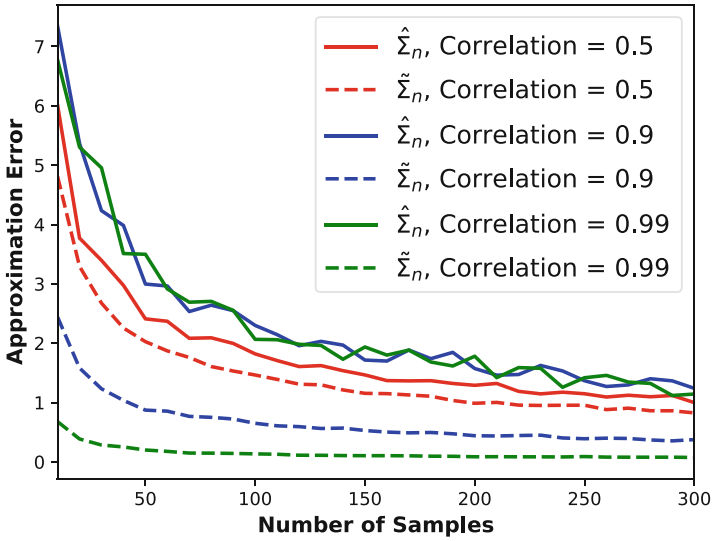



**Fig. 3.2** The experiment from [16] depicts average estimation error of $\hat{\mathbf{\Sigma}}_n$ and $\tilde{\mathbf{\Sigma}}_n$ in operator norm, for $p = 20$, $n$ varying from 10 to 300 and three different choices of the ground truth $\mathbf{\Sigma}$ with ones on the diagonal and off-diagonal entries equal to $c = 0.5$, $c = 0.9$, and $c = 0.99$

which is up to the additional dependence on $\cos(\arcsin(\mathbf{\Sigma}))$ comparable to the error bound in (3.2) for $\hat{\mathbf{\Sigma}}_n$. This is remarkable since $\tilde{\mathbf{\Sigma}}_n$ accesses considerably less information on the samples than $\hat{\mathbf{\Sigma}}_n$.

Theorem 3.7 even suggests that for strongly correlated distributions of $\mathbf{X}$, i.e., $\mathbf{\Sigma} \approx \mathbf{1}$, the dominant first term on the right-hand side of (3.12) vanishes. In other words, the bound in (3.12) predicts $\tilde{\mathbf{\Sigma}}_n$ to outperform $\hat{\mathbf{\Sigma}}_n$ if the entries of $\mathbf{X}$ strongly correlate. Numerical experiments from [16] confirm this counter-intuitive fact, cf. Fig. 3.2. A possible explanation is that by construction, $\tilde{\mathbf{\Sigma}}_n$ implicitly uses the assumption that $\mathbf{\Sigma}$ has ones on its diagonal which is not provided to $\hat{\mathbf{\Sigma}}_n$.

Furthermore, a corresponding lower bound on the second moment of the estimation error shows that the unconventional term $\|\sigma(\mathbf{M} \odot \mathbf{A})\|$ is factual and not an artifact of the proof, cf. [16, Proposition 14].

### *3.4.2 Dithered Quantization*

The results of Sect. 3.4.1 are restricted to the estimation of correlation matrices of Gaussian distributions. Both limitations stem from the chosen quantization model: first, (3.9) is blind to the rescaling of variances and, second, Grothendieck's identity only holds for Gaussian distributions. Nevertheless, by introducing a *dither* in the one-bit quantizer in (3.9), we can fully estimate the covariance matrix of general sub-Gaussian distributions. Dithering means adding artificial random noise (with

a suitably chosen distribution) to the samples before quantizing them to improve reconstruction from quantized observations, cf. [25, 26, 52]. In the context of one-bit compressed sensing, the effect of dithering was recently rigorously analyzed in [3, 17, 18, 32, 36], see also the survey [15].

To be precise, we require two bits per entry of each sample vector where each bit is dithered by an independent uniformly distributed dither, i.e., we are given

$$\operatorname{sign}(\mathbf{X}^k + \boldsymbol{\tau}^k),\ \operatorname{sign}(\mathbf{X}^k + \bar{\boldsymbol{\tau}}^k)^\top, \qquad k = 1, \dots, n, \tag{3.13}$$

where the dithering vectors $\boldsymbol{\tau}^1, \bar{\boldsymbol{\tau}}^1, \dots, \boldsymbol{\tau}^n, \bar{\boldsymbol{\tau}}^n$ are independent and uniformly distributed in $[-\lambda, \lambda]^p$, with $\lambda > 0$ to be specified later. From the quantized observations in (3.13), we construct the estimator

$$\tilde{\boldsymbol{\Sigma}}_n^{\mathrm{dith}} = \tfrac{1}{2}\tilde{\boldsymbol{\Sigma}}_n' + \tfrac{1}{2}(\tilde{\boldsymbol{\Sigma}}_n')^\top, \tag{3.14}$$

where

$$\tilde{\boldsymbol{\Sigma}}_n' = \frac{\lambda^2}{n}\sum_{k=1}^{n} \operatorname{sign}(\mathbf{X}^k + \boldsymbol{\tau}^k)\operatorname{sign}(\mathbf{X}^k + \bar{\boldsymbol{\tau}}^k)^\top. \tag{3.15}$$

**Theorem 3.8 ([16, Theorem 3])** *Let* $\mathbf{X}$ *be a mean-zero,* $K$*-sub-Gaussian vector with covariance matrix* $\mathbb{E}[\mathbf{X}\mathbf{X}^\top] = \boldsymbol{\Sigma}$*. Let* $\mathbf{X}^1, ..., \mathbf{X}^n \overset{\mathrm{d}}{\sim} \mathbf{X}$ *be i.i.d. samples of* $\mathbf{X}$*. Let* $\mathbf{M} \in [0,1]^{p\times p}$ *be a fixed symmetric mask. If* $\lambda^2 \gtrsim_K \log(n)\|\boldsymbol{\Sigma}\|_\infty$*, then with probability at least* $1 - e^{-t}$*,*

$$\begin{aligned}&\|\mathbf{M} \odot \tilde{\boldsymbol{\Sigma}}_n^{dith} - \mathbf{M} \odot \boldsymbol{\Sigma}\| \\ &\qquad \lesssim_K \|\mathbf{M}\|_{1\to 2}(\lambda\|\boldsymbol{\Sigma}\|^{1/2} + \lambda^2)\sqrt{\frac{\log(p)+t}{n}} + \lambda^2\|\mathbf{M}\|\frac{\log(p)+t}{n}.\end{aligned}$$

*In particular, if* $\lambda^2 \approx_K \log(n)\|\boldsymbol{\Sigma}\|_\infty$*, then*

$$\begin{aligned}&\|\mathbf{M} \odot \tilde{\boldsymbol{\Sigma}}_n^{dith} - \mathbf{M} \odot \boldsymbol{\Sigma}\| \\ &\lesssim_K \log(n)\|\mathbf{M}\|_{1\to 2}\sqrt{\frac{\|\boldsymbol{\Sigma}\|\,\|\boldsymbol{\Sigma}\|_\infty(\log(p)+t)}{n}} + \log(n)\|\mathbf{M}\|\|\boldsymbol{\Sigma}\|_\infty\frac{\log(p)+t}{n}.\end{aligned} \tag{3.16}$$

The error bound (3.16) coincides (up to different logarithmic factors) with the best known estimate for the masked sample covariance matrix in Theorem 3.1, even though the sample covariance matrix requires direct access to the samples $\mathbf{X}^k$. This performance, however, heavily depends on the choice of $\lambda$, cf. [16]. Furthermore, it should be mentioned that there are cases where the performance of the dithered

estimator is significantly worse than the performance of the sample covariance matrix. Let us consider for simplicity the case $\mathbf{M} = \mathbf{1}$. If the samples $\mathbf{X}^k$ are Gaussian, then [37] shows that

$$\mathbb{E}\|\hat{\mathbf{\Sigma}}_n - \mathbf{\Sigma}\| \simeq \sqrt{\frac{\|\mathbf{\Sigma}\|\mathrm{Tr}(\mathbf{\Sigma})}{n}} + \frac{\mathrm{Tr}(\mathbf{\Sigma})}{n},$$

whereas (3.16) yields

$$\mathbb{E}\|\tilde{\mathbf{\Sigma}}_n^{\mathrm{dith}} - \mathbf{\Sigma}\| \lesssim \log(n)\sqrt{\frac{p\|\mathbf{\Sigma}\|\,\|\mathbf{\Sigma}\|_\infty \log(p)}{n}} + \log(n)\frac{p\|\mathbf{\Sigma}\|_\infty \log(p)}{n}$$

via tail integration. Since $\mathrm{Tr}(\mathbf{\Sigma}) \leq p\|\mathbf{\Sigma}\|_\infty$, the second estimate is worse in general. Numerical experiments in [16] have shown that this difference is not an artifact of proof. Simply put, $\hat{\mathbf{\Sigma}}_n$ and $\tilde{\mathbf{\Sigma}}_n^{\mathrm{dith}}$ perform similarly if $\mathbf{\Sigma}$ has a constant diagonal, whereas $\hat{\mathbf{\Sigma}}_n$ performs significantly better whenever $\mathrm{Tr}(\mathbf{\Sigma}) \ll p\|\mathbf{\Sigma}\|_\infty$.

Theorem 3.8 can be extended to heavier-tailed random vectors. This, however, requires a larger choice of $\lambda$ and thus more samples to reach the same error. For a sub-exponential random vector $\mathbf{X}$, one would already need $\lambda^2 \gtrsim \log(n)^2 \cdot \max_{i\in[p]} \|X_i\|_{\psi_1}^2$. The dependence of $\lambda$ on $n$, both in the latter statement and in Theorem 3.8, can be observed in numerical experiments [16] as well.

Let us finally mention that the quantized estimators in (3.10) and (3.14) are not necessarily positive semi-definite as one expects from covariance matrices. In applications, one would thus replace both estimators by their projection onto the cone of positive semi-definite matrices, which is efficiently computed via the singular value decomposition [8, Section 8.1.1]. The obtained estimates also apply to the projected estimators since convex projections are 1-Lipschitz.

## 3.5 The Underlying Structures of Massive MIMO Covariance Estimation

Having the just surveyed theoretical insights on covariance matrix estimation in mind, let us return to the massive MIMO setup of Sect. 3.2. To understand the intrinsic structure of $\mathbf{\Sigma}_\mathbf{h}$ in (3.4) and consequent approaches in engineering literature to leverage it, we have to dive deeper into the underlying model and its physical interpretation. We thus follow the notational conventions of engineering literature in this section. Recall that the number of antennas $M$ can be identified with the ambient dimension $p$, the number of time–frequency resources $N$ with the number of samples $n$, and the channel vectors $\mathbf{h}(s)$ correspond to samples $\mathbf{X}^k$.

Under the assumptions stated in the beginning of Sect. 3.2, i.e., the BS is equipped with $M$ antennas in a ULA, the channel vector $\mathbf{h}(s)$ can be written explicitly as

$$\mathbf{h}(s) = \int_{-1}^{1} \rho(\xi, s)\, \mathbf{a}(\xi)\, \mathrm{d}\xi,$$

for $s \in [N]$. Here, $\xi = \frac{\sin(\theta)}{\sin(\theta_{\max})}$ are the normalized *angles of arrival (AoA)* with $\theta_{\max} \in [0, \frac{\pi}{2}]$ being the maximum array angular aperture (cf. Fig. 3.1), the vectors $\mathbf{a}(\xi) \in \mathbb{C}^M$ denote the respective array response at the BS, and the *channel gain* $\rho(\xi, s)$ is a complex Gaussian process with zero mean. By assuming the antenna spacing to be $d = \frac{\lambda}{2}$, where $\lambda = \frac{c_0}{f_0}$ denotes the wavelength with $c_0$ being the speed of light and $f_0$ the carrier frequency, we obtain

$$\mathbf{a}(\xi) = \left(1, e^{j\pi\xi}, \ldots, e^{j\pi(M-1)\xi}\right)^{\top},$$

where $j$ denotes the imaginary unit. With the additional assumption of *wide sense stationary uncorrelated scattering (WSSUS)*, the second-order statistics of the Gaussian process $\rho(\xi, s)$ are time invariant and uncorrelated across AoAs so that

$$\mathbb{E}[\rho(\xi, s)\rho^*(\xi', s)] = \gamma(\xi)\, \delta(\xi - \xi'),$$

where $\gamma\colon [-1, 1] \to \mathbb{R}_{\geq 0}$ is the real and non-negative measure that represents the *angular scattering function (ASF)* and $\delta$ is the Dirac delta function. In particular, this implies that

$$\mathbf{\Sigma}_{\mathbf{h}} = \mathbb{E}[\mathbf{h}(s)\mathbf{h}(s)^{\mathsf{H}}] = \int_{-1}^{1} \gamma(\xi)\, \mathbf{a}(\xi)\mathbf{a}(\xi)^{\mathsf{H}}\, \mathrm{d}\xi. \tag{3.17}$$

Building upon this explicit representation of $\mathbf{h}(s)$ and structural assumptions on $\gamma$, one can refine the estimate obtained from the sample covariance matrix of $\mathbf{y}$ defined in (3.3).

**A Hands-on Approach** In [35], we choose the following approach. First note that by (3.17) the channel covariance matrix belongs to the set

$$\mathcal{M} = \left\{ \int_{-1}^{1} \gamma(\xi)\, \mathbf{a}(\xi)\mathbf{a}(\xi)^{\mathsf{H}}\, \mathrm{d}\xi : \gamma \in \mathcal{A} \right\},$$

where $\mathcal{A}$ denotes the class of typical ASFs in wireless propagation. If one assumes sparse scattering propagation, the set $\mathcal{A}$ consists of sparse ASFs. In particular, we assume that $\gamma(\xi)$ can be decomposed as the sum of a discrete spike component $\gamma_d$ (modeling the power received from *line-of-sight (LOS)* paths and narrow scatterers) and a continuous component $\gamma_c$ (modeling the power received from wide scatterers). Mathematically, we can write

$$\gamma(\xi) = \gamma_d(\xi) + \gamma_c(\xi) = \sum_{k=1}^{r} c_k \delta(\xi - \xi_k) + \gamma_c(\xi), \tag{3.18}$$

where $\gamma_d$ consists of $r \ll M$ Dirac deltas with AoAs $\xi_1, \dots, \xi_r$ and strengths $c_1, \dots, c_r > 0$ corresponding to $r$ specular propagation elements. Furthermore, by sparsity assumptions on $\gamma$, we have that $\mathrm{meas}(\gamma_c) \ll \mathrm{meas}([-1, 1])$, where $\mathrm{meas}(\gamma_c)$ denotes here the measure of the support of $\gamma_c$. Combining (3.17) and (3.18), we decompose the channel covariance matrix as

$$\mathbf{\Sigma_h} = \mathbf{\Sigma}_\mathbf{h}^d + \mathbf{\Sigma}_\mathbf{h}^c = \sum_{k=1}^{r} c_k \, \mathbf{a}(\xi_k)\mathbf{a}(\xi_k)^\mathsf{H} + \int_{-1}^{1} \gamma_c(\xi) \, \mathbf{a}(\xi)\mathbf{a}(\xi)^\mathsf{H} \, \mathrm{d}\xi, \tag{3.19}$$

where $\mathbf{\Sigma}_\mathbf{h}^d$ is rank-$r$ and positive semi-definite and $\mathbf{\Sigma}_\mathbf{h}^c$ is full rank and positive semi-definite with few dominant singular values. We can approximate $\mathbf{\Sigma_h}$ now in three consecutive steps:

(i) *Spike Location Estimation for $\gamma_d$:* Applying the *MUltiple SIgnal Classification (MUSIC)* algorithm [58], we estimate the AoAs $\xi_k$ of the spike component $\gamma_d$ from the noisy samples $\mathbf{y}(1), \dots, \mathbf{y}(N)$, cf. [35, Theorem 1]. Since this step is fairly standard, we do not discuss the details here but refer the interested reader to [35]. Let us only mention that the number of spikes is estimated by the number of dominant eigenvalues of $\mathbf{\Sigma_y} := \mathbb{E}[\mathbf{y}(s)\mathbf{y}(s)^\mathsf{H}]$ (where one can naturally assume a corresponding gap in the spectrum since the power received via LOS paths in $\gamma_d$ dominates the power received from wide scatterers in $\gamma_c$). As a result, we obtain estimated spike locations $\hat{\xi}_k$, for $k \in [\hat{r}]$, and define an approximation of $\gamma_d$

$$\tilde{\gamma}_d(\xi) = \sum_{k=1}^{\hat{r}} \tilde{c}_k \, \delta(\xi - \hat{\xi}_k),$$

where the coefficients $\tilde{c}_1, \dots, \tilde{c}_{\hat{r}} \geq 0$ still need to be estimated.

(ii) *Sparse Dictionary-Based Method:* We approximate the continuous component $\gamma_c$ over a finite dictionary of densities $\mathcal{G}_c := \{\psi_i \colon [-1, 1] \to \mathbb{R}, \ i \in [G]\}$ that are suitably chosen, e.g., Gaussian, Laplacian, or rectangular kernels, cf. Fig. 3.3. We hence define

$$\tilde{\gamma}_c(\xi) = \sum_{i=1}^{G} \tilde{b}_i \psi_i(\xi),$$

where only the coefficients $\tilde{b}_1, \dots, \tilde{b}_G \geq 0$ need to be estimated.

(iii) *Non-Negative Least Square (NNLS) estimator:* Collecting the coefficients in a single vector $\mathbf{u} = (\tilde{b}_1, \dots, \tilde{b}_G, \tilde{c}_1, \dots, \tilde{c}_{\hat{r}})^\top \in \mathbb{R}^{G+\hat{r}}_{\geq 0}$ and recalling (3.19), we define our coefficient-dependent estimate of the channel covariance

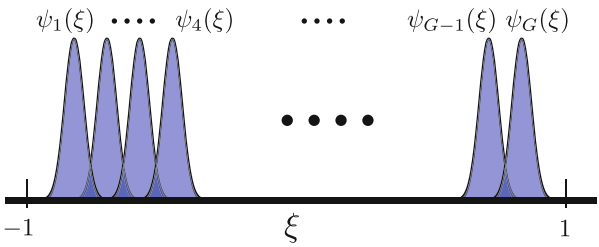



**Fig. 3.3** Example of a Gaussian dictionary that might be used to express $\gamma_c$

$$\mathbf{\Sigma_h}(\mathbf{u}) = \sum_{k=1}^{\hat{r}} \tilde{c}_k \, \mathbf{a}(\hat{\xi}_k)\mathbf{a}(\hat{\xi}_k)^{\mathsf{H}} + \sum_{i=1}^{G} \tilde{b}_i \int_{-1}^{1} \psi_i(\xi) \, \mathbf{a}(\xi)\mathbf{a}(\xi)^{\mathsf{H}} \, \mathrm{d}\xi =: \sum_{i=1}^{G+\hat{r}} u_i \mathbf{S}_i, \tag{3.20}$$

where

$$\mathbf{S}_i = \begin{cases} \int_{-1}^{1} \psi_i(\xi) \, \mathbf{a}(\xi)\mathbf{a}(\xi)^{\mathsf{H}} \, \mathrm{d}\xi & \text{if } 1 \le i \le G \\ \mathbf{a}(\hat{\xi}_{i-G})\mathbf{a}(\hat{\xi}_{i-G})^{\mathsf{H}} & \text{if } G < i \le G + \hat{r}. \end{cases}$$

All that remains is to determine the coefficient vector $\mathbf{u}$. Since $\mathbf{\Sigma_y} = \mathbf{\Sigma_h} + N_0\mathbf{I}$, we can do so by fitting (3.20) to the sample covariance matrix $\hat{\mathbf{\Sigma}}_\mathbf{y}$ of $\mathbf{y}(1), \ldots, \mathbf{y}(N)$, i.e.,

$$\mathbf{u}^* = \arg\min_{\mathbf{u} \ge \mathbf{0}} \left\| \hat{\mathbf{\Sigma}}_\mathbf{y} - \sum_{i=1}^{G+\hat{r}} u_i \mathbf{S}_i - N_0\mathbf{I} \right\|_F^2. \tag{3.21}$$

Since $\mathbf{\Sigma_h}$ is Hermitian Toeplitz, one can incorporate the structure in (3.21) by replacing $\hat{\mathbf{\Sigma}}_\mathbf{h} = \hat{\mathbf{\Sigma}}_\mathbf{y} - N_0\mathbf{I}$ with its projection $\tilde{\mathbf{\Sigma}}_\mathbf{h}$ onto the space of Hermitian Toeplitz matrices (which can be done by averaging the diagonals as in (3.7)). Denoting the first column of $\tilde{\mathbf{\Sigma}}_\mathbf{h}$ by $\tilde{\boldsymbol{\sigma}} \in \mathbb{C}^M$ and collecting the first columns of the matrices $\mathbf{S}_i$ in a matrix $\tilde{\mathbf{S}} \in \mathbb{C}^{M \times (G+\hat{r})}$, we may instead solve

$$\mathbf{u}^* = \arg\min_{\mathbf{u} \ge \mathbf{0}} \left\| \mathbf{W}(\tilde{\mathbf{S}}\mathbf{u} - \tilde{\boldsymbol{\sigma}}) \right\|_F^2, \tag{3.22}$$

where $\mathbf{W} = \mathrm{diag}\big((\sqrt{M}, \sqrt{2(M-1)}, \sqrt{2(M-2)}, ..., \sqrt{2})^\top\big)$ is a weight matrix compensating the averaging process.

**A Hands-on Approach: Empirical Evaluation** Let us empirically compare the NNLS estimator to the sample covariance matrix right away. We consider a ULA with $M = 128$ antennas, where the spacing between two consecutive antenna

elements is set to $d = \frac{\lambda}{2}$. We produce random ASFs in the following general format:

$$\gamma(\xi) = \gamma_d(\xi) + \gamma_c(\xi) = \frac{\alpha}{r} \sum_{i=1}^{r} \delta(\xi - \xi_i) + \frac{1-\alpha}{Z} \left( \sum_{j=1}^{n_r} \texttt{rect}_{\mu_j,\sigma_j}(\xi) + \sum_{k=1}^{n_g} \texttt{Gaussian}_{\mu_k,\sigma_k}(\xi) \right), \tag{3.23}$$

where we set the number of delta, rectangular, and Gaussian functions to be $r := 2, n_r := 2$, and $n_g := 2$, respectively. The spike locations are chosen uniformly at random from $[-1, 1]$, i.e., $\xi_i \sim \text{Unif}([-1, 1])$ for $i \in [2]$. The rectangular functions are defined as

$$\texttt{rect}_{\mu_j,\sigma_j}(\xi) = \chi_{\left[\mu_j - \frac{\sigma_j}{2}, \mu_j + \frac{\sigma_j}{2}\right]}(\xi),$$

where $\mu_1 \sim \text{Unif}([-1, 0])$, $\mu_2 \sim \text{Unif}([0, 1])$, and $\sigma_j \sim \text{Unif}([0.1, 0.3])$, for $j \in [2]$. The Gaussian functions $\texttt{Gaussian}_{\mu_k,\sigma_k}$ are densities of $\mathcal{N}(\mu_k, \sigma_k)$, where $\mu_k \sim \text{Unif}([-0.7, 0.7])$ and $\sigma_k \sim \text{Unif}([0.03, 0.04])$, for $k \in [2]$. Moreover, $\alpha := 0.5$ is set to present the power contribution of discrete spikes. The constant $Z = \int_{-1}^{1} \gamma_c(\xi) d\xi$ normalizes $\gamma_c$ in measure. The signal-to-noise ratio (SNR) is set to 10 dB.

In addition to the sample covariance, we compare our NNLS estimator to sparse iterative covariance-based estimation (SPICE) [57]. This method also exploits the ASF domain to fit a covariance matrix. Note that SPICE can only be applied with Dirac delta dictionaries and that it does not include a step of spike support detection as in our method.

Denoting a generic covariance estimate as $\bar{\mathbf{\Sigma}}$, we consider two metrics to evaluate the estimation quality. The first metric, the *normalized Frobenius-norm error*, is defined as $E_{\text{NF}} = \frac{\|\mathbf{\Sigma_h} - \bar{\mathbf{\Sigma}}\|_{\text{F}}}{\|\mathbf{\Sigma_h}\|_{\text{F}}}$. The second metric, the *power efficiency*, evaluates the similarity of dominant subspaces between the estimated and true matrices, which is an important factor in various applications of massive MIMO such as user grouping and group-based beamforming. Specifically, let $d \in [M]$ denote a subspace dimension parameter, and let $\mathbf{U}_d \in \mathbb{C}^{M \times d}$ and $\bar{\mathbf{U}}_d \in \mathbb{C}^{M \times d}$ be the $d$ dominant eigenvectors of $\mathbf{\Sigma_h}$ and $\bar{\mathbf{\Sigma}}$ corresponding to their largest $d$ eigenvalues, respectively. Then, the power efficiency based on $d$ is defined as $E_{\text{PE}}(d) = 1 - \frac{\langle \mathbf{\Sigma_h}, \bar{\mathbf{U}}_d \bar{\mathbf{U}}_d^{\text{H}} \rangle}{\langle \mathbf{\Sigma_h}, \mathbf{U}_d \mathbf{U}_d^{\text{H}} \rangle}$. Note that $E_{\text{PE}}(d) \in [0, 1]$ where a value closer to 0 means that more power is captured by the estimated $d$-dominant subspace.

SPICE and the proposed NNLS estimators are applied with $G = 2M$ Dirac delta dictionaries for the continuous part $\mathcal{G}_c$. The resulting Frobenius-norm error and power efficiency are depicted in Fig. 3.4. All results are averaged over 20 random ASFs and 200 random channel realizations for each ASF. The proposed NNLS method outperforms the sample covariance matrix and SPICE for both metrics.

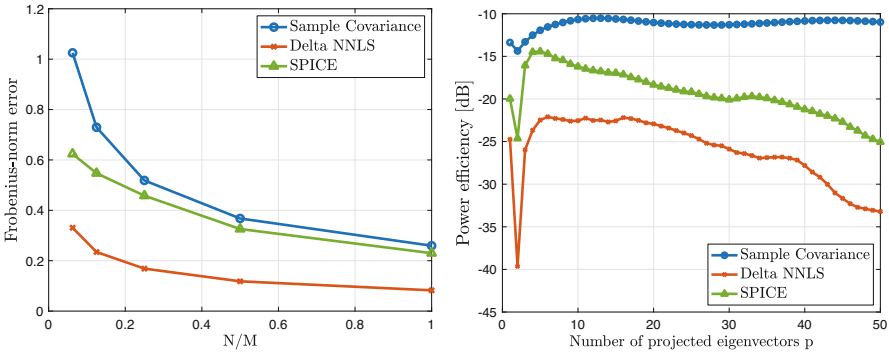



**Fig. 3.4** Frobenius-norm error (left) and power efficiency with $\frac{N}{M} = 0.5$ (right)

Finally, one can observe a similar outcome for smaller sample sizes as well, e.g., $N/M = 0.125$, which occur naturally in massive MIMO.

## 3.6 Conclusion

The present chapter shows that in the last decade good progress has been made on understanding the influence of intrinsic structure of covariance matrices on the non-asymptotic performance of suitably designed estimators. As we have seen, such estimators with strong guarantees are available for sparse, low-rank, and Toeplitz covariance matrices. At the same time, the chapter illustrates that practitioners still continue to tweak the basic sample covariance matrix using their specific knowledge of the application at hand—seemingly unaware of the progress in theory. We hope that this essay helps mathematicians and practitioners alike to gain an overview of recent theoretical developments on structural and quantized covariance estimation and that it motivates mathematicians to look deeper into the underlying physical models of concrete applications to better understand the structures of interest. Furthermore, our recent theoretical progress on quantized covariance estimation suggests that reliable reconstruction of the covariance matrix is possible even under heavy loss of information during sampling. The use of coarse quantization might thus lead to a considerable increase in capacity in massive MIMO systems and related applications.

**Acknowledgments** All authors acknowledge funding by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) through the project CoCoMIMO funded within the priority program SPP 1798 Compressed Sensing in Information Processing (COSIP).

## Appendix: Proof of Theorem 3.6

To prove Theorem 3.6, we need two technical lemmas. In the remaining section, $\boldsymbol{\sigma}$ always refers to the first column of $\boldsymbol{\Sigma}$ and $\hat{\boldsymbol{\sigma}}$ to the first column of $\hat{\boldsymbol{\Sigma}}_n^{\mathrm{Toep}}$.

**Lemma 3.1** *Under the assumptions of Theorem 3.6, we have for $\alpha \in (0,1)$ and $0 < u < 1$ that*

$$\Pr\left[\max_{r\le\alpha p} |\hat{\sigma}_r - \sigma_r| \ge \sqrt{u}\right] \le 2\alpha p e^{-(1-\alpha)\min\left\{\frac{1}{CK^4}, \frac{1}{CK^2}\right\} npu},$$

*where $C > 0$ is an absolute constant.*

***Proof*** We proceed similar as in [33]. First note that, for all $k \in [n], r \in [\alpha p]$, we can write

$$\begin{aligned} |\hat{\sigma}_r - \sigma_r| &= \frac{1}{(p+1)-r} \left| \sum_{j-i=r-1} \left( X_i^k X_j^k - \sigma_r \right) \right| \\ &= \left| \langle \mathbf{M}_r \mathbf{X}^k, \mathbf{X}^k \rangle - \mathbb{E}[\langle \mathbf{M}_r \mathbf{X}^k, \mathbf{X}^k \rangle] \right|, \end{aligned} \tag{3.24}$$

where the mask $\mathbf{M}_r$ is defined by $[M_r]_{i,j} = \frac{1}{(p+1)-r}$ if $j - i = r - 1$ and $[M_r]_{i,j} = 0$ else, i.e., only the $r$-th co-diagonal of $\mathbf{M}$ is non-zero. By using a version of the Hanson–Wright inequality for random vectors with the convex concentration property [1], we get that

$$\Pr\Big[\Big|\underbrace{\langle \mathbf{M}_r \mathbf{X}^k, \mathbf{X}^k \rangle - \mathbb{E}[\langle \mathbf{M}_r \mathbf{X}^k, \mathbf{X}^k \rangle]}_{=: Z_i^r}\Big| \ge u\Big] \le 2e^{-\min\left\{\frac{u^2}{CK^4\|\mathbf{M}_r\|_F^2}, \frac{u}{CK^2\|\mathbf{M}_r\|}\right\}},$$

which, by integration, leads to

$$\begin{aligned} \mathbb{E}[|Z_i^r|^{2q}] &\le 2q(2CK^4\|\mathbf{M}_r\|_F^2)^q \Gamma(q) + 4q(CK^2\|\mathbf{M}_r\|)^{2q}\Gamma(2q) \\ &\le q!(4CK^4\|\mathbf{M}_r\|_F^2)^q + (2q)!(2CK^2\|\mathbf{M}_r\|)^{2q}, \end{aligned}$$

for any $q \ge 1$. The random variables $Z_i^r$ are thus sub-gamma with variance parameter $\nu = 16K^4(C\|\mathbf{M}_r\|_F^2 + C^2\|\mathbf{M}_r\|^2) \le CK^4\|\mathbf{M}_r\|_F^2$ and scale parameter $\gamma = 4CK^2\|\mathbf{M}_r\|^2$ [7, Theorem 2.3]. By independence, we get for all $0 < \mu < \frac{1}{\gamma}$

$$\mathbb{E}\left[e^{\mu\sum_{i=1}^n Z_i^r}\right] = \prod_{i=1}^n \mathbb{E}[e^{\mu Z_i^r}] \le e^{\frac{\mu^2 n\nu}{2(1-\gamma\mu)}}$$

(and the same holds for $-Z_i^r$) such that $\sum_{i=1}^n Z_i^r$ is sub-gamma with variance parameter $vn$ and scale parameter $\gamma$ [7, Chapter 2.4]. Consequently,

$$\Pr\left[\left|\frac{1}{n}\sum_{i=1}^n Z_i^r\right| \geq CK^2\left(\|\mathbf{M}_r\|_F\sqrt{\frac{u}{n}} + \|\mathbf{M}_r\|\frac{u}{n}\right)\right]$$
$$\leq \Pr\left[\left|\sum_{i=1}^n Z_i^r\right| \geq \sqrt{2vnu} + \gamma u\right] \leq 2e^{-u},$$

for any $u > 0$ [7, Chapter 2.4]. Recalling (3.24) and noting that $\|\mathbf{M}_r\|_F^2 = \|\mathbf{M}_r\| = \frac{1}{(p+1)-r}$ yield with the choice $u = \min\left\{\frac{1}{C^2K^4}, \frac{1}{CK^2}\right\}((p+1)-r)n\tilde{u}$ that

$$\Pr\left[|\tilde{\sigma}_r - \sigma_r| \geq 2\sqrt{\tilde{u}}\right] \leq \Pr\left[|\tilde{\sigma}_r - \sigma_r| \geq \sqrt{\tilde{u}} + \tilde{u}\right]$$
$$\leq 2e^{-\min\left\{\frac{1}{C^2K^4}, \frac{1}{CK^2}\right\}((p+1)-r)n\tilde{u}}.$$

A union bound over $r \in [\alpha p]$ and the bound $r \leq \alpha p$ conclude the proof. □

The second lemma follows along the lines of [5, Theorem 1].

**Lemma 3.2** *Under the assumptions of Theorem 3.6, assume in addition that* $\mathbf{\Sigma}$ *has a bandwidth of at most* $\alpha p$*, i.e.,* $\mathbb{B}_{\alpha p}(\mathbf{\Sigma}) = \mathbf{\Sigma}$ *and thus* $\operatorname{supp}(\boldsymbol{\sigma}) \subset [\alpha p]$ *and that*

$$\max_{i,j\in[p]} |\mathbb{B}_{\alpha p}(\hat{\Sigma}_n^{\mathrm{Toep}})_{i,j} - \Sigma_{i,j}| = \max_{r\leq\alpha p} |\hat{\sigma}_r - \sigma_r| \leq (1-\gamma)\tau, \tag{3.25}$$

*for some* $\gamma \in (0,1)$*. Then,*

$$\|\mathbb{T}_\tau(\mathbb{B}_{\alpha p}(\hat{\mathbf{\Sigma}}_n^{\mathrm{Toep}})) - \mathbf{\Sigma}\| \lesssim_\gamma \tau^{1-q}s.$$

***Proof*** For convenience, let us abbreviate $\tilde{\mathbf{\Sigma}} := \mathbb{B}_{\alpha p}(\hat{\mathbf{\Sigma}}_n^{\mathrm{Toep}})$ and denote its first column by $\tilde{\boldsymbol{\sigma}}$. We write

$$\|\mathbb{T}_\tau(\tilde{\mathbf{\Sigma}}) - \mathbf{\Sigma}\| \leq \|\mathbb{T}_\tau(\mathbf{\Sigma}) - \mathbf{\Sigma}\| + \|\mathbb{T}_\tau(\tilde{\mathbf{\Sigma}}) - \mathbb{T}_\tau(\mathbf{\Sigma})\|.$$

Since $\mathbf{\Sigma} \in \mathcal{U}^{\mathrm{Toep}}(q,s,M)$,

$$\sum_{j=1}^p |\Sigma_{i,j}|\chi_{\{|\Sigma_{i,j}|\leq\tau\}} = \sum_{j=1}^p |\Sigma_{i,j}|^q|\Sigma_{i,j}|^{1-q}\chi_{\{|\Sigma_{i,j}|\leq\tau\}} \leq \tau^{1-q}\sum_{j=1}^p |\Sigma_{i,j}|^q$$

and Gershgorin's disc theorem imply

$$\|\mathbb{T}_\tau(\mathbf{\Sigma}) - \mathbf{\Sigma}\| \leq \max_i \sum_{j=1}^{p} |\Sigma_{i,j}| \chi_{\{|\Sigma_{i,j}| \leq \tau\}} \leq \tau^{1-q} s. \tag{3.26}$$

Moreover,

$$\begin{aligned}
\|\mathbb{T}_\tau(\tilde{\mathbf{\Sigma}}) - \mathbb{T}_\tau(\mathbf{\Sigma})\| &\leq \max_i \sum_{j=1}^{p} |\tilde{\Sigma}_{i,j}| \chi_{\{|\tilde{\Sigma}_{i,j}| \geq \tau,\ |\Sigma_{i,j}| < \tau\}} \\
&\quad + \max_i \sum_{j=1}^{p} |\Sigma_{i,j}| \chi_{\{|\tilde{\Sigma}_{i,j}| < \tau,\ |\Sigma_{i,j}| \geq \tau\}} \\
&\quad + \max_i \sum_{j=1}^{p} |\tilde{\Sigma}_{i,j} - \Sigma_{i,j}| \chi_{\{|\tilde{\Sigma}_{i,j}| \geq \tau,\ |\Sigma_{i,j}| \geq \tau\}} \\
&= (I) + (II) + (III).
\end{aligned}$$

First recall that by assumption $\mathrm{supp}(\boldsymbol{\sigma}) \subset [\alpha p]$ and $\mathrm{supp}(\tilde{\boldsymbol{\sigma}}) \subset [\alpha p]$. Hence, using the observation that $\tilde{\sigma}_r = \hat{\sigma}_r$, for $r \leq \alpha p$, and

$$\sum_{j=1}^{p} \chi_{\{|\Sigma_{i,j}| \geq \tau\}} = \sum_{j=1}^{p} \tau^q \tau^{-q} \chi_{\{|\Sigma_{i,j}| \geq \tau\}} \leq \sum_{j=1}^{p} |\Sigma_{i,j}|^q \tau^{-q}, \tag{3.27}$$

we may estimate with (3.25) and $\mathbf{\Sigma} \in \mathcal{U}^{\mathrm{Toep}}(q, s, M)$

$$(III) \leq \max_{r \leq \alpha p} |\hat{\sigma}_r - \sigma_r| \cdot \max_i \sum_{j=1}^{p} |\Sigma_{i,j}|^q \tau^{-q} \leq s \tau^{1-q}.$$

Furthermore,

$$\begin{aligned}
(I) &\leq \max_i \sum_{j=1}^{p} |\tilde{\Sigma}_{i,j} - \Sigma_{i,j}| \chi_{\{|\tilde{\Sigma}_{i,j}| \geq \tau,\ |\Sigma_{i,j}| < \tau\}} + \max_i \sum_{j=1}^{p} |\Sigma_{i,j}| \chi_{\{|\tilde{\Sigma}_{i,j}| \geq \tau,\ |\Sigma_{i,j}| < \tau\}} \\
&= (IV) + (V).
\end{aligned}$$

By (3.26), we know that $(V) \leq \tau^{1-q} s$. Furthermore, we get that

$$(IV) \leq \max_i \sum_{j=1}^{p} |\tilde{\Sigma}_{i,j} - \Sigma_{i,j}| \chi_{\{|\tilde{\Sigma}_{i,j}| \geq \tau,\ |\Sigma_{i,j}| < \gamma\tau\}}$$

$$+ \max_i \sum_{j=1}^{p} |\tilde{\Sigma}_{i,j} - \Sigma_{i,j}| \chi_{\{|\tilde{\Sigma}_{i,j}| \geq \tau,\ \gamma\tau \leq |\Sigma_{i,j}| < \tau\}}$$

$$\leq \max_{r \leq \alpha p} |\tilde{\sigma}_r - \sigma_r| \cdot \max_i N_i(1-\gamma) + s(\gamma\tau)^{-q}\tau,$$

where we defined $N_i(1-\gamma) := \sum_{j=1}^{p} \chi_{\{|\tilde{\Sigma}_{i,j} - \Sigma_{i,j}| > (1-\gamma)\tau\}}$ and reused the bound on $(III)$ for the second term (replacing $\tau$ with $\gamma\tau$ in the summation). Since we have by (3.25) that $N_i(1-\gamma) = 0$, for $i \in [p]$, we get that $(IV) \lesssim_\gamma s\tau^{1-q}$. Hence,

$$(I) \lesssim_\gamma s\tau^{1-q}.$$

Finally, note that by (3.27) and $\mathbf{\Sigma} \in \mathcal{U}^{\mathrm{Toep}}(q, s, M)$,

$$\begin{aligned}(II) &\leq \max_i \sum_{j=1}^{p} (|\tilde{\Sigma}_{i,j} - \Sigma_{i,j}| + |\tilde{\Sigma}_{i,j}|) \chi_{\{|\tilde{\Sigma}_{i,j}| < \tau,\ |\Sigma_{i,j}| \geq \tau\}} \\ &\leq \max_{r \leq \alpha p} |\tilde{\sigma}_r - \sigma_r| \cdot \max_i \sum_{j=1}^{p} \chi_{\{|\Sigma_{i,j}| \geq \tau\}} + \tau \max_i \sum_{j=1}^{p} \chi_{\{|\Sigma_{i,j}| \geq \tau\}} \\ &\leq s\tau^{1-q} + s\tau^{1-q}.\end{aligned}$$

Combining the bounds for $(I)$, $(II)$, and $(III)$ yields the claim. □

***Proof of Theorem 3.6*** Note that

$$\|\mathbb{T}_\tau(\mathbb{B}_{\alpha p}(\hat{\mathbf{\Sigma}}_n^{\mathrm{Toep}})) - \mathbf{\Sigma}\| \leq \|\mathbb{T}_\tau(\mathbb{B}_{\alpha p}(\hat{\mathbf{\Sigma}}_n^{\mathrm{Toep}})) - \mathbb{B}_{\alpha p}(\mathbf{\Sigma})\| + \|\mathbb{B}_{\alpha p}(\mathbf{\Sigma}) - \mathbf{\Sigma}\|. \tag{3.28}$$

By Lemma 3.1, we get with probability at least $1 - (2\alpha p)^{-(c-1)}$ that

$$\max_{r \leq \alpha p} |\hat{\sigma}_r - \sigma_r| \leq \sqrt{\frac{c}{1-\alpha}} \max\{CK^2, \sqrt{C}K\} \sqrt{\frac{\log(p)}{np}} = (1-\gamma)\tau, \tag{3.29}$$

where $c > 1$ and $1 - \gamma = \frac{1}{\sqrt{2}}$. The claim now follows by applying Lemma 3.2 to the first term on the right-hand side of (3.28). □

## References

1. Adamczak, R.: A note on the Hanson-Wright inequality for random vectors with dependencies. Electron. Commun. Probab. **20** (2015)

2. Bar-Shalom, O., Weiss, A.J.: DOA estimation using one-bit quantized measurements. IEEE Trans. Aerospace Electron. Syst. **38**(3), 868–884 (2002)
3. Baraniuk, R.G., Foucart, S., Needell, D., Plan, Y., Wootters, M.: Exponential decay of reconstruction error from binary measurements of sparse signals. IEEE Trans. Inform. Theory **63**(6), 3368–3385 (2017)
4. Benedetto, J.J., Powell, A.M., Yilmaz, O.: Sigma-delta quantization and finite frames. IEEE Trans. Inform. Theory **52**(5), 1990–2005 (2006)
5. Bickel, P.J., Levina, E.: Covariance regularization by thresholding. Ann. Stat. **36**(6), 2577–2604 (2008)
6. Bickel, P.J., Levina, E.: Regularized estimation of large covariance matrices. Ann. Stat. **36**(1), 199–227 (2008)
7. Boucheron, S., Lugosi, G., Massart, P.: Concentration inequalities: a nonasymptotic theory of independence. Oxford University Press, Oxford (2013)
8. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press, Cambridge (2004)
9. Brookes, M.J., Vrba, J., Robinson, S.E., Stevenson, C.M., Peters, A.M., Barnes, G.R., Hillebrand, A., Morris, P.G.: Optimising experimental design for MEG beamformer imaging. Neuroimage **39**(4), 1788–1802 (2008)
10. Cai, T.T., Ren, Z., Zhou, H.H.: Optimal rates of convergence for estimating Toeplitz covariance matrices. Probab. Theory Related Fields **156**(1-2), 101–143 (2013)
11. Cai, T.T., Zhang, C.H., Zhou, H.H.: Optimal rates of convergence for covariance matrix estimation. Ann. Stat. **38**(4), 2118–2144 (2010)
12. Catoni, O.: Challenging the empirical mean and empirical variance: a deviation study. In: Annales de l'IHP Probabilités et statistiques, vol. 48, pp. 1148–1185 (2012)
13. Chen, R.Y., Gittens, A., Tropp, J.A.: The masked sample covariance estimator: an analysis using matrix concentration inequalities. Inform. Inference J. IMA **1**(1), 2–20 (2012)
14. Choi, J., Mo, J., Heath, R.W.: Near maximum-likelihood detector and channel estimator for uplink multiuser massive MIMO systems with one-bit ADCs. IEEE Trans. Commun. **64**(5), 2005–2018 (2016)
15. Dirksen, S.: Quantized compressed sensing: a survey. In: Compressed Sensing and Its Applications: Third International MATHEON Conference 2017, pp. 67–95. Applied and Numerical Harmonic Analysis. Birkhäuser, Cham (2019)
16. Dirksen, S., Maly, J., Rauhut, H.: Covariance estimation under one-bit quantization. arXiv preprint arXiv:2104.01280 (2021)
17. Dirksen, S., Mendelson, S.: Robust one-bit compressed sensing with partial circulant matrices. ArXiv:1812.06719 (2018)
18. Dirksen, S., Mendelson, S.: Non-gaussian hyperplane tessellations and robust one-bit compressed sensing. J. Eur. Math. Soc. Arxiv: 1805.09409 (2021)
19. El Karoui, N.: Operator norm consistent estimation of large-dimensional sparse covariance matrices. Ann. Stat. **36**(6), 2717–2756 (2008)
20. Eldar, Y.C., Li, J., Musco, C., Musco, C.: Sample efficient Toeplitz covariance estimation. In: Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 378–397. SIAM (2020)
21. Fazel, M.: Matrix rank minimization with applications. Ph.D. Thesis, Stanford University (2002)
22. Foucart, S., Rauhut, H.: A Mathematical Introduction to Compressive Sensing. Applied and Numerical Harmonic Analysis. Birkhäuser, Basel (2013)
23. Furrer, R., Bengtsson, T.: Estimation of high-dimensional prior and posterior covariance matrices in Kalman filter variants. J. Multivariate Anal. **98**(2), 227–255 (2007)
24. Goldsmith, A., Jafar, S.A., Jindal, N., Vishwanath, S.: Capacity limits of MIMO channels. IEEE J. Selected Areas Commun. **21**(5), 684–702 (2003)
25. Gray, R.M., Neuhoff, D.L.: Quantization. IEEE Trans. Inform. Theory **44**(6), 2325–2383 (1998)

26. Gray, R.M., Stockham, T.G.: Dithered quantizers. IEEE Trans. Inform. Theory **39**(3), 805–812 (1993)
27. Haghighatshoar, S., Caire, G.: Massive MIMO channel subspace estimation from low-dimensional projections. IEEE Trans. Signal Process. **65**(2), 303–318 (2016)
28. Haghighatshoar, S., Caire, G.: Low-complexity massive MIMO subspace estimation and tracking from low-dimensional projections. IEEE Trans. Signal Process. **66**(7), 1832–1844 (2018)
29. Hubert, M., Rousseeuw, P.J., Van Aelst, S.: High-breakdown robust multivariate methods. Statistical Science, pp. 92–119 (2008)
30. Jacovitti, G., Neri, A.: Estimation of the autocorrelation function of complex Gaussian stationary processes by amplitude clipped signals. IEEE Trans. Inform. Theory **40**(1), 239–245 (1994)
31. Jerrum, M.R., Valiant, L.G., Vazirani, V.V.: Random generation of combinatorial structures from a uniform distribution. Theor. Comput. Sci. **43**, 169–188 (1986)
32. Jung, H.C., Maly, J., Palzer, L., Stollenwerk, A.: Quantized compressed sensing by rectified linear units. IEEE Trans. Inform. Theory **67**(6), 4125–4149 (2021)
33. Kabanava, M., Rauhut, H.: Masked Toeplitz covariance estimation. ArXiv:1709.09377 (2017)
34. Ke, Y., Minsker, S., Ren, Z., Sun, Q., Zhou, W.X.: User-friendly covariance estimation for heavy-tailed distributions. Stat. Sci. **34**(3), 454–471 (2019)
35. Khalilsarai, M.B., Yang, T., Haghighatshoar, S., Caire, G.: Structured channel covariance estimation from limited samples in massive MIMO. In: IEEE International Conference on Communications (ICC), pp. 1–7 (2020)
36. Knudson, K., Saab, R., Ward, R.: One-bit compressive sensing with norm estimation. IEEE Trans. Inform. Theory **62**(5), 2748–2758 (2016)
37. Koltchinskii, V., Lounici, K.: Concentration inequalities and moment bounds for sample covariance operators. Bernoulli **23**(1), 110–133 (2017)
38. Krim, H., Viberg, M.: Two decades of array signal processing research: the parametric approach. IEEE Signal Process. Mag. **13**(4), 67–94 (1996)
39. Lawrence, H., Li, J., Musco, C., Musco, C.: Low-rank Toeplitz matrix estimation via random ultra-sparse rulers. In: ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 4796–4800. IEEE (2020)
40. Levina, E., Vershynin, R.: Partial estimation of covariance matrices. Probab. Theory Related Fields **153**(3-4), 405–419 (2012)
41. Li, Y., Tao, C., Seco-Granados, G., Mezghani, A., Swindlehurst, A.L., Liu, L.: Channel estimation and performance analysis of one-bit massive MIMO systems. IEEE Trans. Signal Process. **65**(15), 4075–4089 (2017)
42. Liu, L., Hawkins, D.M., Ghosh, S., Young, S.S.: Robust singular value decomposition analysis of microarray data. Proc. Nat. Acad. Sci. **100**(23), 13167–13172 (2003)
43. Lounici, K.: High-dimensional covariance matrix estimation with missing observations. Bernoulli **20**(3), 1029–1058 (2014)
44. Lu, L., Li, G.Y., Swindlehurst, A.L., Ashikhmin, A., Zhang, R.: An overview of massive MIMO: Benefits and challenges. IEEE J. Selected Topics Signal Process. **8**(5), 742–758 (2014)
45. Marzetta, T.L., Ngo, H.Q.: Fundamentals of massive MIMO. Cambridge University Press, Cambridge (2016)
46. Mendelson, S., Zhivotovskiy, N.: Robust covariance estimation under L4-L2 norm equivalence. Ann. Stat. **48**(3), 1648–1664 (2020)
47. Minsker, S.: Sub-Gaussian estimators of the mean of a random matrix with heavy-tailed entries. Ann. Stat. **46**(6A), 2871–2903 (2018)
48. Minsker, S., Wei, X.: Robust modifications of U-statistics and applications to covariance estimation problems. Bernoulli **26**(1), 694–727 (2020)
49. Nemirovskij, A.S., Yudin, D.B.: Problem complexity and method efficiency in optimization (1983)
50. Paulraj, A.J., Gore, D.A., Nabar, R.U., Bolcskei, H.: An overview of MIMO communications-a key to gigabit wireless. Proc. IEEE **92**(2), 198–218 (2004)

51. Recht, B., Fazel, M., Parrilo, P.A.: Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. SIAM Rev. **52**(3), 471–501 (2010)
52. Roberts, L.: Picture coding using pseudo-random noise. IRE Trans. Inform. Theory **8**(2), 145–154 (1962)
53. Romero, D., Arianandа, D.D., Tian, Z., Leus, G.: Compressive covariance sensing: Structure-based compressive sensing beyond sparsity. IEEE Signal Process. Mag. **33**(1), 78–93 (2016)
54. Roth, K., Munir, J., Mezghani, A., Nossek, J.A.: Covariance based signal parameter estimation of coarse quantized signals. In: 2015 IEEE International Conference on Digital Signal Processing (DSP), pp. 19–23. IEEE (2015)
55. Schreier, R., Temes, G.C., Norsworthy, S.R.: Delta-Sigma Data Converters: Theory, Design, and Simulation. IEEE Press (1996)
56. Snyder, D.L., O'Sullivan, J.A., Miller, M.I.: The use of maximum likelihood estimation for forming images of diffuse radar targets from delay-doppler data. IEEE Trans. Inform. Theory **35**(3), 536–548 (1989)
57. Stoica, P., Babu, P., Li, J.: SPICE: a sparse covariance-based estimation method for array processing. IEEE Trans. Signal Process. **59**(2), 629–638 (2011)
58. Stoica, P., Moses, R.L.: Spectral analysis of signals (2005)
59. Tse, D., Viswanath, P.: Fundamentals of Wireless Communication. Cambridge University Press, Cambridge (2005)
60. Van Vleck, J.H., Middleton, D.: The spectrum of clipped noise. Proc. IEEE **54**(1), 2–19 (1966)
61. Vershynin, R.: High-Dimensional Probability: An Introduction with Applications in Data Science, vol. 47. Cambridge University Press, Cambridge (2018)

# Chapter 4
# Sparse Deterministic and Stochastic Channels: Identification of Spreading Functions and Covariances


**Alihan Kaplan, Dae Gwan Lee, Götz E. Pfander, and Volker Pohl**

## 4.1 Motivation and Introduction

Many physical and technical systems in science and engineering are well described by linear systems. In practical applications, it is important to identify the parameters describing the linear system at hand. If the system is time-invariant, the channel is uniquely determined from the impulse response of the channel. For time-varying systems, the identification problem becomes much more challenging and it is even not obvious whether a given time-invariant system can actually be identified or not. It is necessary to identify time-varying systems in many areas of science and engineering and especially in communications, control, and system theory. For concreteness, we will focus the discussions in this chapter on the identification of time-varying communication channels.

The transmission of a continuous-time signal $x$ over a dispersive and time-varying channel $H$ can be described formally by the relation

$$(Hx)(t) = \iint_{\mathbb{R}^2} \eta_H(\tau, \nu)\, x(t-\tau) e^{2\pi i (t-\tau)\nu} d\tau d\nu, \qquad t \in \mathbb{R}. \tag{4.1}$$

A. Kaplan · V. Pohl
Technical University of Munich, Institute of Theoretical Information Technology, Munich, Germany
e-mail: alihan.kaplan@tum.de; volker.pohl@tum.de

D. G. Lee (✉) · G. E. Pfander
Katholische Universität Eichstätt-Ingolstadt, Mathematisch-Geographische Fakultät, Eichstätt, Germany
e-mail: pfander@ku.de



G. Kutyniok et al. (eds.), *Compressed Sensing in Information Processing*, Applied and Numerical Harmonic Analysis, https://doi.org/10.1007/978-3-031-09745-4_4

The so-called *(delay-Doppler) spreading function* $\eta_H : \mathbb{R}\times\widehat{\mathbb{R}} \to \mathbb{C}$ completely describes the channel $H$ [2, 9, 19, 34], and identifying a channel of the form (4.1) is then equivalent to the problem of determining the spreading function $\eta_H$ from the channel response $Hg$ to a known test signal $g$. In particular, one asks which spreading functions can be identified by such a probing scheme and how one has to design the test signal $g$ in order to identify the channel. This channel identification problem has a long history starting in the 1960s with groundbreaking works of Kailath and Bello [2, 9] that lead to fundamental results in [12, 23]; it remains an active field of research to date. In recent years, for example, the channel identification problem has been considered for multiple-input multiple-output (MIMO) channels [15, 20], for sparse [7, 24] and stochastic channels [26–28], or for channels with satisfying linear side constraints [17]. For an overview on the fascinating history of this problem, ranging back to the cold war, we refer to the overview article [35].

It is known [7, 22] that the identification problem for continuous-time channels (4.1) can be solved by reducing it to an identification problem for time-discrete, finite-dimensional channels of the form

$$\boldsymbol{H}\boldsymbol{x} = \sum_{k=0}^{L-1}\sum_{\ell=0}^{L-1} \eta(k,\ell)\boldsymbol{M}^{\ell}\boldsymbol{T}^{k}\boldsymbol{x} \tag{4.2}$$

for a signal $\boldsymbol{x} \in \mathbb{C}^L$, with *spreading coefficients* $\eta(k,\ell)$, and where $\boldsymbol{M}$ and $\boldsymbol{T}$ stand for the modulation and translation operators on $\mathbb{C}^L$, respectively. See Definition 4.1 for details. Because of this close relation between (4.1) and (4.2), we will discuss mainly the time-discrete model.

This chapter reviews some recent result concerning the channel identification problem paying particular attention to the question whether linear side constraints will help for identifying the channel. With respect to the channel model (4.2), we may assume that the spreading coefficients $\eta_H$ satisfy one or more equations of the form

$$\sum_{k,\ell=0}^{L-1} a_n(k,\ell)\,\eta_H(k,\ell) = b_n, \qquad n = 1, 2, \ldots, N,$$

with known coefficients $a_n(k,\ell)$ and $b_n$. Does the knowledge of such side constraints help to identify the channel? In general, the answer depends (of course) on the coefficients $a_n(k,\ell)$ and $b_n$, but, as we will see, knowing such side constraints will enable us to identify channels which cannot be identified without these side constraints.

The parameters of a communication channel, i.e., the spreading coefficients $\eta_H$, can change very rapidly since they depend strongly on the position of the transmitter and receiver as well as on the environment (i.e., scatterer such as landscape, buildings, cars, etc.). If the transmitter and/or the receiver move, these scattering

coefficients will change. While formally, such channels can be described by a deterministic spreading function, it is frequently beneficial to model the coefficients $\eta_H(k, \ell)$ as random variables. In addition, random spreading coefficients are used to describe an ensemble of communication channels, for example, described by a particular application or setting.

In this case, channel identification does not aim to identify the spreading coefficients $\eta_H(\tau, \ell)$ itself, but their statistical properties and especially their autocorrelation

$$R_{\eta_H}(k, \ell, k', \ell') = \mathbb{E}\left\{\eta_H(k, \ell)\,\overline{\eta_H(k', \ell')}\right\},$$

where $\mathbb{E}\{\cdot\}$ denotes the expectation. For such *stochastic channels*, one can ask similar questions as in the deterministic case. Which channels are identifiable? How should we choose the identifier? Do linear side constraints satisfied by the autocorrelation $R_{\eta_H}$ improve the ability to identify the channel? In this chapter, we focus again on the last question, and we will show that similarly as in the deterministic case, known side constraints on the autocorrelation of the scattering coefficients are usually beneficial for the identification of stochastic channels.

This chapter is structured as follows. The finite-dimensional model for deterministic and stochastic channels is discussed in detail in Sect. 4.2. Section 4.3 is devoted to deterministic channels. Along with a review of known results for identification of deterministic channels, we discuss the problem of utilizing some known linear constraints between and within subchannels and also discuss the application of transmitting messages with unidentified channels. Section 4.4 is devoted to the problem of identifying stochastic channels, where we also consider the situation of knowing some linear relations between the covariance entries.

## 4.2 Channel Identification and Estimation

While continuous-time channels of the form (4.1) are useful in modeling a larger class of dispersive operators, finite-dimensional channels of the form (4.2) are preferred for applications in communications engineering. In this section, we review the finite-dimensional model for deterministic and stochastic channels and in particular discuss the respective channel identification problem. Thereby, we consider two kinds of communication channels: (1) *deterministic channels* that are characterized by a set of fixed (deterministic) spreading coefficients and (2) *stochastic channels* whose associated spreading coefficients are random variables. We will discuss these channels in detail after recalling some basic notions in the time–frequency analysis.

### 4.2.1 Time–Frequency Analysis in Finite Dimensions

Vectors in $\mathbb{C}^L$ will be denoted by boldface letters and their entries will be indexed by the cyclic group $\mathbb{Z}_L = \{0, 1, \ldots, L-1\}$, that is, we write a vector in $\mathbb{C}^L$ as $\boldsymbol{x} = (x_0, x_1, \ldots, x_{L-1})^{\mathbb{T}} \in \mathbb{C}^L$, where $(\cdot)^{\mathbb{T}}$ denotes the transpose of a vector.

**Definition 4.1** For $L \in \mathbb{N}$, cyclic translation and modulation on $\mathbb{C}^L$ are defined, respectively, as

$$\boldsymbol{T} : \mathbb{C}^L \to \mathbb{C}^L, \quad (x_0, x_1, \ldots, x_{L-1}) \mapsto (x_{L-1}, x_0, \ldots, x_{L-2}) \qquad \text{and}$$
$$\boldsymbol{M} : \mathbb{C}^L \to \mathbb{C}^L, \quad (x_0, x_1, \ldots, x_{L-1}) \mapsto (\omega^0 x_0, \omega^1 x_1 \ldots, \omega^{L-1} x_{L-1}),$$

where $\omega = e^{2\pi i/L}$. We define the *time–frequency shift* operator $\pi(k, \ell) = \boldsymbol{M}^\ell \boldsymbol{T}^k$ for $(k, l) \in \mathbb{Z}_L \times \mathbb{Z}_L$. Moreover, the *short-time Fourier transform* of $\boldsymbol{x} \in \mathbb{C}^L$ with respect to a window $\boldsymbol{c} \in \mathbb{C}^L$ is defined as $V_{\boldsymbol{c}}\boldsymbol{x}(k, \ell) = \langle \boldsymbol{x}, \pi(k, \ell)\boldsymbol{c} \rangle$ for $(k, \ell) \in \mathbb{Z}_L \times \mathbb{Z}_L$.

The non-commutativity of $\boldsymbol{T}$ and $\boldsymbol{M}$ plays a crucial role in time–frequency analysis. The (non-)commutation relation is given by $\boldsymbol{M}^\ell \boldsymbol{T}^k = \omega^{k\ell}\, \boldsymbol{T}^k \boldsymbol{M}^\ell$ for $k, \ell \in \mathbb{Z}_L$.

**Definition 4.2** The *Gabor matrix* generated by a window $\boldsymbol{c} \in \mathbb{C}^L$ is the $L \times L^2$ matrix

$$\begin{aligned} \boldsymbol{G}(\boldsymbol{c}) = \big[\, \boldsymbol{c},\, \boldsymbol{M}\boldsymbol{c},\, \ldots,\, \boldsymbol{M}^{L-1}\boldsymbol{c} \;\big|\; \boldsymbol{T}\boldsymbol{c},\, \boldsymbol{M}\boldsymbol{T}\boldsymbol{c},\, \ldots,\, \boldsymbol{M}^{L-1}\boldsymbol{T}\boldsymbol{c} \;\big| \\ \ldots \;\big|\; \boldsymbol{T}^{L-1}\boldsymbol{c},\, \boldsymbol{M}\boldsymbol{T}^{L-1}\boldsymbol{c},\, \ldots,\, \boldsymbol{M}^{L-1}\boldsymbol{T}^{L-1}\boldsymbol{c} \,\big]. \end{aligned} \tag{4.3}$$

It is easy to compute that the rows of $\boldsymbol{G}(\boldsymbol{c})$ are mutually orthogonal, and in fact, it holds $\boldsymbol{G}(\boldsymbol{c})\, \boldsymbol{G}(\boldsymbol{c})^* = L\|\boldsymbol{c}\|^2 I_L$, which corresponds to the fact that for $\boldsymbol{c} \in \mathbb{C}^L \backslash \{0\}$, the set $\{\boldsymbol{M}^\ell \boldsymbol{T}^k \boldsymbol{c} : (k, \ell) \in \mathbb{Z}_L \times \mathbb{Z}_L\}$ of all time–frequency shifts of $\boldsymbol{c}$ forms a tight $\mathbb{C}^L$-frame with frame bound $L\|\boldsymbol{c}\|^2$ (see [13, Proposition 2]).

We will often deal with matrices that have more columns than rows. For such matrices, the degree of linear independence between columns is quantized by the so-called spark.

**Definition 4.3** For a matrix $\boldsymbol{A} \in \mathbb{C}^{M \times N}$ with $M \leq N$, the *spark* of $\boldsymbol{A}$ is the cardinality of the smallest linearly dependent subset of columns in $\boldsymbol{A}$, that is,

$$\operatorname{spark}(\boldsymbol{A}) = \min \big\{ \|\boldsymbol{z}\|_0 : \boldsymbol{A}\boldsymbol{z} = 0,\, \boldsymbol{z} \in \mathbb{C}^N \backslash \{0\} \big\},$$

where $\|\boldsymbol{z}\|_0 = |\{n \in \mathbb{Z}_N : z_n \neq 0\}|$ denotes the number of nonzero entries in $\boldsymbol{z}$. We say that $\boldsymbol{A}$ has *full spark* if $\operatorname{spark}(\boldsymbol{A}) = M + 1$, that is, if every $M$ columns of $\boldsymbol{A}$ are linearly independent.

In particular, the Gabor matrix $\boldsymbol{G}(\boldsymbol{c}) \in \mathbb{C}^{L \times L^2}$ has full spark for some windows $\boldsymbol{c} \in \mathbb{C}^L$:

**Proposition 4.1 (Theorem 1 in [13] for $L$ Prime and [18] for General $L \in \mathbb{N}$)** *Given $L \in \mathbb{N}$, the matrix $\boldsymbol{G}(\boldsymbol{c})$ has full spark for almost every $\boldsymbol{c} \in \mathbb{C}^L$. Moreover, the set of all such $\boldsymbol{c}$ is a dense open subset $U_L$ of $\mathbb{C}^L$, whose complement set is a finite union of manifolds with zero Lebesgue measure.*

This result is quite involved, it took 10 years to resolve the case for composite $L$. Quite useful, and much simpler, is the following characterization.

**Proposition 4.2 (Theorem 2 in [13])** *Let $L \in \mathbb{N}$ and $\boldsymbol{c} \in \mathbb{C}^L \backslash \{0\}$. The matrix $\boldsymbol{G}(\boldsymbol{c})$ has full spark if and only if for each $\boldsymbol{x} \in \mathbb{C}^L \backslash \{0\}$, the short-time Fourier transform $V_{\boldsymbol{c}}\boldsymbol{x} \in \mathbb{C}^{L^2}$ has at most $L-1$ zero entries.*

Proposition 4.2 implies that the set $U_L$ in Proposition 4.1 is given by

$$U_L = \{\boldsymbol{c} \in \mathbb{C}^L \,:\, V_{\boldsymbol{c}}\boldsymbol{x} \text{ has at most } L-1 \text{ zero entries for every } \boldsymbol{x} \in \mathbb{C}^L \backslash \{0\}\}.$$

We will use this set $U_L$ later when discussing the identification of SISO channels. For the case of MIMO channels, we require the following generalization of Proposition 4.1.

**Proposition 4.3 (Theorem 7 in [17])** *For every $L, N \in \mathbb{N}$, there exists a dense open subset $U_{L,N} \subset (\mathbb{C}^L)^N$ with full measure such that the matrix*

$$\boldsymbol{G}(\boldsymbol{c}^{(1)}, \ldots, \boldsymbol{c}^{(N)}) := \left[\, \boldsymbol{G}(\boldsymbol{c}^{(1)}) \;\middle|\; \boldsymbol{G}(\boldsymbol{c}^{(2)}) \;\middle|\; \cdots \;\middle|\; \boldsymbol{G}(\boldsymbol{c}^{(N)}) \,\right] \in \mathbb{C}^{L \times NL^2} \tag{4.4}$$

*has full spark for $(\boldsymbol{c}^{(1)}, \ldots, \boldsymbol{c}^{(N)}) \in U_{L,N}$.*

### 4.2.2 Deterministic Channels

A SISO communication channel is modeled as a linear map $\boldsymbol{H} : \mathbb{C}^L \to \mathbb{C}^L$. It is well known that the set of all time–frequency shifts $\{\pi(k,\ell)\}_{(k,\ell)\in\mathbb{Z}_L\times\mathbb{Z}_L}$ is a basis for the space $\mathcal{L}(\mathbb{C}^L, \mathbb{C}^L)$ of all linear operators on $\mathbb{C}^L$. Therefore, every $\boldsymbol{H} \in \mathcal{L}(\mathbb{C}^L, \mathbb{C}^L)$ can be written as

$$\boldsymbol{H} = \sum_{k,\ell=0}^{L-1} \eta(k,\ell)\, \boldsymbol{M}^{\ell}\boldsymbol{T}^{k} \tag{4.5}$$

with unique coefficients $\boldsymbol{\eta} = \left\{\eta(k,\ell)\right\}_{(k,\ell)\in\mathbb{Z}_L\times\mathbb{Z}_L}$ called the *spreading coefficients* of $\boldsymbol{H}$ which encode all the characteristics of $\boldsymbol{H}$. We will often write $\boldsymbol{\eta} = \boldsymbol{\eta}_{\boldsymbol{H}} = \left\{\eta_H(k,\ell)\right\}_{(k,\ell)\in\mathbb{Z}_L\times\mathbb{Z}_L}$ when it is necessary to specify the dependence of $\boldsymbol{\eta}$ on $\boldsymbol{H}$. In the context of communications, each coefficient $\eta(k,\ell)$ can be understood as a gain factor associated with a transmission path with time delay $k$ (due to the traveling distance) and frequency shift $\ell$ (due to the Doppler effect). Note that for an input

signal $\boldsymbol{x} \in \mathbb{C}^L$, we have

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} = \sum_{k,\ell=0}^{L-1} \eta(k,\ell)\, \boldsymbol{M}^\ell \boldsymbol{T}^k \boldsymbol{x} = \boldsymbol{G}(\boldsymbol{x})\, \boldsymbol{\eta}\,, \tag{4.6}$$

which relates the channel output $\boldsymbol{H}\boldsymbol{x}$ to the Gabor matrix $\boldsymbol{G}(\boldsymbol{x})$ discussed in Sect. 4.2.1.

It is straightforward to extend the above SISO model to MIMO channels. A MIMO communication channel with $N \in \mathbb{N}$ inputs and $M \in \mathbb{N}$ outputs is described by a linear map $\boldsymbol{H} : (\mathbb{C}^L)^N \to (\mathbb{C}^L)^M$ with $MN$ subchannels $\boldsymbol{H}_{mn} : \mathbb{C}^L \to \mathbb{C}^L$, $m = 1, \ldots, M$, $n = 1, \ldots, N$, where each $\boldsymbol{H}_{mn}$ is of the form (4.5) and describes the transmission associated with the $n$-th input and the $m$-th output. For an input signal $\boldsymbol{x} = \{\boldsymbol{x}^{(n)}\}_{n=1}^N \in (\mathbb{C}^L)^N$, the $m$-th output of $\boldsymbol{H}$ is then given by

$$\boldsymbol{y}_m = \boldsymbol{H}_m\, \boldsymbol{x} = \sum_{n=1}^{N} \boldsymbol{H}_{m,n}\, \boldsymbol{x}^{(n)}, \qquad m = 1, 2, \ldots, M. \tag{4.7}$$

Note that $\boldsymbol{H}$ can represented by the $M \times N$ block matrix

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{H}_{1,1} & \cdots & \boldsymbol{H}_{1,N} \\ \vdots & & \vdots \\ \boldsymbol{H}_{M,1} & \cdots & \boldsymbol{H}_{M,N}, \end{bmatrix} \tag{4.8}$$

where each $\boldsymbol{H}_{m,n} \in \mathbb{C}^{L\times L}$ is the matrix representation of $\boldsymbol{H}_{m,n} \in \mathcal{L}(\mathbb{C}^L, \mathbb{C}^L)$. The spreading coefficients of each subchannel $\boldsymbol{H}_{m,n}$ will be denoted by $\boldsymbol{\eta}_{m,n} = [\eta_{m,n}(k,\ell)]_{k,\ell=0}^{L-1} \in \mathbb{C}^{L^2}$, and their collection will be denoted by $\boldsymbol{\eta} = \{\boldsymbol{\eta}_{m,n}\}_{m=1\,n=1}^{M\quad N}$. Substituting the expression (4.6) for individual SISO subchannels into (4.7) yields

$$\begin{aligned} \boldsymbol{y}_m = \boldsymbol{H}_m\, \boldsymbol{x} = \sum_{n=1}^{N} \boldsymbol{G}(\boldsymbol{x}^{(n)})\, \boldsymbol{\eta}_{m,n} &= \left[\, \boldsymbol{G}(\boldsymbol{x}^{(1)}) \mid \boldsymbol{G}(\boldsymbol{x}^{(2)}) \mid \cdots \mid \boldsymbol{G}(\boldsymbol{x}^{(N)}) \,\right] \boldsymbol{\eta}_m \\ &= \boldsymbol{G}\big(\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N)}\big)\, \boldsymbol{\eta}_m \end{aligned}$$

with $\boldsymbol{\eta}_m = \big(\boldsymbol{\eta}_{m,1}, \ldots, \boldsymbol{\eta}_{m,N}\big) \in (\mathbb{C}^{L^2})^N$. This expression relates the signal at the $m$-th output with the concatenated Gabor matrix $\boldsymbol{G}\big(\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N)}\big) \in \mathbb{C}^{L\times NL^2}$ discussed in Proposition 4.3.

Note that both SISO and MIMO channels are essentially linear maps from $\mathbb{C}^{L_1}$ to $\mathbb{C}^{L_2}$. The case of SISO channels corresponds to $L_1 = L_2 = L$, and the case of $N$-input $M$-output channels corresponds to $L_1 = NL$ and $L_2 = ML$.

**Definition 4.4** A class of operators $\mathcal{H} \subset \mathcal{L}(\mathbb{C}^{L_1}, \mathbb{C}^{L_2})$ is *identifiable* if there exists a vector $\boldsymbol{c} \in \mathbb{C}^{L_1}$ such that the map $\boldsymbol{\Phi}_{\boldsymbol{c}} : \mathcal{H} \to \mathbb{C}^{L_2}, \quad \boldsymbol{H} \mapsto \boldsymbol{H}\boldsymbol{c}$ is injective. Such a vector $\boldsymbol{c}$ is called an *identifier* for $\mathcal{H}$.

Note that if $\mathcal{H}$ is an identifiable linear space of $\mathcal{L}(\mathbb{C}^{L_1}, \mathbb{C}^{L_2})$, then it is necessarily of dimension at most $L_2$.

### 4.2.3 Stochastic Channels

We now consider channels that vary rapidly and unpredictably over time and channel ensembles, both of which are modelled as so-called *stochastic channels*. Such channels can be obtained by replacing the spreading coefficients in the deterministic channel model with some random variables. Adapting the expression (4.6) for deterministic SISO channels, we model stochastic SISO channels as

$$\begin{aligned} \boldsymbol{y}(\xi) = \boldsymbol{H}\boldsymbol{x}(\xi) &= \sum_{(k,\ell)\in\mathbb{Z}_L\times Z_L} \eta_H(k,\ell;\xi)\, \boldsymbol{M}^{\ell}\boldsymbol{T}^{k}\boldsymbol{x} \\ &= \boldsymbol{G}(\boldsymbol{x})\, \boldsymbol{\eta}_{\boldsymbol{H}}(\xi) \qquad \text{for } \boldsymbol{x} \in \mathbb{C}^{L}, \end{aligned} \tag{4.9}$$

where each $\eta_H(k,\ell;\cdot)$ is a complex-valued random variable with zero mean and finite second moments. We denote the space of all such random variables by $\mathscr{V}$, so that the family $\boldsymbol{\eta}_{\boldsymbol{H}} = \{\eta_H(k,\ell;\cdot)\}_{(k,\ell)\in\mathbb{Z}_L\times\mathbb{Z}_L}$ of $L^2$ random variables can be understood as an element of $\mathscr{V}^{L^2}$.

For a (deterministic) input signal $\boldsymbol{x}$ in (4.9), both $\boldsymbol{\eta}_H(\xi)$ and the output $\boldsymbol{y}(\xi)$ are *zero mean* random vectors. The second moments of these vectors, i.e., the corresponding *covariance* matrices, are then given by

$$\begin{aligned} R_{\eta_H}(\lambda,\lambda') &:= \mathbb{E}\{\eta_H(\lambda;\xi)\,\overline{\eta_H(\lambda';\xi)}\} \quad \text{for } \lambda,\lambda' \in \mathbb{Z}_L\times\mathbb{Z}_L\,, \\ R_y(m,m') &:= \mathbb{E}\{y_m(\xi)\,\overline{y_{m'}(\xi)}\} \quad \text{for } m,m' \in \mathbb{Z}_L\,, \end{aligned} \tag{4.10}$$

respectively, where $y_m(\xi)$ denotes the $m$-th coordinate entry of $\boldsymbol{y}(\xi)$, that is, $\boldsymbol{y}(\xi) = \big(y_1(\xi),\ldots,y_L(\xi)\big)^{\mathbb{T}}$. It is apparent from these definitions that both matrices $\boldsymbol{R}_{\boldsymbol{\eta}_H} \in \mathbb{C}^{L^2\times L^2}$ and $\boldsymbol{R}_{\boldsymbol{y}} \in \mathbb{C}^{L\times L}$ are Hermitian, that is,

$$R_{\eta_H}(\lambda,\lambda') = \overline{R_{\boldsymbol{\eta}_H}(\lambda',\lambda)} \qquad \text{and} \qquad R_y(m,m') = \overline{R_{\boldsymbol{y}}(m,m')}\,, \tag{4.11}$$

respectively. Moreover, using (4.9), we see that these matrices are related by

$$\boldsymbol{R}_{\boldsymbol{y}} = \mathbb{E}\{\boldsymbol{y}(\xi)\,\boldsymbol{y}(\xi)^*\} = \mathbb{E}\{\boldsymbol{G}(\boldsymbol{x})\,\boldsymbol{\eta}_H(\xi)\,\boldsymbol{\eta}_H(\xi)^*\,\boldsymbol{G}(\boldsymbol{x})^*\} = \boldsymbol{G}(\boldsymbol{x})\,\boldsymbol{R}_{\boldsymbol{\eta}_H}\,\boldsymbol{G}(\boldsymbol{x})^*,$$

which can be vectorized to obtain

$$\operatorname{vec} \boldsymbol{R}_{\boldsymbol{y}} = \left(\overline{\boldsymbol{G}(\boldsymbol{x})} \otimes \boldsymbol{G}(\boldsymbol{x})\right) \operatorname{vec} \boldsymbol{R}_{\boldsymbol{\eta}_H} . \tag{4.12}$$

Here, the matrices $\boldsymbol{R}_{\boldsymbol{y}}$ and $\boldsymbol{R}_{\boldsymbol{\eta}_H}$ are of dimension $L \times L$ and $L^2 \times L^2$, respectively, and the vectorization operator $\operatorname{vec} : \mathbb{C}^{N\times N} \to \mathbb{C}^{N^2}$ for $N \in \mathbb{N}$ is defined as

$$\operatorname{vec} \boldsymbol{X}(k) = \boldsymbol{X}(k \bmod N, \lfloor k/N \rfloor) \quad \text{for } k = 0, 1, \ldots, N^2 - 1, \tag{4.13}$$

that is, $\operatorname{vec} \boldsymbol{X}$ is the vector formed by stacking the columns of $\boldsymbol{X} \in \mathbb{C}^{N\times N}$.

Stochastic MIMO channels are modeled as in (4.8) with each subchannel $\boldsymbol{H}_{mn}$ being a stochastic SISO channel of the form (4.9). Note that as for the deterministic channels, both SISO and MIMO stochastic channels are essentially linear maps from $\mathbb{C}^{L_1}$ to $\mathscr{V}^{L_2}$ for some $L_1, L_2 \in \mathbb{N}$. Indeed, a SISO channel corresponds to $L_1 = L_2 = L$, and an $N$-input $M$-output channel corresponds to $L_1 = NL$ and $L_2 = ML$.

**Definition 4.5** A class of stochastic operators $\mathcal{H} \subset \mathcal{L}(\mathbb{C}^{L_1}, \mathscr{V}^{L_2})$ is *identifiable (up to second-order statistics)*, if there exists a vector $\boldsymbol{c} \in \mathbb{C}^{L_1}$ such that operators in $\mathcal{H}$ with different covariances $\boldsymbol{R}_{\boldsymbol{\eta}_H} = \mathbb{E}\left\{\operatorname{vec} \boldsymbol{\eta}_{\boldsymbol{H}} (\operatorname{vec} \boldsymbol{\eta}_{\boldsymbol{H}})^*\right\}$ yield different output covariances $\boldsymbol{R}_{\boldsymbol{Hc}} = \mathbb{E}\{\boldsymbol{Hc}(\boldsymbol{Hc})^*\}$; more formally, if there exists a vector $\boldsymbol{c} \in \mathbb{C}^{L_1}$ such that the map[1]

$$\boldsymbol{\Psi}_{\boldsymbol{c},\mathcal{H}} : (\mathcal{H}/\sim) \to \mathbb{C}^{L_2\times L_2}, \quad [\boldsymbol{H}] \mapsto \boldsymbol{R}_{\boldsymbol{Hc}}$$

is injective, where $\mathcal{H}/\sim$ denotes the set of all equivalence classes of $\mathcal{H}$ by the equivalence relation $\sim$ defined as $\boldsymbol{H} \sim \boldsymbol{H}'$ if and only if $\boldsymbol{R}_{\boldsymbol{\eta}_H} = \boldsymbol{R}_{\boldsymbol{\eta}_{H'}}$. Such a vector $\boldsymbol{c}$ is called an *identifier* for $\mathcal{H}$.

It is easily seen that the full class of stochastic SISO channels $\boldsymbol{H}$ of the form (4.9) is not identifiable. Indeed, the map $\boldsymbol{\Psi}_{\boldsymbol{c},\mathcal{H}} : (\mathcal{H}/\sim) \to \mathbb{C}^{L\times L}$ in this case is essentially described by Eq. (4.12) with $\boldsymbol{x} = \boldsymbol{c}$, which is an underdetermined linear system associated with the $L^2 \times L^4$ matrix $\overline{\boldsymbol{G}(\boldsymbol{c})} \otimes \boldsymbol{G}(\boldsymbol{c})$, and hence $\boldsymbol{\Psi}_{\boldsymbol{c},\mathcal{H}}$ cannot be injective. Considering the degrees of freedom, one would need to by far restrict the class of stochastic SISO channels to achieve the identifiability.

*Remark 4.1 (Transition to Continuous-Time Setting)* It should be noted that all results established in the finite-dimensional (discrete-time) setting can be carried over to the continuous-time setting in a straightforward way. For more details, we refer to [17, Sect. 5] and also [15, 25].

[1] The map $\boldsymbol{\Psi}_{\boldsymbol{c}}$ is well defined due to the relation (4.12).

## 4.3 Results in Deterministic Setting

This section is dedicated to deterministic channels. In Sect. 4.3.1, we give a short review of known results on identification of SISO/MIMO channels. In Sect. 4.3.2, we discuss how to utilize some known linear constraints between and within subchannels. Section 4.3.3 addresses the application of transmitting messages through channels that are not identified in advance.

### *4.3.1 Classical Results on Channel Identification*

#### 4.3.1.1 Identification of SISO Channels

Recalling that every $\boldsymbol{H} \in \mathcal{L}(\mathbb{C}^L, \mathbb{C}^L)$ can be expressed in the form of (4.5) with unique spreading coefficients $\boldsymbol{\eta_H} = \big\{\eta_H(k,\ell)\big\}_{(k,\ell)\in\mathbb{Z}_L\times\mathbb{Z}_L}$ supported in $\mathbb{Z}_L\times\mathbb{Z}_L$, we define the following class of operators with restricted spreading support $\Lambda \subset \mathbb{Z}_L\times\mathbb{Z}_L$.

**Definition 4.6** For $\Lambda \subset \mathbb{Z}_L\times\mathbb{Z}_L$, the *single-input single-output (SISO) operator Paley–Wiener space* on $\Lambda$ is defined as

$$OPW(\Lambda) = \operatorname{span}\big\{\boldsymbol{M}^\ell\boldsymbol{T}^k : (k,\ell)\in\Lambda\big\} = \big\{\boldsymbol{H}\in\mathcal{L}(\mathbb{C}^L,\mathbb{C}^L) : \operatorname{supp}\boldsymbol{\eta_H}\subset\Lambda\big\}.$$

For instance, the class of operators which consists of linear combinations of translations $\boldsymbol{T}^k$, $k = 0, 1, \ldots, L-1$, corresponds to the space $OPW(\Lambda)$ with $\Lambda = \mathbb{Z}_L\times\{0\}$.

According to Definition 4.4, the space $OPW(\Lambda)$ is identifiable if and only if there is a vector $\boldsymbol{c}\in\mathbb{C}^L$ such that for $\boldsymbol{H}\in OPW(\Lambda)$ the equation

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{c} = \sum_{(k,\ell)\in\Lambda} \eta_H(k,\ell)\,\boldsymbol{M}^\ell\boldsymbol{T}^k\boldsymbol{c} \tag{4.14}$$

is uniquely solvable in $\boldsymbol{\eta}_{\boldsymbol{H},\Lambda} = \{\eta_H(k,\ell)\}_{(k,\ell)\in\Lambda} \in \mathbb{C}^\Lambda$. With $\boldsymbol{G}(\boldsymbol{c}) \in \mathbb{C}^{L\times L^2}$ denoting the Gabor matrix generated by $\boldsymbol{c} \in \mathbb{C}^L$ (see Sect. 4.2.1), one can rewrite (4.14) as

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{c} = \boldsymbol{G}(\boldsymbol{c})|_\Lambda\,\boldsymbol{\eta}|_\Lambda.$$

This implies that $OPW(\Lambda)$ is identifiable if and only if the matrix $\boldsymbol{G}(\boldsymbol{c})|_\Lambda$ has linearly independent columns. Note that by Proposition 4.1, there exists a vector $\boldsymbol{c}\in\mathbb{C}^L$ such that $\boldsymbol{G}(\boldsymbol{c})$ has full spark, that is, every $L$ columns of $\boldsymbol{G}(\boldsymbol{c})$ are linearly independent. Consequently, we have the following characterization for identifiability of $OPW(\Lambda)$ given only in terms of the size of $\Lambda$.

**Corollary 4.1 ([13, 18])** *For $\Lambda \subset \mathbb{Z}_L \times \mathbb{Z}_L$, the space $OPW(\Lambda)$ is identifiable if and only if $|\Lambda| \leq L$.*

#### 4.3.1.2 Identification of MIMO Channels

As discussed in Sect. 4.2.2, deterministic channels with $N$-inputs and $M$-outputs are modeled as linear maps from $(\mathbb{C}^L)^N$ to $(\mathbb{C}^L)^M$. Recall from Definition 4.4 that a class of operators $\mathcal{H} \subset \mathcal{L}\left((\mathbb{C}^L)^N, (\mathbb{C}^L)^M\right)$ is identifiable if and only if there exists a vector $\boldsymbol{c} \in (\mathbb{C}^L)^N$ such that the map $\mathcal{H} \to (\mathbb{C}^L)^M$, $\boldsymbol{H} \mapsto \boldsymbol{Hc}$ is injective.

**Definition 4.7** For $\boldsymbol{\Lambda} = [\Lambda_{m,n}]_{m=1}^{M}{}_{n=1}^{N}$ with $\Lambda_{m,n} \subset \mathbb{Z}_L \times \mathbb{Z}_L$, the *MIMO operator Paley–Wiener space* on $\boldsymbol{\Lambda}$ is defined as

$$OPW(\boldsymbol{\Lambda}) = \left\{\boldsymbol{H} : \boldsymbol{H}_{mn} \in OPW(\Lambda_{m,n}),\ m = 1, \ldots, M,\ n = 1, \ldots, N\right\}.$$

The space $OPW(\boldsymbol{\Lambda})$ is identifiable if and only if there exists a vector $\boldsymbol{c} = (\boldsymbol{c}^{(1)}, \ldots, \boldsymbol{c}^{(N)}) \in (\mathbb{C}^L)^N$ such that the map $\boldsymbol{H} \mapsto \boldsymbol{Hc}$ is injective on $OPW(\boldsymbol{\Lambda})$. Using the full sparkness of the concatenated Gabor matrices in Proposition 4.3, we obtain the following characterization for identifiability of $OPW(\boldsymbol{\Lambda})$.

**Corollary 4.2 ([20])** *For $\boldsymbol{\Lambda} = [\Lambda_{m,n}]_{m=1}^{M}{}_{n=1}^{N}$ with $\Lambda_{m,n} \subset \mathbb{Z}_L \times \mathbb{Z}_L$, the space $OPW(\boldsymbol{\Lambda})$ is identifiable if and only if $\sum_{n=1}^{N} |\Lambda_{m,n}| \leq L$ for all $m = 1, \ldots, M$.*

Corollary 4.2 implies that $OPW(\boldsymbol{\Lambda})$ is identifiable if and only if for each $m$ the space $OPW(\boldsymbol{\Lambda}_m)$ with $\boldsymbol{\Lambda}_m = \left\{\Lambda_{m,n}\right\}_{n=1}^{N}$ is identifiable. This reflects the fact that $N$-input $M$-output channels can be separated into $M$ systems of $N$-input single-output channels.

### *4.3.2 Linear Constraints*

The necessary and sufficient condition for identifiability of $OPW(\boldsymbol{\Lambda})$ presented in Corollary 4.2 is based on the assumption that all subchannels and their components are independent. If some linear relationship between and within subchannel is known, for instance, if transmission antennas (or receiving antennas) are not well separated, one could take advantage of such information in channel identification.

Let us formalize the concept of such relationship in terms of linear constraints. In the SISO setting, we express the linear relations between the entries of $\boldsymbol{\eta} = \left\{\eta(k, \ell)\right\}_{(k,\ell) \in \mathbb{Z}_L \times \mathbb{Z}_L}$ by the equation $\boldsymbol{b} = \boldsymbol{A\eta}$, where $\boldsymbol{b} \in \mathbb{C}^P$ and $\boldsymbol{A} \in \mathbb{C}^{P \times L^2}$ for some $P \in \mathbb{N}$. Combining with (4.6), we obtain

$$\begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{b} \end{bmatrix} = \begin{bmatrix} \boldsymbol{G}(\boldsymbol{c}) \\ \boldsymbol{A} \end{bmatrix} \boldsymbol{\eta}.$$

If $\boldsymbol{\eta} \in \mathbb{C}^{L^2}$ is known to be supported in a set $\Lambda \subset \mathbb{Z}_L \times \mathbb{Z}_L$, the system reduces to

$$\begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{b} \end{bmatrix} = \begin{bmatrix} \boldsymbol{G}(\boldsymbol{c})|_\Lambda \\ \boldsymbol{A}|_\Lambda \end{bmatrix} \boldsymbol{\eta}|_\Lambda. \tag{4.15}$$

In the MIMO setting, writing (see (4.6))

$$\boldsymbol{H}_{m,n}\, \boldsymbol{c}^{(n)} = \boldsymbol{G}(\boldsymbol{c}^{(n)})\, \boldsymbol{\eta}_{m,n} \quad \text{for } m = 1, \dots, M, \quad n = 1, \dots, N,$$

yields the equation

$$\begin{bmatrix} \boldsymbol{y}_1 \\ \vdots \\ \boldsymbol{y}_M \end{bmatrix} = \boldsymbol{H}\boldsymbol{c} = \begin{bmatrix} \boldsymbol{H}_{1,1} & \cdots & \boldsymbol{H}_{1,N} \\ \vdots & & \vdots \\ \boldsymbol{H}_{M,1} & \cdots & \boldsymbol{H}_{M,N} \end{bmatrix} \begin{bmatrix} \boldsymbol{c}^{(1)} \\ \vdots \\ \boldsymbol{c}^{(N)} \end{bmatrix} = \begin{bmatrix} \sum_{n=1}^N \boldsymbol{G}(\boldsymbol{c}^{(n)})\, \boldsymbol{\eta}_{1,n} \\ \vdots \\ \sum_{n=1}^N \boldsymbol{G}(\boldsymbol{c}^{(n)})\, \boldsymbol{\eta}_{M,n} \end{bmatrix}$$

$$= \begin{bmatrix} \boldsymbol{G}(\boldsymbol{c}^{(1)}, \dots, \boldsymbol{c}^{(N)}) & 0 & \cdots & 0 \\ 0 & \boldsymbol{G}(\boldsymbol{c}^{(1)}, \dots, \boldsymbol{c}^{(N)}) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \boldsymbol{G}(\boldsymbol{c}^{(1)}, \dots, \boldsymbol{c}^{(N)}) \end{bmatrix} \begin{bmatrix} \boldsymbol{\eta}_1 \\ \boldsymbol{\eta}_2 \\ \vdots \\ \boldsymbol{\eta}_M \end{bmatrix},$$

where $\boldsymbol{G}(\boldsymbol{c}^{(1)}, \dots, \boldsymbol{c}^{(N)}) := [\, \boldsymbol{G}(\boldsymbol{c}^{(1)}) \,|\, \cdots \,|\, \boldsymbol{G}(\boldsymbol{c}^{(N)}) \,] \in \mathbb{C}^{L \times NL^2}$ and $\boldsymbol{\eta}_m = \{\boldsymbol{\eta}_{m,n}\}_{n=1}^N \in (\mathbb{C}^{L^2})^N$ for $m = 1, \dots, M$. Similarly, linear relations between and within the vectors $\boldsymbol{\eta}_m$, $m = 1, \dots, M$, are expressed by the equation

$$\boldsymbol{b} = \textstyle\sum_{m=1}^M \boldsymbol{A}_m\, \boldsymbol{\eta}_m,$$

where $\boldsymbol{b} \in \mathbb{C}^P$ and $\boldsymbol{A}_m \in \mathbb{C}^{P \times L^2}$ for some $P \in \mathbb{N}$. Combining the above equations, we obtain

$$\begin{bmatrix} \boldsymbol{y}_1 \\ \boldsymbol{y}_2 \\ \vdots \\ \boldsymbol{y}_M \\ \boldsymbol{b} \end{bmatrix} = \begin{bmatrix} \boldsymbol{G}(\boldsymbol{c}^{(1)}, \dots, \boldsymbol{c}^{(N)}) & 0 & \cdots & 0 \\ 0 & \boldsymbol{G}(\boldsymbol{c}^{(1)}, \dots, \boldsymbol{c}^{(N)}) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \boldsymbol{G}(\boldsymbol{c}^{(1)}, \dots, \boldsymbol{c}^{(N)}) \\ \boldsymbol{A}_1 & \boldsymbol{A}_2 & \cdots & \boldsymbol{A}_M \end{bmatrix} \begin{bmatrix} \boldsymbol{\eta}_1 \\ \boldsymbol{\eta}_2 \\ \vdots \\ \boldsymbol{\eta}_M \end{bmatrix}.$$

If each $\boldsymbol{\eta}_m = \{\boldsymbol{\eta}_{m,n}\}_{n=1}^N \in (\mathbb{C}^{L^2})^N$ is supported in $\Lambda_m = \{\Lambda_{m,n}\}_{n=1}^N \subset (\mathbb{Z}_L \times \mathbb{Z}_L)^N$, the equation reduces to

$$\begin{bmatrix} \boldsymbol{y}_1 \\ \boldsymbol{y}_2 \\ \vdots \\ \boldsymbol{y}_M \\ \boldsymbol{b} \end{bmatrix} = \begin{bmatrix} \boldsymbol{G}(\boldsymbol{c}^{(1)},\ldots,\boldsymbol{c}^{(N)})|_{\Lambda_1} & 0 & \cdots & 0 \\ 0 & \boldsymbol{G}(\boldsymbol{c}^{(1)},\ldots,\boldsymbol{c}^{(N)})|_{\Lambda_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \boldsymbol{G}(\boldsymbol{c}^{(1)},\ldots,\boldsymbol{c}^{(N)})|_{\Lambda_M} \\ \boldsymbol{A}_1|_{\Lambda_1} & \boldsymbol{A}_2|_{\Lambda_2} & \cdots & \boldsymbol{A}_M|_{\Lambda_M} \end{bmatrix} \begin{bmatrix} \boldsymbol{\eta}_1|_{\Lambda_1} \\ \boldsymbol{\eta}_2|_{\Lambda_2} \\ \vdots \\ \boldsymbol{\eta}_M|_{\Lambda_M} \end{bmatrix}. \tag{4.16}$$

The constraints $\boldsymbol{b} = \boldsymbol{A}|_\Lambda\, \boldsymbol{\eta}|_\Lambda$ in the SISO case, and $\boldsymbol{b} = \sum_{m=1}^{M} \boldsymbol{A}_m|_{\Lambda_m}\, \boldsymbol{\eta}_m|_{\Lambda_m}$ in the MIMO case, are referred to as the *side constraints* associated with $OPW(\Lambda)$ and $OPW(\boldsymbol{\Lambda})$, respectively.

The discussion above immediately leads to the following result.

**Proposition 4.4**

*(a) For $\Lambda \subset \mathbb{Z}_L \times \mathbb{Z}_L$ and $\boldsymbol{A} \in \mathbb{C}^{P\times L^2}$ with some $P \in \mathbb{N}$, the space $OPW(\Lambda)$ with side constraints of the form $\boldsymbol{b} = \boldsymbol{A}|_\Lambda\, \boldsymbol{\eta}|_\Lambda$ is identifiable if and only if there exists $\boldsymbol{c} \in \mathbb{C}^L$ such that the matrix*

$$\begin{bmatrix} \boldsymbol{G}(\boldsymbol{c})|_\Lambda \\ \boldsymbol{A}|_\Lambda \end{bmatrix} \tag{4.17}$$

*is injective.*

*(b) For $\boldsymbol{\Lambda} = [\Lambda_{m,n}]_{m=1\,n=1}^{M\quad N}$ with $\Lambda_{m,n} \subset \mathbb{Z}_L \times \mathbb{Z}_L$ and matrices $\boldsymbol{A}_m \in \mathbb{C}^{P\times L^2}$, $m = 1,\ldots,M$, with some $P \in \mathbb{N}$, the space $OPW(\boldsymbol{\Lambda})$ with side constraints of the form $\boldsymbol{b} = \sum_{m=1}^{M} \boldsymbol{A}_m|_{\Lambda_m}\, \boldsymbol{\eta}_m|_{\Lambda_m}$ is identifiable if and only if there exists $\boldsymbol{c} = (\boldsymbol{c}^{(1)},\ldots,\boldsymbol{c}^{(N)}) \in (\mathbb{C}^L)^N$ such that the matrix*

$$\begin{bmatrix} \boldsymbol{G}(\boldsymbol{c}^{(1)},\ldots,\boldsymbol{c}^{(N)})|_{\boldsymbol{\Lambda}_1} & 0 & \cdots & 0 \\ 0 & \boldsymbol{G}(\boldsymbol{c}^{(1)},\ldots,\boldsymbol{c}^{(N)})|_{\boldsymbol{\Lambda}_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \boldsymbol{G}(\boldsymbol{c}^{(1)},\ldots,\boldsymbol{c}^{(N)})|_{\boldsymbol{\Lambda}_M} \\ \boldsymbol{A}_1|_{\boldsymbol{\Lambda}_1} & \boldsymbol{A}_2|_{\boldsymbol{\Lambda}_2} & \cdots & \boldsymbol{A}_M|_{\boldsymbol{\Lambda}_M} \end{bmatrix} \tag{4.18}$$

*is injective. Here, $\boldsymbol{G}(\boldsymbol{c}^{(1)},\ldots,\boldsymbol{c}^{(N)}) := [\boldsymbol{G}(\boldsymbol{c}^{(1)}) \,|\, \cdots \,|\, \boldsymbol{G}(\boldsymbol{c}^{(N)})] \in \mathbb{C}^{L\times NL^2}$ and $\boldsymbol{\Lambda}_m = \{\Lambda_{m,n}\}_{n=1}^{N} \subset (\mathbb{Z}_L \times \mathbb{Z}_L)^N$.*

Clearly, choosing the empty set of side constraints would reduce the matrix in (4.17) to $\boldsymbol{G}(\boldsymbol{c})|_\Lambda$, which, for an appropriate choice of $\boldsymbol{c} \in \mathbb{C}^L$, is injective whenever $|\Lambda| \le L$ (see Corollary 4.1). Likewise, for an appropriate choice of $\boldsymbol{c} \in (\mathbb{C}^L)^N$, the matrix (4.18) without the last row is injective if $\sum_{n=1}^{N} |\Lambda_{m,n}| \le L$ for all $m = 1,\ldots,M$ (see Corollary 4.2).

Proposition 4.4 converts the problem of identifiability of SISO/MIMO operator Paley–Wiener spaces with side constraints, into injectivity of certain matrices, namely, the matrix (4.17) in the SISO case and the matrix (4.18) in the MIMO case. In the SISO case, we will show that if $\boldsymbol{A} \in \mathbb{C}^{P\times L^2}$ consists of a single row, i.e., if $P = 1$, then for each $\Lambda \subset \mathbb{Z}_L \times \mathbb{Z}_L$ with $|\Lambda| = L + 1$ the matrix (4.17) is injective for some $\boldsymbol{c} \in \mathbb{C}^L$, and hence, the space $OPW(\Lambda)$ with side constraints of the form $\boldsymbol{b} = \boldsymbol{A}|_\Lambda\, \boldsymbol{\eta}|_\Lambda$ is always identifiable. Compared to Corollary 4.1, this result overcomes the fundamental restriction on the size of $\Lambda$ by exploiting the additional constraints.

**Theorem 4.1 ([16, 17])** *For any $\Lambda \subset \mathbb{Z}_L \times \mathbb{Z}_L$ with $|\Lambda| = L+1$ and $\boldsymbol{a} \in \mathbb{C}^{L+1}\backslash\{0\}$, there exists a vector $\boldsymbol{c} \in \mathbb{C}^L$ for which the $(L+1)\times(L+1)$ matrix $\begin{bmatrix} \boldsymbol{G}(\boldsymbol{c})|_\Lambda \\ \boldsymbol{a}^* \end{bmatrix}$ is invertible. Moreover, such vectors $\boldsymbol{c}$ form a dense open subset of $\mathbb{C}^L$ with full measure.*

The proof of this theorem is based on the following lemma.

**Lemma 4.1 ([16, 17])** *Let $\Lambda \subset \mathbb{Z}_L \times \mathbb{Z}_L$ with $L + 1 \le R := |\Lambda| \le 2L$. Then $span\{ker\, \boldsymbol{G}(\boldsymbol{c})|_\Lambda : \boldsymbol{c} \in U_L\} = \mathbb{C}^R$, where $U_L$ is the set of all $\boldsymbol{c} \in \mathbb{C}^L$ so that $\boldsymbol{G}(\boldsymbol{c})$ has full spark (see Proposition 4.1).*

Unfortunately, to obtain an identifiability result from this lemma requires to restrict ourselves to $|\Lambda| = L + 1$ as in Theorem 4.1.

Theorem 4.1 does not allow us to draw conclusions for the case of linear constraints with multiple equations. Indeed, if $\boldsymbol{A} \in \mathbb{C}^{P\times L^2}$ has multiple rows, i.e., if $P \ge 2$, the intersection of the row spaces of $\boldsymbol{A}|_\Lambda$ and $\boldsymbol{G}(\boldsymbol{c})|_\Lambda$ may depend on the choice of $\boldsymbol{c}$. Below we give an example of $\Lambda \subset \mathbb{Z}_L \times \mathbb{Z}_L$ with size $L + 2$ and linear constraints of two equations such that the matrix $\begin{bmatrix} \boldsymbol{G}(\boldsymbol{c})|_\Lambda \\ \boldsymbol{A}|_\Lambda \end{bmatrix}$ is singular for all $\boldsymbol{c} \in \mathbb{C}^L$.

*Example 4.1* Let $L = 3$ and $\Lambda = \{(0, 0), (0, 1), (0, 2), (1, 0), (1, 1)\}$. For every $\boldsymbol{c} = (c_0, c_1, c_2)^{\mathbb{T}} \in \mathbb{C}^3$, the matrix

$$\begin{bmatrix} \boldsymbol{G}(\boldsymbol{c})|_\Lambda \\ \boldsymbol{A}|_\Lambda \end{bmatrix} = \begin{bmatrix} c_0 & c_0 & c_0 & c_1 & c_1 \\ c_1 & \omega c_1 & \omega^2 c_1 & c_2 & \omega c_2 \\ c_2 & \omega^2 c_2 & \omega^4 c_2 & c_0 & \omega^2 c_0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

is singular because the first row is a linear combination of the fourth and fifth rows.

We also provide a matrix $\boldsymbol{A} \in \mathbb{C}^{L\times L^2}$ with the property that the matrix $\begin{bmatrix} \boldsymbol{G}(\boldsymbol{c})|_\Lambda \\ \boldsymbol{A}|_\Lambda \end{bmatrix}$ is not injective for all $\boldsymbol{c} \in \mathbb{C}^L$ and $\Lambda \subset \mathbb{Z}_L \times \mathbb{Z}_L$ with size $2L$.

*Example 4.2* Let $\boldsymbol{A} = [\, \boldsymbol{I}_L \,|\, \boldsymbol{M}^{-1} \,|\, \cdots \,|\, \boldsymbol{M}^{-(L-1)} \,] \in \mathbb{C}^{L\times L^2}$. The $2L \times L^2$ matrix $\begin{bmatrix} \boldsymbol{G}(\boldsymbol{c}) \\ \boldsymbol{A} \end{bmatrix}$ is rank deficient for all $\boldsymbol{c} \in \mathbb{C}^L$.

We now extend Lemma 4.1 and Theorem 4.1 to the MIMO setting.

**Lemma 4.2 ([16, 17])** *Let $L \geq 2$, $N \geq 1$, and $\Lambda^{(1)}, \ldots, \Lambda^{(N)} \subset \mathbb{Z}_L \times \mathbb{Z}_L$ with $L+1 \leq R = \sum_{n=1}^{N} |\Lambda^{(n)}| < 2L$. Then,*

$$\mathrm{span}\left\{ \ker\left[ \boldsymbol{G}(\boldsymbol{c}^{(1)})|_{\Lambda^{(1)}} \ \cdots \ \boldsymbol{G}(\boldsymbol{c}^{(N)})|_{\Lambda^{(N)}} \right] : (\boldsymbol{c}^{(1)}, \ldots, \boldsymbol{c}^{(N)}) \in U_{L,N} \right\} = \mathbb{C}^R,$$

*where $U_{L,N}$ is the set of all $(\boldsymbol{c}^{(1)}, \ldots, \boldsymbol{c}^{(N)}) \in (\mathbb{C}^L)^N$ such that $[\boldsymbol{G}(\boldsymbol{c}^{(1)}) \ \cdots \ \boldsymbol{G}(\boldsymbol{c}^{(N)})]$ has full spark (see Proposition 4.3).*

**Theorem 4.2 ([16, 17])** *Let $L \geq 2$, $M, N \geq 1$, $\boldsymbol{\Lambda} = [\Lambda_{m,n}]_{m=1}^{M}{}_{n=1}^{N}$ with $\Lambda_{m,n} \subset \mathbb{Z}_L \times \mathbb{Z}_L$ and $\sum_{m=1}^{M} \sum_{n=1}^{N} |\Lambda_{m,n}| = L+1$, and $\boldsymbol{a} = (\boldsymbol{a}_1, \ldots, \boldsymbol{a}_M)^{\mathbb{T}} \in \mathbb{C}^{L+1} \backslash \{0\}$, where $\boldsymbol{a}_m$ is a vector of dimension $\sum_{n=1}^{N} |\Lambda_{m,n}|$. There exists a vector $(\boldsymbol{c}^{(1)}, \ldots, \boldsymbol{c}^{(N)}) \in (\mathbb{C}^L)^N$ such that the $(L+1) \times (L+1)$ matrix*

$$\begin{bmatrix} \boldsymbol{G}(\boldsymbol{c}^{(1)}, \ldots, \boldsymbol{c}^{(N)})|_{\Lambda_1} & 0 & \cdots & 0 \\ 0 & \boldsymbol{G}(\boldsymbol{c}^{(1)}, \ldots, \boldsymbol{c}^{(N)})|_{\Lambda_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \boldsymbol{G}(\boldsymbol{c}^{(1)}, \ldots, \boldsymbol{c}^{(N)})|_{\Lambda_M} \\ \boldsymbol{a}_1^* & \boldsymbol{a}_2^* & \cdots & \boldsymbol{a}_M^*, \end{bmatrix} \tag{4.19}$$

*where $\Lambda_m = \{\Lambda_{m,n}\}_{n=1}^{N} \subset (\mathbb{Z}_L \times \mathbb{Z}_L)^N$, is invertible. Moreover, such vectors $(\boldsymbol{c}^{(1)}, \ldots, \boldsymbol{c}^{(N)})$ constitute a dense open subset of $(\mathbb{C}^L)^N$ with full measure. Hence, the MIMO operator Paley–Wiener space $OPW(\boldsymbol{\Lambda}) = [\, OPW(\Lambda_{m,n}) \,]_{m=1}^{M}{}_{n=1}^{N}$ with side constraints $b = \sum_{m=1}^{M} \boldsymbol{a}_m^* \, \boldsymbol{\eta}_m|_{\Lambda_m}$, where $b \in \mathbb{C}$ and $\boldsymbol{\eta}_m = \{\boldsymbol{\eta}_{m,n}\}_{n=1}^{N} \in (\mathbb{C}^{L^2})^N$ for $m = 1, \ldots, M$, is identifiable.*

Concerning side constraints with multiple equations, Example 4.1 in the SISO case clearly implies that Theorem 4.2 cannot be generalized to linear side constraints with two or more equations.

We remark that our results are in the fully deterministic setting: for a *given* set of linear constraints, we seek generators $\boldsymbol{c}$ for which the associated matrix is invertible. It would be interesting to consider the case where the linear constraints are chosen randomly. For generic linear constraints, the situation like Example 4.1 could be ignored and therefore larger support sets could be considered.

### 4.3.3 Message Transmission Using Unidentified Channels

In this section, we discuss the topic of transmitting messages with unknown channels where the primary goal is to transmit messages exactly and the secondary

goal is to identify the channel if possible. In communications, one usually tests a communication channel with a pilot signal to identify/estimate the channel before using it to transmit messages. The receiver is then able to recover messages based on the channel information [6, 31]. However, such a two-step method is not so useful for rapidly varying channels. There are several known methods in the literature on how to improve the reliability of transmission scheme [32, 33], but there is little work in the direction of simultaneous message transmission and channel identification [8, 10, 11, 14].

#### 4.3.3.1 Problem Formulation

We consider deterministic SISO channels $\boldsymbol{H} \in OPW(\Lambda)$ with $\Lambda \subset \mathbb{Z}_L \times \mathbb{Z}_L$ (see Definition 4.6). The standard approach for message transmission using $\boldsymbol{H}$ is in two steps: first, the channel $\boldsymbol{H}$ is tested with a pilot signal $\boldsymbol{c} \in \mathbb{C}^L$ to identify/estimate the channel, and then messages $\boldsymbol{z} \in \mathcal{Z}$ are transmitted through $\boldsymbol{H}$ and the receiver decodes the received signal based on the channel information. Our strategy is to combine these two steps and to rather send $\boldsymbol{z} + \boldsymbol{c}$ into the channel $\boldsymbol{H}$ without identifying $\boldsymbol{H}$ in advance.

**Message Transmission Problem** We assume that $\Lambda \subset \mathbb{Z}_L \times \mathbb{Z}_L$ and $\mathcal{Z} \subset \mathbb{C}^L$ are given, while the choice of $\boldsymbol{c} \in \mathbb{C}^L$ is up to the user. What conditions on $\Lambda \subset \mathbb{Z}_L \times \mathbb{Z}_L$ and $\mathcal{Z} \subset \mathbb{C}^L$ are necessary and/or sufficient so that there exists a vector $\boldsymbol{c} \in \mathbb{C}^L$ with the property that every $\boldsymbol{z} \in \mathcal{Z}$ can be recovered uniquely from $\boldsymbol{y} = \boldsymbol{H}(\boldsymbol{z} + \boldsymbol{c})$ with $\boldsymbol{H} \in OPW(\Lambda)$ unknown? Certainly, one may also consider the case where $\Lambda \subset \mathbb{Z}_L \times \mathbb{Z}_L$ is unknown.

#### 4.3.3.2 Message Transmission with Known Support

We first consider the case where $\Lambda \subset \mathbb{Z}_L \times \mathbb{Z}_L$ is known. A naive approach to the problem is to first identify the channel $\boldsymbol{H} \in OPW(\Lambda)$ and then use the channel information to transmit the message $\boldsymbol{z} \in \mathcal{Z}$. Our goal, however, is to successfully transmit and recover the message, so identifying the channel $\boldsymbol{H}$ is in principle not necessary. Let us clearly define what we mean by the message being uniquely recoverable.

**Definition 4.8** Let $\mathcal{H} = OPW(\Lambda)$ with $\Lambda \subset \mathbb{Z}_L \times \mathbb{Z}_L$, and let $\mathcal{Z} \subset \mathbb{C}^L$ and $\boldsymbol{c} \in \mathbb{C}^L$. We say that every $\boldsymbol{z} \in \mathcal{Z}$ is *uniquely recoverable* from the measurement $\boldsymbol{y} = \boldsymbol{H}(\boldsymbol{z} + \boldsymbol{c})$ with $\boldsymbol{H} \in \mathcal{H} \backslash \{0\}$ unknown if

$$\boldsymbol{H}(\boldsymbol{z} + \boldsymbol{c}) = \boldsymbol{H}'(\boldsymbol{z}' + \boldsymbol{c}) \text{ for some } \boldsymbol{H}, \boldsymbol{H}' \in \mathcal{H} \backslash \{0\} \text{ and } \boldsymbol{z}, \boldsymbol{z}' \in \mathcal{Z} \text{ implies } \boldsymbol{z} = \boldsymbol{z}'.$$

For the input signal $\boldsymbol{z} + \boldsymbol{c}$, the channel output of $\boldsymbol{H} \in OPW(\Lambda)$ is given by

$$\boldsymbol{y} = \boldsymbol{H}(\boldsymbol{z} + \boldsymbol{c}) = \boldsymbol{G}(\boldsymbol{z} + \boldsymbol{c})|_\Lambda \, \boldsymbol{\eta}_{\boldsymbol{H},\Lambda} = \boldsymbol{G}(\boldsymbol{z})|_\Lambda \, \boldsymbol{\eta}_{\boldsymbol{H},\Lambda} + \boldsymbol{G}(\boldsymbol{c})|_\Lambda \, \boldsymbol{\eta}_{\boldsymbol{H},\Lambda}, \tag{4.20}$$

where $\boldsymbol{\eta}_{\boldsymbol{H},\Lambda} = \{\eta_{\boldsymbol{H}}(k,\ell)\}_{(k,\ell)\in\Lambda}$. If $\operatorname{ran}\boldsymbol{G}(\boldsymbol{z})|_\Lambda \cap \operatorname{ran}\boldsymbol{G}(\boldsymbol{c})|_\Lambda = \{0\}$, one could immediately write $\boldsymbol{y} = \boldsymbol{y}' + \boldsymbol{y}''$ with unique components $\boldsymbol{y}' = \boldsymbol{G}(\boldsymbol{z})|_\Lambda \boldsymbol{\eta}_{\boldsymbol{H},\Lambda} \in \operatorname{ran}\boldsymbol{G}(\boldsymbol{z})|_\Lambda$ and $\boldsymbol{y}'' = \boldsymbol{G}(\boldsymbol{c})|_\Lambda \boldsymbol{\eta}_{\boldsymbol{H},\Lambda} \in \operatorname{ran}\boldsymbol{G}(\boldsymbol{c})|_\Lambda$. This would then allow to identify $\boldsymbol{\eta}_{\boldsymbol{H},\Lambda}$ from $\boldsymbol{y}''$ (provided that $\boldsymbol{G}(\boldsymbol{c})|_\Lambda$ is an injective matrix) and, in turn, to recover $\boldsymbol{z}$ from $\boldsymbol{y}' = \boldsymbol{H}\boldsymbol{z}$ (provided that $\boldsymbol{H}$ is injective on $\mathcal{Z}$).

Let us consider the case where $\mathcal{Z} \subset \mathbb{C}^L$ is a one-dimensional subspace of $\mathbb{C}^L$, say, $\mathcal{Z} = \operatorname{span}\{\boldsymbol{x}\}$ for some fixed $\boldsymbol{x} \in \mathbb{C}^L\backslash\{0\}$. Then (4.20) becomes

$$\boldsymbol{y} = \boldsymbol{H}(\boldsymbol{z}+\boldsymbol{c}) = \begin{bmatrix}\boldsymbol{G}(\boldsymbol{x})|_\Lambda & \boldsymbol{G}(\boldsymbol{c})|_\Lambda\end{bmatrix}\begin{bmatrix}u\boldsymbol{\eta}_{\boldsymbol{H},\Lambda} \\ \boldsymbol{\eta}_{\boldsymbol{H},\Lambda}\end{bmatrix}. \tag{4.21}$$

and recovering the message vector $\boldsymbol{z} = u\boldsymbol{x}$ with $u \in \mathbb{C}$ is equivalent to recovering its coefficient $u$. To this end, it is desirable that $\dim\operatorname{ran}\boldsymbol{G}(\boldsymbol{x})|_\Lambda$ is as small as possible so as to reserve enough space for $\operatorname{ran}\boldsymbol{G}(\boldsymbol{c})|_\Lambda$. Note that $\boldsymbol{G}(\boldsymbol{c})|_\Lambda$ needs to have linearly independent columns for the exact recovery of $\boldsymbol{\eta}_{\boldsymbol{H},\Lambda}$ from $\boldsymbol{G}(\boldsymbol{c})|_\Lambda\boldsymbol{\eta}_{\boldsymbol{H},\Lambda}$. Note however that the problem depends on the choice of $\boldsymbol{x}$ and $\Lambda$ (while $\boldsymbol{c} \in \mathbb{C}^L$ can be chosen by the user), and there is no general solution for the recovery of $u$.

The following theorem appeared in [14] without proof and provides a solution to our problem in the case that $\operatorname{ran}\boldsymbol{G}(\boldsymbol{x})|_\Lambda$ is a one-dimensional subspace of $\mathbb{C}^L$ and $|\Lambda| \le L-1$ (for instance, consider $\boldsymbol{z} = (1,1,\ldots,1)$ and $\Lambda = \{0,\ldots,L-2\}\times\{0\}$). Indeed, if $\operatorname{ran}\boldsymbol{G}(\boldsymbol{x})|_\Lambda = \operatorname{span}\{\boldsymbol{a}\}$, then there exists a vector $\boldsymbol{c} \in \mathbb{C}^L$ such that $\operatorname{ran}\boldsymbol{G}(\boldsymbol{z})|_\Lambda \cap \operatorname{ran}\boldsymbol{G}(\boldsymbol{c})|_\Lambda = \{0\}$ and so one can use the arguments described above.

**Theorem 4.3 ([14])** *Let $\Lambda \subset \mathbb{Z}_L\times\mathbb{Z}_L$ with $1 \le |\Lambda| \le L-1$ and $\boldsymbol{a} \in \mathbb{C}^L\backslash\{0\}$. There exists a vector $\boldsymbol{c} \in \mathbb{C}^L\backslash\{0\}$ such that the matrix $[\,\boldsymbol{G}(\boldsymbol{c})|_\Lambda,\ \boldsymbol{a}\,] \in \mathbb{C}^{L\times(|\Lambda|+1)}$ has full rank.*

Here we provide a short proof of Theorem 4.3, which relies on the following lemma whose detailed proof is given in section "Proof of Lemma 4.3".

**Lemma 4.3 ([14])** *Let $\Lambda \subset \mathbb{Z}_L\times\mathbb{Z}_L$ with $1 \le R = |\Lambda| \le L-1$. Then $\operatorname{span}\{\ker(\boldsymbol{G}(\boldsymbol{c})|_\Lambda)^* : \boldsymbol{c} \in \mathcal{S}\} = \mathbb{C}^L$, where $U_L$ is the set of all $\boldsymbol{c} \in \mathbb{C}^L$ such that $\boldsymbol{G}(\boldsymbol{c})$ has full spark.*

***Proof of Theorem 4.3*** By Lemma 4.3, we have

$$\bigcap_{\boldsymbol{c}\in U_L}\operatorname{ran}\boldsymbol{G}(\boldsymbol{c})|_\Lambda = \bigcap_{\boldsymbol{c}\in U_L}\left(\ker(\boldsymbol{G}(\boldsymbol{c})|_\Lambda)^*\right)^\perp = \{0\}.$$

This implies that for each $\boldsymbol{a} \in \mathbb{C}^L\backslash\{0\}$, there exists a vector $\boldsymbol{c} \in U_L$ satisfying $\boldsymbol{a} \notin \operatorname{ran}\boldsymbol{G}(\boldsymbol{c})|_\Lambda$. Since $\boldsymbol{G}(\boldsymbol{c})$ has full spark, the matrix $\boldsymbol{G}(\boldsymbol{c})|_\Lambda$ has full column rank, and hence we conclude that $[\,\boldsymbol{G}(\boldsymbol{c})|_\Lambda,\ \boldsymbol{a}\,]$ has full rank. □

*Remark 4.2* Theorem 4.3 and Lemma 4.3 are in analogy with Theorem 4.1 and Lemma 4.1 which are concerned with having an additional *row* to the Gabor submatrix $\boldsymbol{G}(\boldsymbol{c})|_\Lambda$.

The previous discussion relies on the condition $\operatorname{ran} \boldsymbol{G}(\boldsymbol{z})|_\Lambda \cap \operatorname{ran} \boldsymbol{G}(\boldsymbol{c})|_\Lambda = \{0\}$. This, however, is often more than what is needed. For instance, if $\mathcal{Z} \subset \mathbb{C}^L$ is an $R$-dimensional subspace of $\mathbb{C}^L$, then requiring $\operatorname{ran} \boldsymbol{G}(\boldsymbol{z})|_\Lambda \cap \operatorname{ran} \boldsymbol{G}(\boldsymbol{c})|_\Lambda = \{0\}$ for all $\boldsymbol{z} \in \mathcal{Z}\backslash\{0\}$ would imply $|\Lambda| \cdot (R+1) \leq L$, which is a very tough restriction. It turns out that only $|\Lambda| + R \leq L$ is necessary for the recovery of $\boldsymbol{z} \in \mathcal{Z}$, as we will see in Proposition 4.6 below.

For $\Lambda \subset \mathbb{Z}_L \times \mathbb{Z}_L$, we define $B(\Lambda) = \{\boldsymbol{M}^\ell \boldsymbol{T}^k : (k, \ell) \in \Lambda\}$, which is a basis for $OPW(\Lambda)$. In particular, for the cyclic subgroups

$$\Gamma_s = \big\langle (1, s) \big\rangle = \{(0,0), (1,s), \ldots, (L-1, (L-1)s)\} \quad \text{for } s = 0, \ldots, L-1,$$

$$\Gamma_\infty = \big\langle (0,1) \big\rangle = \{(0,0), (0,1), \ldots, (0, L-1)\},$$

it follows from the commutation relation $\boldsymbol{M}^\ell \boldsymbol{T}^k = \omega^{k\ell}\, \boldsymbol{T}^k \boldsymbol{M}^\ell$, $k, \ell = 0, \ldots, L-1$, that all elements in each family $B(\Gamma_s)$ commute. Therefore, they are simultaneously diagonalizable.[2] The next proposition provides the common eigenvectors for $B(\Gamma_s)$.

For $L \in \mathbb{N}$, the Fourier vectors in $\mathbb{C}^L$ are defined as $\boldsymbol{v}_j = \frac{1}{\sqrt{L}} \left(1, \omega^j, \ldots, \omega^{(L-1)j}\right)^{\mathbb{T}}$ for $j = 0, \ldots, L-1$, where $\omega = e^{2\pi i/L}$. Let $\boldsymbol{D} \in \mathbb{C}^{L \times L}$ be the diagonal matrix with $D_{n,n} = \omega^{0+1+\cdots+n} = \omega^{n(n+1)/2}$ for $n = 0, 1, \ldots, L-1$.

**Proposition 4.5** *Let $L \in \mathbb{N}$ be an odd integer.*

*(a) The family $B(\Gamma_s)$ with $s \in \{0, \ldots, L-1\}$ has common eigenvectors $\boldsymbol{D}^s \boldsymbol{v}_j$, $j = 0, \ldots, L-1$, which form an orthonormal basis of $\mathbb{C}^L$.*

*(b) The family $B(\Gamma_\infty) = \{\mathbf{I}, \boldsymbol{M}, \ldots, \boldsymbol{M}^{L-1}\}$ has common eigenvectors $\boldsymbol{e}_j$, $j = 0, \ldots, L-1$.*

The proof of Proposition 4.5 follows easily from the following lemma.

**Lemma 4.4 (Cf. Lemma 2 in [11])** *Let $L \in \mathbb{N}$ be an odd integer. Then, for all $j, s \in \mathbb{Z}_L$, one has*

$$\boldsymbol{T}^k \boldsymbol{D}^s \boldsymbol{v}_j = \omega^{-jk + \frac{k(k-1)s}{2}} \boldsymbol{D}^s \boldsymbol{v}_{j-ks} \qquad \text{for } k \in \mathbb{Z}_L$$

$$\boldsymbol{M}^\ell \boldsymbol{D}^s \boldsymbol{v}_j = \boldsymbol{D}^s \boldsymbol{v}_{j+\ell} \qquad \text{for } \ell \in \mathbb{Z}_L.$$

[2] It is well known that a family $\mathcal{S}$ of diagonalizable square matrices is simultaneously diagonalizable, that is, there exists an invertible matrix $U$ such that $U^{-1}AU$ is a diagonal matrix for every $A \in \mathcal{S}$, if and only if all matrices in $\mathcal{S}$ commute.

*Consequently, for all* $j, k, s \in \mathbb{Z}_L$*, one has*

$$\boldsymbol{M}^{ks}\boldsymbol{T}^{k}\boldsymbol{D}^{s}\boldsymbol{v}_{j} = \omega^{-jk+\frac{k(k-1)s}{2}}\boldsymbol{D}^{s}\boldsymbol{v}_{j}. \tag{4.22}$$

*Remark 4.3* In Lemma 4.4, we require $L \in \mathbb{N}$ to be an odd integer to ensure that $D_{0,0} = D_{L,L} = \omega^{0+1+\cdots+L} = (\omega^{L})^{(L+1)/2} = 1$.

**Proposition 4.6** *Let* $L \geq 3$ *be a prime number, and let* $s \in \{0, 1, \ldots, L-1, \infty\}$*. If* $\Lambda \subset \Gamma_s$ *and if* $\mathcal{Z} \subset \mathbb{C}^L$ *is spanned by* $R$ *common eigenvectors of* $B(\Gamma_s)$ *with* $|\Lambda| + R \leq L$*, then any message* $\boldsymbol{z} \in \mathcal{Z}$ *can be uniquely recovered after sending through the above transmission scheme.*

A proof of Proposition 4.6 is given in section "Proof of Proposition 4.6". In the proof, the condition that $L \in \mathbb{N}$ is odd is required for applying Lemma 4.4, while the condition $L \in \mathbb{N}$ prime is needed when applying Chebotarev's theorem on roots of unity.

#### 4.3.3.3 Message Transmission with Unknown Support

Before addressing the message transmission problem for channels with unknown support, let us recall some necessary notions. The coherence for a matrix $\boldsymbol{\Phi} = [\varphi_1, \ldots, \varphi_N]$ with $\ell_2$-normalized columns, i.e., $\|\varphi_n\|_2 = 1$ for all $n$, is defined as

$$\mu(\boldsymbol{\Phi}) = \max_{i \neq j} \left|\langle \varphi_i, \varphi_j \rangle\right| .$$

In compressed sensing, the coherence is often used as a simple measure for the quality of measurement matrices, since recovery algorithms perform better for measurement matrices with smaller coherence [5]. There are known constructions of deterministic Gabor matrices with small coherence. For instance, the Gabor matrix $\boldsymbol{G}(\boldsymbol{c})$ generated by the *Alltop* window $\boldsymbol{c} \in \mathbb{C}^L$ with $L \geq 5$ prime,[3] defined as (see [1])

$$\boldsymbol{c}(n) = \tfrac{1}{\sqrt{L}}\, e^{2\pi i n^3/L} \quad \text{for } n \in \mathbb{Z}_L\,, \tag{4.23}$$

has coherence $\mu\left(\boldsymbol{G}(\boldsymbol{c})\right) = \frac{1}{\sqrt{L}}$ which is very close to the optimal lower bound $\frac{1}{\sqrt{L+1}}$, i.e., the Welch bound for $L \times L^2$ matrices (see [5, Proposition 5.13] for a computation of the coherence). Later, we will use the Alltop window as the pilot signal $\boldsymbol{c}$.

[3] For composite numbers $L \in \mathbb{N}$, the Alltop window $\boldsymbol{c} \in \mathbb{C}^L$ does not guarantee small coherence of $\boldsymbol{G}(\boldsymbol{c})$. See, for instance, [21].

For an odd integer $L \in \mathbb{N}$, Lemma 4.4 implies for all $j, k, \ell, r \in \mathbb{Z}_L$,

$$\boldsymbol{M}^{\ell}\boldsymbol{T}^{k}\boldsymbol{D}^{2r}\boldsymbol{v}_j = \omega^{rk^2-(r+j)k}\,\boldsymbol{D}^{2r}\boldsymbol{v}_{j+\ell-2rk}\,. \tag{4.24}$$

In particular, setting $\ell = 2rk$ gives

$$\boldsymbol{M}^{2rk}\boldsymbol{T}^{k}\boldsymbol{D}^{2r}\boldsymbol{v}_j = \omega^{rk^2-(r+j)k}\,\boldsymbol{D}^{2r}\boldsymbol{v}_j, \tag{4.25}$$

where $\omega = e^{2\pi i/L}$ and $\boldsymbol{D}^{2r}\boldsymbol{v}_j(n) = \frac{1}{\sqrt{L}}\,\omega^{rn^2+(r+j)n}$ for $n \in \mathbb{Z}_L$. To simplify the notation, we define the chirp signal $\boldsymbol{x}_{mL+r} \in \mathbb{C}^L$ with *base frequency* $m \in \mathbb{Z}_L$ and *chirp rate* $r \in \mathbb{Z}_L$ as

$$\boldsymbol{x}_{mL+r}(n) = \tfrac{1}{\sqrt{L}}\,\omega^{rn^2+mn} \quad \text{for } n \in \mathbb{Z}_L\,.$$

Setting $m = r + j$, one has $\boldsymbol{x}_{mL+r} = \boldsymbol{D}^{2r}\boldsymbol{v}_j$, and thus (4.24) and (4.25) can be written as

$$\boldsymbol{M}^{\ell}\boldsymbol{T}^{k}\boldsymbol{x}_{mL+r} = \omega^{rk^2-mk}\boldsymbol{x}_{(m+\ell-2rk)L+r} \tag{4.26}$$

and

$$\boldsymbol{M}^{2rk}\boldsymbol{T}^{k}\boldsymbol{x}_{mL+r} = \omega^{rk^2-mk}\boldsymbol{x}_{mL+r}\,, \tag{4.27}$$

respectively. We collect all the $L^2$ chirp signals in $\mathbb{C}^L$ as columns of the matrix $\boldsymbol{E} = [\boldsymbol{X}_0, \boldsymbol{X}_1, \ldots, \boldsymbol{X}_{L-1}] \in \mathbb{C}^{L\times L^2}$, where each $\boldsymbol{X}_r = \left[\boldsymbol{x}_r,\ \boldsymbol{x}_{L+r},\ \ldots,\ \boldsymbol{x}_{(L-1)L+r}\right]$ is the unitary matrix with columns consisting of all chirp signals with chirp rate $r$.

**Lemma 4.5 (Lemma 4 in [11])** *Let $L \geq 5$ be a prime, and let $\boldsymbol{c} \in \mathbb{C}^L$ be the Alltop window defined in (4.23). The coherence of the matrix $[\boldsymbol{G}(\boldsymbol{c})\ \boldsymbol{E}] \in \mathbb{C}^{L\times 2L^2}$ is bounded above by $2/\sqrt{L}$.*

We are now ready to address the message transmission problem for channels $\boldsymbol{H} \in OPW(\Lambda)$ with unknown support $\Lambda \subset \mathbb{Z}_L\times\mathbb{Z}_L$. Let $\boldsymbol{u} = \{u_r\}_{r=0}^{R-1} \in \mathbb{C}^R$ be a message vector of size $R$. The signal to be sent through $\boldsymbol{H}$ will be designed as

$$\boldsymbol{x} = \textstyle\sum_{r=0}^{R-1} u_r\boldsymbol{x}_r + \boldsymbol{c}, \tag{4.28}$$

where $\boldsymbol{c} \in \mathbb{C}^L$ is the Alltop window and $\boldsymbol{x}_r$, $r = 0, \ldots, R-1$, are chirp signals with base frequency $m = 0$ and chirp rate $r$. Note that since all $\boldsymbol{x}_r$ are linearly independent, the message vector $\boldsymbol{u} = \{u_r\}_{r=0}^{R-1}$ encoded in $\boldsymbol{z} = \sum_{r=0}^{R-1} u_r\boldsymbol{x}_r \in \mathcal{Z}$ can be retrieved uniquely from $\boldsymbol{z}$.

As the exact spreading support $\Lambda \subset \mathbb{Z}_L\times\mathbb{Z}_L$ of $\boldsymbol{H} \in OPW(\Lambda)$ is not known, we will simply employ the representation (4.5) for general linear maps

$\boldsymbol{H} \in \mathcal{L}(\mathbb{C}^L, \mathbb{C}^L)$, but with the assumption that $\boldsymbol{\eta} = \{\eta(k,\ell)\}_{(k,\ell)\in\mathbb{Z}_L\times\mathbb{Z}_L}$ is sparse. Substituting (4.28) into $\boldsymbol{H}$ yields (cf. (4.6))

$$\begin{aligned}
\boldsymbol{y} = \boldsymbol{H}(\boldsymbol{z}+\boldsymbol{c}) &= \sum_{(k,\ell)\in\mathbb{Z}_L\times\mathbb{Z}_L} \eta(k,\ell)\, \boldsymbol{M}^\ell \boldsymbol{T}^k \Big(\mathbf{c} + \textstyle\sum_{r=0}^{R-1} u_r \boldsymbol{x}_r\Big) \\
&= \sum_{(k,\ell)\in\mathbb{Z}_L\times\mathbb{Z}_L} \eta(k,\ell)\, \boldsymbol{M}^\ell \boldsymbol{T}^k \mathbf{c} + \sum_{r=0}^{R-1} \sum_{(k,\ell)\in\mathbb{Z}_L\times\mathbb{Z}_L} \eta(k,\ell)\, u_r\, \omega^{rk^2} \boldsymbol{x}_{(\ell-2rk)L+r} \\
&= \boldsymbol{G}(\boldsymbol{c})\,\boldsymbol{\eta} + \sum_{r=0}^{R-1}\sum_{m=0}^{L-1} \Big( \sum_{k=0}^{L-1} \eta(k, m+2rk)\, \omega^{rk^2}\, u_r \Big) \boldsymbol{x}_{mL+r} \\
&= \boldsymbol{G}(\boldsymbol{c})\,\boldsymbol{\eta} + \boldsymbol{E}\boldsymbol{s} = [\boldsymbol{G}(\boldsymbol{c})\ \boldsymbol{E}] \begin{bmatrix} \boldsymbol{\eta} \\ \boldsymbol{s} \end{bmatrix},
\end{aligned} \tag{4.29}$$

where $\boldsymbol{s} = \{s_{r,m}\}_{(r,m)\in\mathbb{Z}_L\times\mathbb{Z}_L}$ is given by

$$s_{r,m} = \begin{cases} \sum_{k=0}^{L-1} \eta(k, m+2rk)\, \omega^{rk^2}\, u_r & \text{if } 0 \le r \le R-1 \text{ and } m \in \mathbb{Z}_L\,, \\ 0 & \text{if } R \le r \le L-1 \text{ and } m \in \mathbb{Z}_L. \end{cases}$$

This corresponds to an underdetermined linear system consisting of $L$ equations in the $2L^2$ variables $[\boldsymbol{\eta}, \boldsymbol{s}]^{\mathbb{T}}$, and the associated $L\times 2L^2$ matrix $[\boldsymbol{G}(\boldsymbol{c})\ \boldsymbol{E}]$ is guaranteed to have small coherence by Lemma 4.5. If $[\boldsymbol{\eta}, \boldsymbol{s}]^{\mathbb{T}}$ is known to be sparse, one could apply compressed sensing methods to recover it from (4.29). Note that for $\boldsymbol{H} \in OPW(\Lambda)$, we have $|\mathrm{supp}\,(\boldsymbol{s})| \le R\,|\Lambda|$ and thus $|\mathrm{supp}\,([\boldsymbol{\eta}, \boldsymbol{s}]^{\mathbb{T}})| \le (1+R)\,|\Lambda|$. If $R \le \frac{\sqrt{L}}{4|\Lambda|} - 1$, then

$$|\mathrm{supp}\,([\boldsymbol{\eta}, \boldsymbol{s}]^{\mathbb{T}})| \le (1+R)\,|\Lambda| \le \tfrac{\sqrt{L}}{4} \le \frac{1}{2\,\mu\big([\boldsymbol{G}(\boldsymbol{c})\ \boldsymbol{E}]\big)} \le \tfrac{1}{2}\Big(1 + \frac{1}{\mu\big([\boldsymbol{G}(\boldsymbol{c})\ \boldsymbol{E}]\big)}\Big),$$

so one could immediately apply [4, Theorems 4.3 and 4.5] (also see [5, Corollary 5.4 and Theorems 5.14 and 5.15]) to obtain the following result.

**Theorem 4.4 (Theorem 5 in [11])** *Let $L \ge 5$ be a prime, and let $\boldsymbol{H} \in OPW(\Lambda)$ with unknown $\Lambda \subset \mathbb{Z}_L\times\mathbb{Z}_L$. If $R \le \frac{\sqrt{L}}{4|\Lambda|} - 1$, then any message vector of size $R$ can be transmitted through $\boldsymbol{H}$ and be recovered exactly via orthogonal matching pursuit (OMP) or basis pursuit.*

Figure 4.1 shows a simulation result for Theorem 4.4 with randomly generated channels and messages. The $x$-axis is the size of $\Lambda \subset \mathbb{Z}_L\times\mathbb{Z}_L$ and the $y$-axis is the size $R$ of messages. The $\ell_2$-error of the recovered messages is shown in grayscale with black and white meaning small and large error, respectively. The green line $R|\Lambda| = L$ is a fundamental threshold due to degrees of freedom. Message recovery is in principle not possible in the region above this threshold. We refer to [11] for more details.

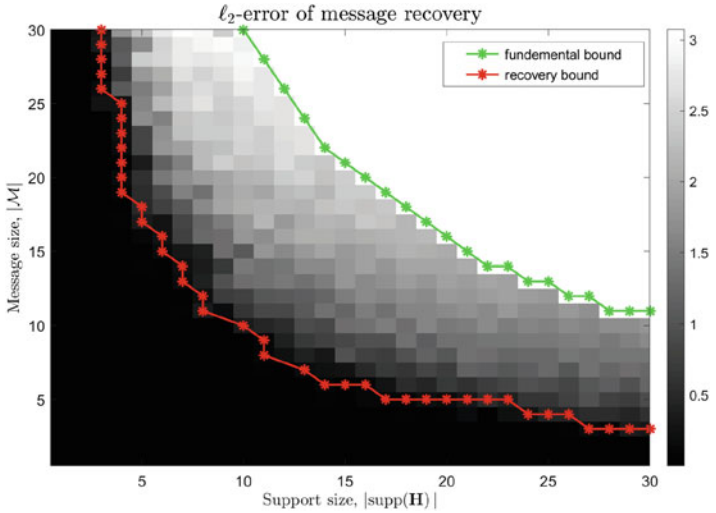



**Fig. 4.1** Message recovery error rates for $L = 307$

## 4.4 Results in Stochastic Setting

We now turn to the case of stochastic channels. In contrast to deterministic channels whose identifiability depends only on the size of the spreading support (see Corollaries 4.1 and 4.2), we will see that identifiability of stochastic channels (in the sense of Definition 4.5) relies not only on the size but also on the geometry of the support set of the covariance $R_{\eta_H}$. In Sect. 4.4.1, we discuss some support patterns which allow for channel identification (called *permissible patterns*) and those which do not (called *defective patterns*). As in the deterministic setting, we consider in Sect. 4.4.2 the problem of utilizing known linear side constraints in the stochastic setting. Some numerical experiments supporting our results are presented in Sect. 4.4.3.

A stochastic SISO channel is described by (4.9), and using a pilot input signal $\boldsymbol{c} \in \mathbb{C}^L$ yields

$$\boldsymbol{y}(\xi) = \boldsymbol{H}\boldsymbol{c}(\xi) = \boldsymbol{G}(\boldsymbol{c})\,\boldsymbol{\eta}_H(\xi)\,, \tag{4.30}$$

where $\boldsymbol{\eta}_H(\xi) = \{\eta_H(k,\ell;\xi)\}_{(k,\ell)\in\mathbb{Z}_L\times\mathbb{Z}_L}$ is a random vector in $\mathbb{C}^{L^2}$ and $\boldsymbol{G}(\boldsymbol{c})$ is the Gabor matrix generated by $\boldsymbol{c}$. Recall from Sect. 4.2.3 that the covariance matrices $\boldsymbol{R}_{\boldsymbol{\eta}_H} \in \mathbb{C}^{L^2\times L^2}$ and $\boldsymbol{R}_{\boldsymbol{y}} \in \mathbb{C}^{L\times L}$, of the random vectors $\boldsymbol{\eta}_H(\xi)$ and $\boldsymbol{y}(\xi)$, are defined by (4.10), and they are related by the equation $\boldsymbol{R}_{\boldsymbol{y}} = \boldsymbol{G}(\boldsymbol{c})\,\boldsymbol{R}_{\boldsymbol{\eta}_H}\,\boldsymbol{G}(\boldsymbol{c})^*$. This equation can be vectorized to obtain

$$\mathrm{vec}\,\boldsymbol{R}_{\boldsymbol{y}} = \left(\overline{\boldsymbol{G}(\boldsymbol{c})} \otimes \boldsymbol{G}(\boldsymbol{c})\right)\mathrm{vec}\,\boldsymbol{R}_{\boldsymbol{\eta}_H}\,, \tag{4.31}$$

which is an underdetermined linear system with the $L^2 \times L^4$ matrix $\overline{\boldsymbol{G}(\boldsymbol{c})} \otimes \boldsymbol{G}(\boldsymbol{c})$. So for (4.31) to be uniquely solvable, the covariance matrix $\boldsymbol{R}_{\boldsymbol{\eta}_H} \in \mathbb{C}^{L^2 \times L^2}$ has to be sparse with at most $L^2$ nonzero entries.

**Reduction of Variables** Assume that there is a set $\Lambda \subseteq \mathbb{Z}_L \times \mathbb{Z}_L$ such that $\eta_H(k, \ell; \xi) = 0$ a.e. $\xi$ for $(k, \ell) \notin \Lambda$. Then the linear system (4.30) reduces to

$$\boldsymbol{y}(\xi) = \boldsymbol{G}(\boldsymbol{c})|_\Lambda \, \boldsymbol{\eta}_{\boldsymbol{H},\Lambda}(\xi)\,, \tag{4.32}$$

where $\boldsymbol{\eta}_{\boldsymbol{H},\Lambda}(\xi) = \{\eta_H(k, \ell; \xi)\}_{(k,\ell)\in\Lambda}$ is the restriction of $\boldsymbol{\eta}_H(\xi)$ to the set $\Lambda$. Correspondingly, the linear system (4.31) reduces to

$$\operatorname{vec} \boldsymbol{R}_{\boldsymbol{y}} = \left(\overline{\boldsymbol{G}(\boldsymbol{c})} \otimes \boldsymbol{G}(\boldsymbol{c})\right)\big|_{\Lambda\times\Lambda} \operatorname{vec} \boldsymbol{R}_{\boldsymbol{H},\Lambda}, \tag{4.33}$$

which is a linear system with the $L^2 \times |\Lambda|^2$ matrix $\overline{\boldsymbol{G}(\boldsymbol{c})} \otimes \boldsymbol{G}(\boldsymbol{c})|_{\Lambda\times\Lambda}$ with the vectorization $\operatorname{vec} \boldsymbol{R}$ of a matrix $\boldsymbol{R} \in \mathbb{C}^{N\times N}$ as defined in (4.13). Note that for $\boldsymbol{R} \in \mathbb{C}^{N\times N}$,

$$\begin{aligned} \operatorname{supp} \boldsymbol{R} &= \{(m, n) \in \mathbb{Z}_N \times \mathbb{Z}_N \,:\, R(m, n) \neq 0\} \quad \subseteq \mathbb{Z}_N \times \mathbb{Z}_N, \\ \operatorname{supp}(\operatorname{vec} \boldsymbol{R}) &= \{\ell \in \mathbb{Z}_{N^2} \,:\, \operatorname{vec} X(\ell) \neq 0\} \quad \subseteq \mathbb{Z}_{N^2}. \end{aligned}$$

Through the vectorization, each support pattern $\Lambda$ in $\mathbb{Z}_N \times \mathbb{Z}_N$ is converted into a support pattern $\widetilde{\Lambda}$ in $\mathbb{Z}_{N^2}$, and vice versa. For brevity, we will often abuse notations and not distinguish the sets $\Lambda$ and $\widetilde{\Lambda}$.

Covariance matrices have a particular structure which should be reflected by the support pattern. Therefore, a set $\Lambda \subseteq \mathbb{Z}_N \times \mathbb{Z}_N$ is called a *positive semi-definite (psd) pattern* if

$$(i, j) \in \Lambda \quad \text{implies} \quad (i, i),\ (j, i),\ (j, j) \in \Lambda. \tag{4.34}$$

### *4.4.1 Permissible and Defective Support Patterns*

Motivated by Eq. (4.33), we consider matrices $\boldsymbol{G} \in \mathbb{C}^{M\times N}$ with $M \leq N$ and seek for support patterns $\Gamma \subseteq \mathbb{Z}_N \times \mathbb{Z}_N$ such that the matrix $\overline{\boldsymbol{G}} \otimes \boldsymbol{G}|_\Gamma$ is injective for some $\boldsymbol{G} \in \mathbb{C}^{M\times N}$. Note in particular that injectivity of the matrix $\overline{\boldsymbol{G}} \otimes \boldsymbol{G}|_\Gamma$ with $\Gamma = \Lambda \times \Lambda$ and $\boldsymbol{G} = \boldsymbol{G}(\boldsymbol{c})$ for some $\boldsymbol{c} \in \mathbb{C}^L$ would allow us to solve Eq. (4.33) uniquely in $\boldsymbol{R}_{\boldsymbol{H},\Lambda}$.

**Definition 4.9** For $M, N \in \mathbb{N}$ with $M \leq N$, a pattern $\Gamma \subseteq \mathbb{Z}_N \times \mathbb{Z}_N$ is called $(M, N)$-*defective* if for every $\boldsymbol{G} \in \mathbb{C}^{M\times N}$ the matrix $\overline{\boldsymbol{G}} \otimes \boldsymbol{G}|_\Gamma$ is not injective; otherwise, $\Gamma$ is called $(M, N)$-*permissible*.

We start with a lemma that provides several equivalent conditions for injectivity of $\overline{\boldsymbol{G}} \otimes \boldsymbol{G}|_\Gamma$ for general non-structured matrices $\boldsymbol{G} \in \mathbb{C}^{M\times N}$ and psd patterns $\Gamma \subseteq \mathbb{Z}_N \times \mathbb{Z}_N$.

**Lemma 4.6 (Lemma 10 in [27])** *Let $\boldsymbol{G} \in \mathbb{C}^{M\times N}$ with $M, N \in \mathbb{N}$, $M \leq N$, and let $\Gamma \subseteq \mathbb{Z}_N \times \mathbb{Z}_N$ be a psd pattern. The following are equivalent:*

*(a)* $\boldsymbol{X} \mapsto \boldsymbol{G}\boldsymbol{X}\boldsymbol{G}^*$ *is injective on the (nonlinear) cone* $\{\boldsymbol{X} \in \mathbb{C}^{N\times N} : \boldsymbol{X} \geq 0,\ supp\boldsymbol{X} \subseteq \Gamma\}$.
*(b)* $\boldsymbol{X} \mapsto \boldsymbol{G}\boldsymbol{X}\boldsymbol{G}^*$ *is injective on the subspace* $\{\boldsymbol{X} \in \mathbb{C}^{N\times N} : \boldsymbol{X}^* = \boldsymbol{X},\ supp\boldsymbol{X} \subseteq \Gamma\}$.
*(c)* $\boldsymbol{X} \mapsto \boldsymbol{G}\boldsymbol{X}\boldsymbol{G}^*$ *is injective on the subspace* $\{\boldsymbol{X} \in \mathbb{C}^{N\times N} : supp\boldsymbol{X} \subseteq \Gamma\}$.
*(d)* $\overline{\boldsymbol{G}} \otimes \boldsymbol{G}$ *is injective on the subspace* $\{w \in \mathbb{C}^{N^2} : suppw \subseteq \Gamma\}$, *that is,* $\overline{\boldsymbol{G}} \otimes \boldsymbol{G}|_\Gamma$ *is injective.*

In the case of tensor structured psd patterns, i.e., $\Gamma = \Lambda \times \Lambda$ with $\Lambda \subseteq \mathbb{Z}_N$, the injectivity of $\overline{\boldsymbol{G}} \otimes \boldsymbol{G}|_\Gamma$ is simply characterized by the injectivity of $\boldsymbol{G}|_\Lambda$.

**Proposition 4.7** *Let $\boldsymbol{G} \in \mathbb{C}^{M\times N}$ with $M, N \in \mathbb{N}$, $M \leq N$, and $\Lambda \subseteq \mathbb{Z}_N$ with $|\Lambda| \geq 1$. Let $diag(\Lambda) = \{(n, n) : n \in \Lambda\} \subset \mathbb{Z}_N \times \mathbb{Z}_N$. The following are equivalent:*

*(a) The matrix $\boldsymbol{G}|_\Lambda \in \mathbb{C}^{M\times|\Lambda|}$ is injective, i.e., the columns of $\boldsymbol{G}|_\Lambda$ are linearly independent.*
*(b) The matrix $\overline{\boldsymbol{G}} \otimes \boldsymbol{G}|_{\Lambda\times\Lambda} \in \mathbb{C}^{M^2\times|\Lambda|^2}$ is injective.*
*(c) There exist nonempty disjoint subsets $\Lambda_1, \Lambda_2 \subset \Lambda$ with $\Lambda_1 \cup \Lambda_2 = \Lambda$ such that $\overline{\boldsymbol{G}} \otimes \boldsymbol{G}|_{(\Lambda_1\times\Lambda_1)\cup(\Lambda_2\times\Lambda_2)}$ is injective.*
*(d) There exist nonempty disjoint subsets $\Lambda_1, \Lambda_2 \subset \Lambda$ with $\Lambda_1 \cup \Lambda_2 = \Lambda$ such that $\overline{\boldsymbol{G}} \otimes \boldsymbol{G}|_{(\Lambda_1\times\Lambda_2)\cup(\Lambda_2\times\Lambda_1)\cup diag(\Lambda)}$ is injective.*
*(e) There exists an element $n \in \Lambda$ such that $\overline{\boldsymbol{G}} \otimes \boldsymbol{G}|_{(\{n\}\times\Lambda)\cup(\Lambda\times\{n\})\cup diag(\Lambda)}$ is injective.*

*Moreover, in this case, $|\Lambda| \leq rk(\boldsymbol{G})$ $(\leq M)$ and $\overline{\boldsymbol{G}} \otimes \boldsymbol{G}|_\Gamma$ is injective for every $\Gamma \subseteq \Lambda \times \Lambda$.*

*Remark 4.4* All permissible patterns in (b)–(e) are psd patterns which are contained in $\Lambda \times \Lambda$.

We refer to section "Proof of Proposition 4.7" for a proof of Proposition 4.7. Note that injectivity of $\overline{\boldsymbol{G}} \otimes \boldsymbol{G}|_\Gamma$ for $\Gamma = \Lambda \times \Lambda$ and its subpatterns appearing in Proposition 4.7 $(b) - (e)$ depend only on the injectivity of $\boldsymbol{G}|_\Lambda$. This argument however does not apply to patterns that are more distributed in $\mathbb{Z}_N \times \mathbb{Z}_N$. An extreme case is the diagonal pattern $\Gamma = \text{diag} := \{(n, n) : n \in \mathbb{Z}_N\} \subset \mathbb{Z}_N \times \mathbb{Z}_N$, which will be discussed in Propositions 4.8 and 4.9 below.

As a direct consequence of Proposition 4.7, we also obtain some fundamental limitations on the patterns $\Gamma$ for which $\overline{\boldsymbol{G}} \otimes \boldsymbol{G}|_\Gamma$ is injective.

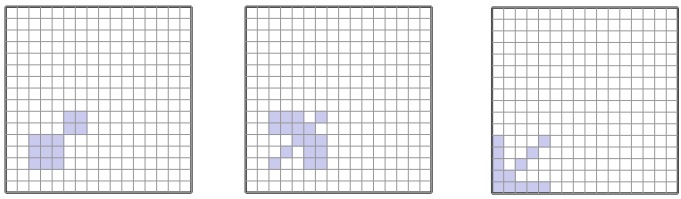



**Fig. 4.2** Examples of (4, 16)-defective patterns due to Corollary 4.3

**Corollary 4.3 (cf. Propositions 15 and 17 in [27])**
*Let $\boldsymbol{G} \in \mathbb{C}^{M\times N}$ with $M, N \in \mathbb{N}$ and $M \leq N$.*

*(i) Let $\Lambda_1, \Lambda_2 \subseteq \mathbb{Z}_N$ be disjoint sets. If $\boldsymbol{G}|_{\Lambda_1\cup\Lambda_2}$ is not injective (in particular, if $|\Lambda_1|+|\Lambda_2| > rk\boldsymbol{G}$), then $\overline{\boldsymbol{G}} \otimes \boldsymbol{G}|_{(\Lambda_1\times\Lambda_1)\cup(\Lambda_2\times\Lambda_2)}$ and $\overline{\boldsymbol{G}} \otimes \boldsymbol{G}|_{(\Lambda_1\times\Lambda_2)\cup(\Lambda_2\times\Lambda_1)\cup diag(\Lambda)}$ are not injective.*
*(ii) Let $\Lambda \subseteq \mathbb{Z}_N$. If $\boldsymbol{G}|_\Lambda$ is not injective (in particular, if $|\Lambda| > rk\boldsymbol{G}$), then for each $n \in \Lambda$ the matrix $\overline{\boldsymbol{G}} \otimes \boldsymbol{G}|_{(\{n\}\times\Lambda)\cup(\Lambda\times\{n\})\cup diag(\Lambda)}$ is not injective. Consequently, if a psd pattern $\Gamma \subseteq \mathbb{Z}_N\times\mathbb{Z}_N$ contains more than $rk(\boldsymbol{G})$ elements in a row/column, then $\overline{\boldsymbol{G}} \otimes \boldsymbol{G}|_\Gamma$ is not injective.*

Corollary 4.3 provides three types of defective patterns which are illustrated in Fig. 4.2. The defective patterns of the form $(\Lambda_1\times\Lambda_1)\cup(\Lambda_2\times\Lambda_2)$ and $(\Lambda_1\times\Lambda_2)\cup(\Lambda_2\times\Lambda_1)\cup \mathrm{diag}(\Lambda)$ are, respectively, called the *two squares* pattern and the *butterfly* pattern (see [26, Definition 5.1]). The defective patterns of the form $(\{n\}\times\Lambda)\cup(\Lambda\times\{n\})\cup\mathrm{diag}(\Lambda)$ with $n \in \Lambda$ are called the *arrow* (*arrowhead*) pattern.

Considering the channel model (4.30) in which Gabor matrices $\boldsymbol{G}(\boldsymbol{c}) \in \mathbb{C}^{L\times L^2}$ arise naturally, we will now restrict $\boldsymbol{G}$ to $L\times L^2$ dimensional matrices. The following result (whose proof is given in section "Proof of Proposition 4.8") provides some necessary conditions for injectivity of $\overline{\boldsymbol{G}} \otimes \boldsymbol{G}|_{\mathrm{diag}}$ with $\boldsymbol{G} \in \mathbb{C}^{L\times L^2}$.

**Proposition 4.8** *Let $\boldsymbol{G} = [G_{j,k}]_{j=0\ k=0}^{L-1\ L^2-1} \in \mathbb{C}^{L\times L^2}$ with $L \geq 2$. The matrix $\overline{\boldsymbol{G}} \otimes \boldsymbol{G}|_{diag} \in \mathbb{C}^{L^2\times L^2}$ is singular if one of the following conditions holds:*

*(i) There exist two rows of $\boldsymbol{G}$ that are linearly dependent when the entrywise phase factors are removed, that is, there exist some $j_1 \neq j_2$ and a constant $\eta \geq 0$ satisfying $|G_{j_1,k}| = \eta\,|G_{j_2,k}|$ for all $k$.*
*(ii) There exist two rows of $\boldsymbol{G}$ such that all entries in each row have the same magnitude, that is, there exist some $j_1 \neq j_2$ and constants $r_{j_1}, r_{j_2} \geq 0$ satisfying $|G_{j_1,k}| = r_{j_1}$ and $|G_{j_2,k}| = r_{j_2}$ for all $k$.*
*(iii) The matrix $\boldsymbol{G}$ has all real-valued entries or all imaginary-valued entries, that is, the entries $G_{j,k}$ are all real or all imaginary.*

In the case of Gabor matrices $\boldsymbol{G} = \boldsymbol{G}(\boldsymbol{c})$ with $\boldsymbol{c} \in \mathbb{C}^L$, we have the following criterion for invertibility of $\overline{\boldsymbol{G}} \otimes \boldsymbol{G}|_{\text{diag}} \in \mathbb{C}^{L^2 \times L^2}$ given in terms of the short-time Fourier transform $V_{\boldsymbol{c}}\boldsymbol{c}$.

**Proposition 4.9 (Theorem 16 in [27] and Theorem II.2 in [28])** *Let diag* $= \{(\lambda, \lambda) : \lambda \in \mathbb{Z}_L \times \mathbb{Z}_L\}$ *be the diagonal pattern on the* $L^2 \times L^2$ *grid. Then* $\boldsymbol{\Phi}(\boldsymbol{c}) := \overline{\boldsymbol{G}(\boldsymbol{c})} \otimes \boldsymbol{G}(\boldsymbol{c})|_{diag} \in \mathbb{C}^{L^2 \times L^2}$ *is invertible if and only if* $\boldsymbol{c}$ *is in the set*

$$V_L = \{c \in \mathbb{C}^L \ : \ V_{\boldsymbol{c}}\boldsymbol{c} \text{ has no zero entries}\},$$

*which is a dense open subset of* $\mathbb{C}^L$ *with full measure. Moreover, the singular values of* $\boldsymbol{\Phi}(\boldsymbol{c})$ *are given by*

$$\sqrt{L} \cdot |V_{\boldsymbol{c}}\boldsymbol{c}(k, \ell)| = \sqrt{L} \cdot |\langle c, \pi(k, \ell)\boldsymbol{c}\rangle|, \quad k, \ell = 0, \ldots, L-1.$$

*Remark 4.5*

(a) If $\boldsymbol{c} = (c_0, \ldots, c_{L-1})^{\mathbb{T}} \in \mathbb{C}^L$ with $|c_0| = \ldots = |c_{L-1}| > 0$, then all entries in $\boldsymbol{G}(\boldsymbol{c})$ are identical in magnitude, and therefore $\boldsymbol{\Phi}(\boldsymbol{c})$ must be singular by Proposition 4.8. This is indeed confirmed with Proposition 4.9 by observing that $V_{\boldsymbol{c}}\boldsymbol{c}(0, \ell) = \langle \boldsymbol{c}, \boldsymbol{M}^{\ell}\boldsymbol{c}\rangle = 0$ for all $\ell \in \mathbb{Z}_L \backslash \{0\}$.

(b) The columns of $\boldsymbol{\Phi}(\boldsymbol{c})$ cannot form an orthogonal basis for $\mathbb{C}^{L^2}$. Indeed, the columns of $\boldsymbol{\Phi}(\boldsymbol{c})$ being orthogonal would imply that for any $(k', \ell') \neq (k, \ell)$,

$$\begin{aligned} |\langle \boldsymbol{c}, \pi(k' - k, \ell' - \ell)\boldsymbol{c}\rangle|^2 &= |\langle \pi(k, \ell)\boldsymbol{c}, \pi(k', \ell')\boldsymbol{c}\rangle|^2 \\ &= \langle \overline{\pi(k, \ell)\boldsymbol{c}} \otimes \pi(k, \ell)\boldsymbol{c}, \overline{\pi(k', \ell')\boldsymbol{c}} \otimes \pi(k', \ell')\boldsymbol{c}\rangle = 0, \end{aligned}$$

while the invertibility of $\boldsymbol{\Phi}(\boldsymbol{c})$ would imply $|\langle \boldsymbol{c}, \pi(k, \ell)\boldsymbol{c}\rangle| \neq 0$ for all $(k, \ell) \in \mathbb{Z}_L \times \mathbb{Z}_L$ by Proposition 4.9, yielding a contradiction.

(c) It is easily seen that $V_L$ is not a subset of the set $U_L$ in Proposition 4.1, and vice versa. For instance, $\boldsymbol{c} = (1, e^{\pi i/4})^{\mathbb{T}}$ belongs to $U_2 \backslash V_2$, while $\boldsymbol{c} = (3, -e^{2\pi i/3}, -2e^{4\pi i/3})^{\mathbb{T}}$ belongs to $V_3 \backslash U_3$. Furthermore, the set $U_L \cap V_L$ is a dense open subset of $\mathbb{C}^L$ with full measure.

From Propositions 4.7 and 4.9, we conclude that all $L \times L$ tensor patterns and the diagonal pattern are $(L, L^2)$-permissible (see Fig. 4.3 for an illustration of these patterns).

### 4.4.2 Linear Constraints in Stochastic Setting

A special type of stochastic channel operators, which has important applications in engineering and physics, is the so-called WSSUS channels.

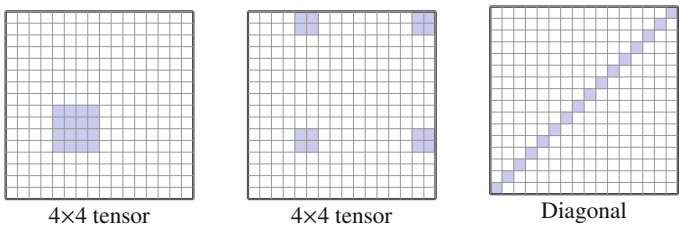



**Fig. 4.3** Examples of (4, 16)-permissible patterns: tensor structured psd patterns (Proposition 4.7) and the diagonal pattern (Proposition 4.9)

**Definition 4.10** A stochastic SISO channel $\boldsymbol{H}$ is called *wide-sense stationary with uncorrelated scattering* (WSSUS) if its covariance matrix $\boldsymbol{R}_{\boldsymbol{\eta}_{\boldsymbol{H}}}$ is diagonal, that is, $\boldsymbol{R}_{\boldsymbol{\eta}_{\boldsymbol{H}}}(\lambda, \lambda') = 0$ for all $\lambda \neq \lambda'$ in $\mathbb{Z}_L \times \mathbb{Z}_L$. In this case, we have $\boldsymbol{R}_{\boldsymbol{\eta}_{\boldsymbol{H}}}(\lambda, \lambda') = \delta_{\lambda,\lambda'} C_{\boldsymbol{\eta}_{\boldsymbol{H}}}(\lambda)$ for some $\boldsymbol{C}_{\boldsymbol{\eta}_{\boldsymbol{H}}} = \{C_{\boldsymbol{\eta}_{\boldsymbol{H}}}(\lambda)\}_{\lambda \in \mathbb{Z}_L \times \mathbb{Z}_L} \in \mathbb{C}^{L^2}$ which is called the *scattering function* of $\boldsymbol{H}$.

For WSSUS channels, the linear system (4.31) reduces to

$$\operatorname{vec} \boldsymbol{R}_{\boldsymbol{y}} = \left(\overline{\boldsymbol{G}(\boldsymbol{c})} \otimes \boldsymbol{G}(\boldsymbol{c})|_{\text{diag}}\right) \boldsymbol{C}_{\boldsymbol{\eta}_{\boldsymbol{H}}}, \tag{4.35}$$

which is an exactly determined linear system associated with the $L^2 \times L^2$ matrix $\overline{\boldsymbol{G}(\boldsymbol{c})} \otimes \boldsymbol{G}(\boldsymbol{c})|_{\text{diag}}$. Proposition 4.9 guarantees that this linear system is uniquely solvable for all $\boldsymbol{c}$ in the set $V_L$, which is a dense open subset of $\mathbb{C}^L$ with full measure. Hence, the class of WSSUS channels is identifiable up to second-order statistics in the sense of Definition 4.5.

#### 4.4.2.1 WSSUS Pattern with Additional Off-diagonal Contributions

Let us now weaken the WSSUS condition in Definition 4.10 to allow for some additional off-diagonal contributions, i.e., we assume that the covariance matrix $\boldsymbol{R}_{\boldsymbol{\eta}_{\boldsymbol{H}}} \in \mathbb{C}^{L^2 \times L^2}$ satisfies

$$\boldsymbol{R}_{\boldsymbol{\eta}_{\boldsymbol{H}}}(\lambda, \lambda') = \delta_{\lambda,\lambda'} C_{\boldsymbol{\eta}_{\boldsymbol{H}}}(\lambda) + \text{extra off-diagonal components.}$$

Note that $\boldsymbol{R}_{\boldsymbol{\eta}_H}$ is Hermitian and satisfies (4.11). Therefore, $\boldsymbol{R}_{\boldsymbol{\eta}_H}(\lambda, \lambda') \neq 0$ implies $\boldsymbol{R}_{\boldsymbol{\eta}_H}(\lambda', \lambda) \neq 0$ and so we will always have an even number of off-diagonal elements. Moreover, since the $\left|\text{supp}\boldsymbol{R}_{\boldsymbol{\eta}_H}\right| > L$, it is clear that such a channel will not be identifiable up to second-order statistics. Only if $\boldsymbol{R}_{\boldsymbol{\eta}_H}$ is known to satisfy additional side constraints, it might become identifiable and it is clear that one needs at least as many side-constraints as $R_{\boldsymbol{\eta}_H}$ has nonzero off-diagonal elements.

The following proposition shows that two (symmetric) nonzero off-diagonal elements in the covariance matrix $\boldsymbol{R}_{\eta_H}$ can always be compensated by two linear side constraints that are chosen appropriately.

**Proposition 4.10** *Let $\lambda, \lambda' \in \mathbb{Z}_L \times \mathbb{Z}_L$ with $\lambda \neq \lambda'$, and let $\boldsymbol{a} \in \mathbb{R}^{L^2}$, $b_1, b_2 \in \mathbb{R}$ be such that the vectors $(\boldsymbol{a}^{\mathbb{T}}, b_1, b_2)$ and $(\boldsymbol{a}^{\mathbb{T}}, b_1, b_2)$ are linearly independent, i.e.,*

$$\boldsymbol{\Psi}_0 = \begin{bmatrix} \boldsymbol{a}^{\mathbb{T}} & b_1 & b_2 \\ \boldsymbol{a}^{\mathbb{T}} & b_2 & b_1 \end{bmatrix} \in \mathbb{C}^{2\times(L^2+2)} \quad \textit{has full row rank.} \tag{4.36}$$

*There exists a vector $\boldsymbol{c} \in \mathbb{C}^L$ such that the $(L^2+2)\times(L^2+2)$ matrix*

$$\boldsymbol{\Psi} = \begin{bmatrix} \overline{\boldsymbol{G}(\boldsymbol{c})} \otimes \boldsymbol{G}(\boldsymbol{c})|_{diag} & \overline{\pi(\lambda)\boldsymbol{c}} \otimes \pi(\lambda')\boldsymbol{c} & \overline{\pi(\lambda')\boldsymbol{c}} \otimes \pi(\lambda)\boldsymbol{c} \\ \boldsymbol{a}^{\mathbb{T}} & b_1 & b_2 \\ \boldsymbol{a}^{\mathbb{T}} & b_2 & b_1 \end{bmatrix} \tag{4.37}$$

*is invertible. Moreover, the set of all such vectors $\boldsymbol{c} \in \mathbb{C}^L$ is a dense open subset of $\mathbb{C}^L$ with full measure.*

Proposition 4.10 shows that for almost WSSUS channels, one has a very similar behavior as in Theorem 4.1 for deterministic channels with side constraints. It shows that one additional (symmetric) support component of the covariance can always be compensated by an additional linear side constraint. A proof of Proposition 4.10 is given in section "Proof of Proposition 4.10".

We would like to mention that the *complex-valued* case $\boldsymbol{a} \in \mathbb{C}^{L^2}$, $b_1, b_2 \in \mathbb{C}$ is not fully resolved, as we are missing a rigorous proof for the determinant of $\boldsymbol{\Psi}$ being a *nontrivial* polynomial in the variables $c_0, \ldots, c_{L-1}$ and its complex conjugates $\overline{c_0}, \ldots, \overline{c_{L-1}}$. A rigorous argument would require similar techniques as in [13, 18]. Precisely, we wish to prove following statement: let $\lambda, \lambda' \in \mathbb{Z}_L \times \mathbb{Z}_L$ with $\lambda \neq \lambda'$, and let $\boldsymbol{a} \in \mathbb{C}^{L^2}$, $b_1, b_2 \in \mathbb{C}$ be such that

$$\boldsymbol{\Psi}_0 = \begin{bmatrix} \boldsymbol{a}^{\mathbb{T}} & b_1 & b_2 \\ \overline{\boldsymbol{a}}^{\mathbb{T}} & \overline{b_2} & \overline{b_1} \end{bmatrix} \in \mathbb{C}^{2\times(L^2+2)} \quad \text{has full row rank.}$$

There exists a vector $\boldsymbol{c} \in \mathbb{C}^L$ such that the $(L^2+2)\times(L^2+2)$ matrix

$$\boldsymbol{\Psi} = \begin{bmatrix} \overline{\boldsymbol{G}(\boldsymbol{c})} \otimes \boldsymbol{G}(\boldsymbol{c})|_{\text{diag}} & \overline{\pi(\lambda)\boldsymbol{c}} \otimes \pi(\lambda')\boldsymbol{c} & \overline{\pi(\lambda')\boldsymbol{c}} \otimes \pi(\lambda)\boldsymbol{c} \\ \boldsymbol{a}^{\mathbb{T}} & b_1 & b_2 \\ \overline{\boldsymbol{a}}^{\mathbb{T}} & \overline{b_2} & \overline{b_1} \end{bmatrix}$$

is invertible. Moreover, the set of all such vectors $\boldsymbol{c} \in \mathbb{C}^L$ is a dense open subset of $\mathbb{C}^L$ with full measure.

#### 4.4.2.2 Tensor Product Pattern with Additional Contributions

We now assume that the covariance matrix $\boldsymbol{R}_{\boldsymbol{\eta}_H} \in \mathbb{C}^{L^2 \times L^2}$ is supported in a tensor product pattern with some additional contributions, that is, $\text{supp}\boldsymbol{R}_{\boldsymbol{\eta}_H} \subset (\Lambda \times \Lambda) \cup \Omega$ with $\Lambda \subset \mathbb{Z}_L \times \mathbb{Z}_L$, $|\Lambda| = L$, and $\Omega \subset (\mathbb{Z}_L \times \mathbb{Z}_L)^2 \backslash (\Lambda \times \Lambda)$.

The case where $\Omega = \{(\lambda, \lambda)\}$ with $\lambda \in (\mathbb{Z}_L \times \mathbb{Z}_L) \backslash \Lambda$ is resolved by the following proposition. Note that if $\boldsymbol{c}$ is chosen from the set $U_L$ in Proposition 4.1 (so that $\boldsymbol{G}(\boldsymbol{c})$ has full spark), then $\boldsymbol{G}(\boldsymbol{c})|_\Lambda \in \mathbb{C}^{L \times L}$ is invertible and thus $\overline{\boldsymbol{G}(\boldsymbol{c})} \otimes \boldsymbol{G}(\boldsymbol{c})|_{\Lambda \times \Lambda} = \overline{\boldsymbol{G}(\boldsymbol{c})|_\Lambda} \otimes \boldsymbol{G}(\boldsymbol{c})|_\Lambda \in \mathbb{C}^{L^2 \times L^2}$ is invertible.

**Proposition 4.11** *Given a subset $\Lambda \subset \mathbb{Z}_L \times \mathbb{Z}_L$ with $|\Lambda| = L$, an element $\lambda \in (\mathbb{Z}_L \times \mathbb{Z}_L) \backslash \Lambda$, and any $\boldsymbol{a} \in \mathbb{C}^{L^2}$, $b \in \mathbb{C}$ with $\|\boldsymbol{a}\|_2^2 + |b|^2 \neq 0$, there exists a vector $\boldsymbol{c} \in \mathbb{C}^L$ such that the matrix*

$$\boldsymbol{\Psi} = \begin{bmatrix} \overline{\boldsymbol{G}(\boldsymbol{c})} \otimes \boldsymbol{G}(\boldsymbol{c})|_{\Lambda \times \Lambda} & \overline{\pi(\lambda)\boldsymbol{c}} \otimes \pi(\lambda)\boldsymbol{c} \\ \boldsymbol{a}^{\mathbb{T}} & b \end{bmatrix} \in \mathbb{C}^{(L^2+1) \times (L^2+1)}$$

*is invertible. Moreover, the set of all such vectors $\boldsymbol{c} \in \mathbb{C}^L$ is a dense open subset of $\mathbb{C}^L$ with full measure.*

A proof of Proposition 4.11 is given in section "Proof of Proposition 4.11". The problem of extending Proposition 4.11 to the general case $\Omega = \{(\lambda_1, \lambda_1), \ldots, (\lambda_K, \lambda_K)\}$ with $K \geq 2$ distinct elements $\lambda_1, \ldots, \lambda_K$ in $(\mathbb{Z}_L \times \mathbb{Z}_L) \backslash \Lambda$ is left open.

As a final remark, we note that most of the results in Sect. 4.4 are for stochastic SISO channels. While some results extend directly to stochastic MIMO channels (for instance, Proposition 4.9 can be extended immediately to WSSUS MIMO channels), the others require a careful modification and more involved proofs. We leave the investigation on stochastic MIMO channels as a future work.

### *4.4.3 Numerical Simulations*

In Sect. 4.4.2, we have shown that the matrices $\boldsymbol{\Psi}$ in Propositions 4.10 and 4.11 are invertible if the generating vector $\boldsymbol{c} \in \mathbb{C}^L$ belongs in a certain open dense subset of $\mathbb{C}^L$ with full measure. This indicates that the matrix $\boldsymbol{\Psi}$ with a randomly generated window $\boldsymbol{c} \in \mathbb{C}^L$ and randomly generated additional rows must be invertible with high probability. Although we have only treated some particular cases of patterns with additional contributions in Propositions 4.10 and 4.11, we believe that these results extend to general cases with more additional contributions. To verify this claim, we made some numerical experiments described below.

As in Propositions 4.10 and 4.11, we generated matrices of the form

$$\boldsymbol{\Psi} = \begin{pmatrix} \overline{\mathbf{G}(\mathbf{c})} \otimes \mathbf{G}(\mathbf{c})|_{\Gamma} & \overline{\mathbf{G}(\mathbf{c})} \otimes \mathbf{G}(\mathbf{c})|_{\Delta} \\ \mathbf{A} & \mathbf{B} \end{pmatrix}, \tag{4.38}$$

where $\Gamma \subset (\mathbb{Z}_L \times \mathbb{Z}_L)^2$ with $|\Gamma| = L^2$ is a permissible support pattern in the sense of Definition 4.9, and $\Delta \subset (\mathbb{Z}_L \times \mathbb{Z}_L)^2 \backslash \Gamma$ with $|\Delta| = N$ is the support of additional $N$ contributions chosen so that $\Gamma \dot{\cup} \Delta$ is a psd pattern (i.e., satisfies (4.34)). The matrices $\boldsymbol{A} \in \mathbb{C}^{N \times L^2}$ and $\boldsymbol{B} \in \mathbb{C}^{N \times N}$ represent additional side constraints, and the full matrix $\boldsymbol{\Psi}$ is therefore of dimension $(L^2+N) \times (L^2+N)$. In the simulations, we considered the two cases: (i) $\Gamma$ is the diagonal set (WSSUS) and (ii) $\Gamma$ is a random tensor product set.

(i) First, when $\Gamma$ is the diagonal set, i.e., $\Gamma = \{(\lambda, \lambda) : \lambda \in \mathbb{Z}_L \times \mathbb{Z}_L\}$, we picked $\Delta \subset (\mathbb{Z}_L \times \mathbb{Z}_L)^2 \backslash \Gamma$ in a way that $\Gamma \dot{\cup} \Delta$ is a psd pattern and then generated 20 pilot vectors $\mathbf{c} \in \mathbb{C}^L$ with each vector chosen uniformly from the $L$-dimensional complex unit ball. For each $\mathbf{c}$, we generated 20 pairs of $\boldsymbol{A}$ and $\boldsymbol{B}$. The matrix $\mathbf{A}$ is generated by choosing its rows uniformly from the $L^2$-dimensional complex unit ball, and the matrix $\mathbf{B}$ is generated by choosing its entries uniformly from the set $\{x + iy : -1 \leq x, y \leq 1\}$ independently. Thus, for each dimension $L$, we generated 400 instances of $\boldsymbol{\Psi}$ and computed the smallest singular value of each matrix $\boldsymbol{\Psi}$. The average of the smallest singular value is shown in Fig. 4.4.

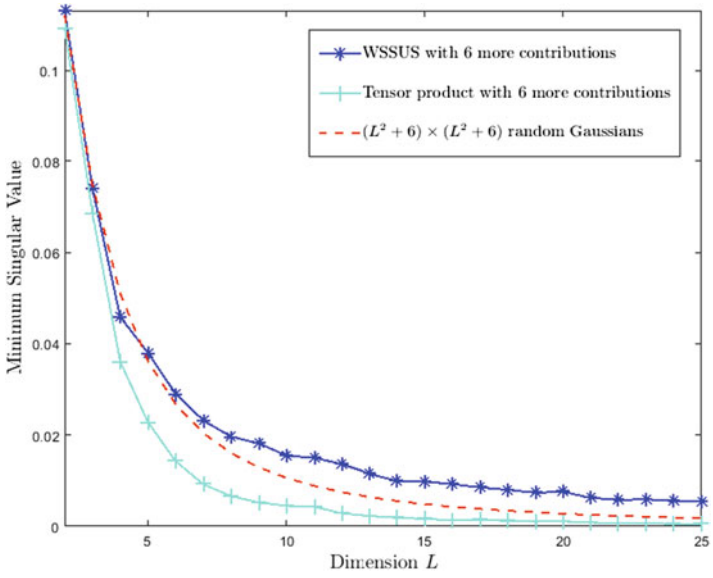



**Fig. 4.4** Illustrated is the empirical expected value of the minimal singular value of the matrix $\boldsymbol{\Psi}$ in (4.38) for different support sets and with different number of side constraints. Additionally, the theoretical lower bound on the expectation of the minimal singular value of squared matrices [3] with i.i.d. Gaussian random variable entries with zero mean and variance one is depicted

(ii) Next, we generated a tensor product set $\Gamma \subset (\mathbb{Z}_L \times \mathbb{Z}_L)^2$ in a random way and then picked $\Delta \subset (\mathbb{Z}_L \times \mathbb{Z}_L)^2 \backslash \Gamma$ so that $\Gamma \dot{\cup} \Delta$ is a psd pattern. For each generated psd pattern, we chose $\boldsymbol{c}$, $\boldsymbol{A}$, and $\boldsymbol{B}$ similarly as before and generated 400 instances of the matrix $\boldsymbol{\Psi}$. The smallest singular value of each matrix is computed and their average is shown in Fig. 4.4.

In Fig. 4.4, the described two cases with $|\Delta| = N = 6$ are compared with the expected smallest singular value of $(L^2+N)\times(L^2+N)$ random Gaussian matrices (whose entries are i.i.d. Gaussian random variables with mean zero and variance one) [3].

Figure 4.4 indicates that the matrix $\boldsymbol{\Psi}$ is always invertible for generic vectors $\boldsymbol{c}$ and matrices $\boldsymbol{A}$ and $\boldsymbol{B}$. The comparison of minimum singular values implies that recovery for the WSSUS case is more stable than the tensor product case. Certainly, the minimum singular value of $\boldsymbol{\Psi}$ decreases as its dimension grows, that is, as $L$ grows and as more additional support components are considered. It is interesting to note that the minimum singular value for the WSSUS case is larger than Gaussian matrices, while that for the tensor product case is smaller than Gaussian matrices. Note also that the decay rate for the WSSUS case is much slower than Gaussian matrices and the tensor product case.

Based on the experiments above, one could expect that matrices $\boldsymbol{\Psi}$ of the form (4.38) are almost always invertible in practical scenarios, and hence the knowledge of linear side constraints would help to overcome the fundamental limitation due to degrees of freedom when recovering the covariance matrix $\boldsymbol{R}_{\boldsymbol{\eta}_H} \in \mathbb{C}^{L^2 \times L^2}$ from $\boldsymbol{R}_{\boldsymbol{y}} \in \mathbb{C}^{L \times L}$ in Eq. (4.31).

## Appendix

### *Proof of Lemma 4.3*

Let us label the elements of $\Lambda$ by $\Lambda = \{(k_1, \ell_1), \ldots, (k_R, \ell_R)\} \subset \mathbb{Z}_L \times \mathbb{Z}_L$ and fix any $\boldsymbol{d} \in U_L$ so that $\boldsymbol{G}(\boldsymbol{d})$ has full spark. Then $\boldsymbol{G}(\boldsymbol{d})|_\Lambda \in \mathbb{C}^{L \times R}$ has the full rank $R \leq L-1$, and therefore $\ker\,(\boldsymbol{G}(\boldsymbol{c})|_\Lambda)^* \subset \mathbb{C}^L$ is an $L - R \geq 1$-dimensional subspace of $\mathbb{C}^L$. Fix any nontrivial vector, $\boldsymbol{x} \in \ker\,(\boldsymbol{G}(\boldsymbol{c})|_\Lambda)^*$.

For $0 \leq p, q \leq L-1$, the (non-)commutation relation $\boldsymbol{M}^\ell \boldsymbol{T}^k = \omega^{k\ell}\, \boldsymbol{T}^k \boldsymbol{M}^\ell$ with $\omega = e^{2\pi i/L}$ gives

$$\boldsymbol{M}^\ell \boldsymbol{T}^k (\boldsymbol{M}^q \boldsymbol{T}^p \boldsymbol{d}) = \omega^{\ell p - kq}\, \boldsymbol{M}^q \boldsymbol{T}^p (\boldsymbol{M}^\ell \boldsymbol{T}^k \boldsymbol{d}), \quad k, \ell = 0, \ldots, L-1,$$

and since $\boldsymbol{G}(\boldsymbol{d})$ has full spark, it follows that $\boldsymbol{G}(\boldsymbol{M}^q \boldsymbol{T}^p \boldsymbol{d})$ has full spark as well, that is, $\boldsymbol{M}^q \boldsymbol{T}^p \boldsymbol{d} \in U_L$. Collecting the equation for $(k, \ell) \in \Lambda$, we have

$$\boldsymbol{G}(\boldsymbol{M}^q \boldsymbol{T}^p \boldsymbol{d})|_\Lambda = \boldsymbol{M}^q \boldsymbol{T}^p\, \boldsymbol{G}(\boldsymbol{d})|_\Lambda\, \boldsymbol{D}^{(p,q)}$$

with $\boldsymbol{D}^{(p,q)} := \mathrm{diag}(\omega^{\ell_1 p - k_1 q}, \ldots, \omega^{\ell_R p - k_R q}) \in \mathbb{C}^{R\times R}$, so that

$$\big(\boldsymbol{G}(\boldsymbol{M}^q\boldsymbol{T}^p\boldsymbol{d})|_\Lambda\big)^* = \overline{\boldsymbol{D}^{(p,q)}}\,\big(\boldsymbol{G}(\boldsymbol{d})|_\Lambda\big)^*\,\boldsymbol{M}^{-q}\boldsymbol{T}^{-p}.$$

Since $\boldsymbol{x} \in \ker\,(\boldsymbol{G}(\boldsymbol{c})|_\Lambda)^*$, it follows that for $p, q = 0, \ldots, L-1$,

$$\boldsymbol{M}^q\boldsymbol{T}^p\boldsymbol{x} \in \ker\,(\boldsymbol{G}(\boldsymbol{M}^q\boldsymbol{T}^p\boldsymbol{d})|_\Lambda)^* \quad \text{where} \quad \boldsymbol{M}^q\boldsymbol{T}^p\boldsymbol{d} \in U_L.$$

To prove that $\mathrm{span}\big\{\ker\,(\boldsymbol{G}(\boldsymbol{c})|_\Lambda)^* : \boldsymbol{c} \in U_L\big\} = \mathbb{C}^L$, it suffices to show that $\mathrm{span}\{\boldsymbol{M}^q\boldsymbol{T}^p\boldsymbol{x} : p, q = 0, \ldots, L-1\} = \mathbb{C}^L$. But this is always true since $\{\boldsymbol{M}^q\boldsymbol{T}^p\boldsymbol{z} : p, q = 0, \ldots, L-1\}$ with any nontrivial vector $\boldsymbol{z} \in \mathbb{C}^L$ is a tight frame for $\mathbb{C}^L$ with frame bound $L\,\|\boldsymbol{z}\|_2^2$.

## *Proof of Proposition 4.6*

We first assume that $s \in \{0, 1, \ldots, L-1\}$, and write $\Lambda = \{(k_1, k_1 s), \ldots, (k_N, k_N s)\}$ for some $0 \le k_1 < \ldots < k_N \le L-1$. Let $\mathcal{Z} = \mathrm{span}\{\boldsymbol{D}^s\boldsymbol{v}_{j_1}, \ldots, \boldsymbol{D}^s\boldsymbol{v}_{j_R}\}$ for some $0 \le j_1 < \ldots < j_R \le L-1$, where $N + R \le L$, and we label the elements in $\mathbb{Z}_L\backslash\{j_1, \ldots, j_R\}$ as $j_{R+1} < \ldots < j_L$.

For a message vector $\boldsymbol{u} = \{u_r\}_{r=1}^R$ of length $R$, we set $\boldsymbol{z} = \sum_{r=1}^R u_r\,\boldsymbol{D}^s\boldsymbol{v}_{j_r}$ and $\boldsymbol{c} = \sum_{r=R+1}^L \boldsymbol{D}^s\boldsymbol{v}_{j_r}$. Then for $\boldsymbol{H} = \sum_{n=1}^N a_{k_n}\boldsymbol{M}^{k_n s}\boldsymbol{T}^{k_n} \in OPW(\Lambda)$, we have

$$\boldsymbol{y} = \boldsymbol{H}(\boldsymbol{z}+\boldsymbol{c}) = \sum_{n=1}^N a_{k_n}\boldsymbol{M}^{k_n s}\boldsymbol{T}^{k_n}\Big(\sum_{r=1}^R u_r\,\boldsymbol{D}^s\boldsymbol{v}_{j_r} + \sum_{r=R+1}^L \boldsymbol{D}^s\boldsymbol{v}_{j_r}\Big) \tag{4.39}$$

$$= \sum_{r=1}^R u_r\Big(\sum_{n=1}^N a_{k_n}\omega^{-j_r k_n + \frac{k_n(k_n-1)s}{2}}\Big)\boldsymbol{D}^s\boldsymbol{v}_{j_r} + \sum_{r=R+1}^L \Big(\sum_{n=1}^N a_{k_n}\omega^{-j_r k_n + \frac{k_n(k_n-1)s}{2}}\Big)\boldsymbol{D}^s\boldsymbol{v}_{j_r}.$$

Since the vectors $\boldsymbol{D}^s\boldsymbol{v}_j$, $j = 0, \ldots, L-1$, form an orthonormal basis of $\mathbb{C}^L$, we immediately obtain the basis representation coefficients in (4.39) by taking the inner product of $\boldsymbol{y}$ with $\boldsymbol{D}^s\boldsymbol{v}_j$, $j = 0, \ldots, L-1$:

$$\langle \boldsymbol{y}, \boldsymbol{D}^s\boldsymbol{v}_{j_r}\rangle =: b_r = \begin{cases} u_r\Big(\sum_{n=1}^N a_{k_n}\,\omega^{-j_r k_n + \frac{k_n(k_n-1)s}{2}}\Big) & \text{for } r = 1, \ldots, R, \\ \sum_{n=1}^N a_{k_n}\,\omega^{-j_r k_n + \frac{k_n(k_n-1)s}{2}} & \text{for } r = R{+}1, \ldots, L. \end{cases} \tag{4.40}$$

Note that since $N \le L - R$, the linear system

$$\begin{bmatrix} b_{R+1} \\ \vdots \\ b_L \end{bmatrix} = \begin{bmatrix} \omega^{-j_{R+1}k_1} & \dots & \omega^{-j_{R+1}k_N} \\ \vdots & & \\ \omega^{-j_L k_1} & \dots & \omega^{-j_L k_N} \end{bmatrix} \begin{bmatrix} a_{k_1}\omega^{\frac{k_1(k_1-1)s}{2}} \\ \vdots \\ a_{k_N}\omega^{\frac{k_N(k_N-1)s}{2}} \end{bmatrix}$$

is a (over-)determined system. In fact, the coefficients $a_{k_1}, \dots, a_{k_N}$ can be recovered uniquely due to Chebotarev's theorem on roots of unity (see, for instance, [29] and [30, Lemma 1.3]), which asserts that for $L \in \mathbb{N}$ prime, every square submatrix of the $L \times L$ discrete Fourier matrix $(e^{-2\pi i k\ell/L})_{k,\ell=0}^{L-1}$ formed by eliminating arbitrary rows and columns is invertible. Once the coefficients $a_{k_1}, \dots, a_{k_N}$ are recovered, it is straightforward to compute the message values $u_1, \dots, u_R$ from (4.40), provided that $\sum_{n=1}^{N} a_{k_n}\, \omega^{-j_r k_n + \frac{k_n(k_n-1)s}{2}} \neq 0$ for $r = 1, \dots, R$.

Now we assume that $s = \infty$, meaning that $\Lambda = \{(0, \ell_1), \dots, (0, \ell_N)\}$ for some $0 \le \ell_1 < \dots < \ell_N \le L-1$. Let $\mathcal{Z} = \operatorname{span}\{\boldsymbol{e}_{j_1}, \dots, \boldsymbol{e}_{j_R}\}$ for some $0 \le j_1 < \dots < j_R \le L-1$, where $N + R \le L$, and we label the elements in $\mathbb{Z}_L \backslash \{j_1, \dots, j_R\}$ as $j_{R+1} < \dots < j_L$.

For a message vector $\boldsymbol{u} = \{u_r\}_{r=1}^{R}$ of length $R$, we set $\boldsymbol{z} = \sum_{r=1}^{R} u_r\, \boldsymbol{e}_{j_r}$ and $\boldsymbol{c} = \sum_{r=R+1}^{L} \boldsymbol{e}_j$. Then for $\boldsymbol{H} = \sum_{n=1}^{N} a_{\ell_n} \boldsymbol{M}^{\ell_n} \in OPW(\Lambda)$, we have

$$\boldsymbol{y} = \boldsymbol{H}(\boldsymbol{z} + \boldsymbol{c}) = \sum_{r=1}^{R} u_r \Big( \sum_{n=1}^{N} a_{\ell_n}\, \omega^{j_r \ell_n} \Big) \boldsymbol{e}_{j_r} + \sum_{r=R+1}^{L} \Big( \sum_{n=1}^{N} a_{\ell_n}\, \omega^{j_r \ell_n} \Big) \boldsymbol{e}_{j_r}.$$

The rest of the proof is similar to the case $s \in \{0, 1, \dots, L-1\}$.

## *Proof of Proposition 4.7*

(a) $\Leftrightarrow$ (b): Note that $\operatorname{rk}(\overline{\boldsymbol{G}} \otimes \boldsymbol{G}|_{\Lambda\times\Lambda}) = \operatorname{rk}(\overline{\boldsymbol{G}}|_{\Lambda} \otimes \boldsymbol{G}|_{\Lambda}) = (\operatorname{rk} \boldsymbol{G}|_{\Lambda})^2$. If $\boldsymbol{G}|_{\Lambda}$ is injective, i.e., if $\operatorname{rk} \boldsymbol{G}|_{\Lambda} = |\Lambda| \le M$, then $\operatorname{rk}(\overline{\boldsymbol{G}} \otimes \boldsymbol{G}|_{\Lambda\times\Lambda}) = |\Lambda|^2$ and hence $\overline{\boldsymbol{G}} \otimes \boldsymbol{G}|_{\Lambda\times\Lambda}$ is injective. Conversely, assume that $\overline{\boldsymbol{G}} \otimes \boldsymbol{G}|_{\Lambda\times\Lambda}$ is injective. If a vector $\boldsymbol{v} \in \mathbb{C}^N$ with $\operatorname{supp}\boldsymbol{v} \subseteq \Lambda$ satisfies $\boldsymbol{G}\boldsymbol{v} = 0$, then $\boldsymbol{G}\boldsymbol{v}\boldsymbol{v}^*\boldsymbol{G}^* = \boldsymbol{G}\boldsymbol{v}(\boldsymbol{G}\boldsymbol{v})^* = 0$ and since $\operatorname{supp}(\boldsymbol{v}\boldsymbol{v}^*) \subseteq \Lambda\times\Lambda$, we have from Lemma 4.6 that $\boldsymbol{v}\boldsymbol{v}^* = 0$ which implies $\boldsymbol{v} = 0$. Hence, $\boldsymbol{G}|_{\Lambda}$ is injective.

(b) $\Rightarrow$ (c), (d), (e): These implications are obvious, since the index sets $(\Lambda_1\times\Lambda_1) \cup (\Lambda_2\times\Lambda_2)$, $(\Lambda_1\times\Lambda_2) \cup (\Lambda_2\times\Lambda_1) \cup \operatorname{diag}(\Lambda)$, $(\{n\}\times\Lambda) \cup (\Lambda\times\{n\}) \cup \operatorname{diag}(\Lambda)$, with $\Lambda_1 \subseteq \Lambda$ and $n \in \Lambda$, are subsets of $\Lambda\times\Lambda$.

(c) $\Rightarrow$ (a): Suppose to the contrary that $\boldsymbol{v} \in \mathbb{C}^N$ is a nontrivial vector with $\operatorname{supp}\boldsymbol{v} \subseteq \Lambda$ and $\boldsymbol{G}\boldsymbol{v} = 0$. Let $\Lambda_1, \Lambda_2 \subset \Lambda$ be nonempty disjoint sets such that $\Lambda_1 \cup \Lambda_2 = \Lambda$ and $\overline{\boldsymbol{G}} \otimes \boldsymbol{G}|_{(\Lambda_1\times\Lambda_1)\cup(\Lambda_2\times\Lambda_2)}$ is injective. We write $0 \neq \boldsymbol{v} = \boldsymbol{v}_1 + \boldsymbol{v}_2$ with $\operatorname{supp}\boldsymbol{v}_1 \subseteq \Lambda_1$ and $\operatorname{supp}\boldsymbol{v}_2 \subseteq \Lambda_2$. Then $\boldsymbol{G}\boldsymbol{v}_1 = \boldsymbol{G}(-\boldsymbol{v}_2)$, which implies $\boldsymbol{G}\boldsymbol{v}_1\boldsymbol{v}_1^*\boldsymbol{G}^* = \boldsymbol{G}\boldsymbol{v}_2\boldsymbol{v}_2^*\boldsymbol{G}^*$ and thus $\boldsymbol{G}\boldsymbol{Z}\boldsymbol{G}^* = 0$, where $\boldsymbol{Z} = \boldsymbol{v}_1\boldsymbol{v}_1^* - \boldsymbol{v}_2\boldsymbol{v}_2^* \in \mathbb{C}^{N\times N}$ is a nontrivial (Hermitian) matrix supported in $(\Lambda_1\times\Lambda_1) \cup (\Lambda_2\times\Lambda_2)$. The matrix

$\boldsymbol{Z}$ is nontrivial because at least one of its diagonal entries is nonzero. However, in view of Lemma 4.6, this contradicts with the injectivity of $\overline{\boldsymbol{G}} \otimes \boldsymbol{G}|_{(\Lambda_1 \times \Lambda_1) \cup (\Lambda_2 \times \Lambda_2)}$. Therefore, such a vector $\boldsymbol{v} \in \mathbb{C}^N$ must not exist which means that $\boldsymbol{G}|_\Lambda$ is injective.

(e) $\Rightarrow$ (d): If $n \in \Lambda$, then $(n, n) \in \mathrm{diag}(\Lambda)$, so that $(\{n\} \times \Lambda) \cup (\Lambda \times \{n\}) \cup \mathrm{diag}(\Lambda) = (\{n\} \times \Lambda \backslash \{n\}) \cup (\Lambda \backslash \{n\} \times \{n\}) \cup \mathrm{diag}(\Lambda)$. Therefore, the condition (e) implies (d) with $\Lambda_1 = \{n\}$ and $\Lambda_2 = \Lambda \backslash \{n\}$.

(d) $\Rightarrow$ (a): Suppose to the contrary that $\boldsymbol{v} \in \mathbb{C}^N$ is a nontrivial vector with $\mathrm{supp}\boldsymbol{v} \subseteq \Lambda$ and $\boldsymbol{G}\boldsymbol{v} = 0$. Let $\Lambda_1, \Lambda_2 \subset \Lambda$ be nonempty disjoint sets such that $\Lambda_1 \cup \Lambda_2 = \Lambda$ and $\overline{\boldsymbol{G}} \otimes \boldsymbol{G}|_{(\Lambda_1 \times \Lambda_2) \cup (\Lambda_2 \times \Lambda_1) \cup \mathrm{diag}(\Lambda)}$ is injective. As before, we write $0 \neq \boldsymbol{v} = \boldsymbol{v}_1 + \boldsymbol{v}_2$ with $\mathrm{supp}\boldsymbol{v}_1 \subseteq \Lambda_1$ and $\mathrm{supp}\boldsymbol{v}_2 \subseteq \Lambda_2$ and consider the following three cases:

(i) If $\boldsymbol{v}_1 \neq 0$ and $\boldsymbol{v}_2 = 0$, then choose any $n_2 \in \Lambda_2$ and set

$$\boldsymbol{A} = [0 \ldots 0 \underbrace{\boldsymbol{v}_1}_{n_2\text{-th}} 0 \ldots 0] \quad \in \mathbb{C}^{N \times N}.$$

Then $\boldsymbol{Z} = \boldsymbol{A} + \boldsymbol{A}^* \in \mathbb{C}^{N \times N}$ is a nontrivial (Hermitian) matrix supported in $(\Lambda_1 \times \{n_2\}) \cup (\{n_2\} \times \Lambda_1)$ satisfying $\boldsymbol{G}\boldsymbol{Z}\boldsymbol{G}^* = 0$.

(ii) If $\boldsymbol{v}_1 = 0$ and $\boldsymbol{v}_2 \neq 0$, then choose any $n_1 \in \Lambda_1$ and set

$$\boldsymbol{B} = [0 \ldots 0 \underbrace{\boldsymbol{v}_2}_{n_1\text{-th}} 0 \ldots 0] \quad \in \mathbb{C}^{N \times N}.$$

Then $\boldsymbol{Z} = \boldsymbol{B} + \boldsymbol{B}^* \in \mathbb{C}^{N \times N}$ is a nontrivial (Hermitian) matrix supported in $(\Lambda_2 \times \{n_1\}) \cup (\{n_1\} \times \Lambda_2)$ satisfying $\boldsymbol{G}\boldsymbol{Z}\boldsymbol{G}^* = 0$.

(iii) Otherwise, if both $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ are nontrivial, then $\boldsymbol{G}\boldsymbol{v}_1 = \boldsymbol{G}(-\boldsymbol{v}_2)$ which implies $\boldsymbol{G}\boldsymbol{v}_1\boldsymbol{v}_2^*\boldsymbol{G}^* = \boldsymbol{G}\boldsymbol{v}_2\boldsymbol{v}_1^*\boldsymbol{G}^*$, and thus we have $\boldsymbol{G}\boldsymbol{Z}\boldsymbol{G}^* = 0$ where $\boldsymbol{Z} = \boldsymbol{v}_1\boldsymbol{v}_2^* - \boldsymbol{v}_2\boldsymbol{v}_1^* \in \mathbb{C}^{N \times N}$ is a nontrivial (Hermitian) matrix supported in $(\Lambda_1 \times \Lambda_2) \cup (\Lambda_2 \times \Lambda_1)$.

In all three cases, we deduce from Lemma 4.6 that $\overline{\boldsymbol{G}} \otimes \boldsymbol{G}|_{(\Lambda_1 \times \Lambda_2) \cup (\Lambda_2 \times \Lambda_1)}$ is not injective, yielding a contradiction. Therefore, such a vector $\boldsymbol{v} \in \mathbb{C}^N$ must not exist which means that $\boldsymbol{G}|_\Lambda$ is injective.

## *Proof of Proposition 4.8*

For each $k = 1, \ldots, L^2$, let $\boldsymbol{v}_k = \{v_k(j)\}_{j=0}^{L-1} = \{G_{j,k}\}_{j=0}^{L-1} \in \mathbb{C}^L$ be the $k$-th column vector of $\boldsymbol{G}$. Note that $\overline{\boldsymbol{G}} \otimes \boldsymbol{G}|_{\mathrm{diag}} = \{\overline{\boldsymbol{v}_k} \otimes \boldsymbol{v}_k\}_{k=1}^{L^2} = \{\boldsymbol{v}_k\boldsymbol{v}_k^*\}_{k=1}^{L^2} \in \mathbb{C}^{L^2 \times L^2}$ is not injective if and only if there exists a nontrivial vector $\boldsymbol{a} = \{a_k\}_{k=1}^{L^2} \in \mathbb{C}^{L^2}$ satisfying

$$\sum_{k=1}^{L^2} a_k \, \boldsymbol{v}_k \boldsymbol{v}_k^* = 0, \tag{4.41}$$

which can be understood as a linear system with $L^2$ equations in the $L^2$ variables $a_1, \ldots, a_{L^2}$. Note that the $L$ equations reflecting the diagonal entries of the square matrices $\boldsymbol{v}_k \boldsymbol{v}_k^*$ are of the form

$$\sum_{k=1}^{L^2} a_k \cdot \left| v_k(j) \right|^2 = 0, \quad j = 0, \ldots, L-1. \tag{4.42}$$

(i) If there exist some $j_1 \neq j_2$ and a constant $\eta \geq 0$ satisfying $|G_{j_1,k}| = \eta \, |G_{j_2,k}|$ for all $k$, that is, $|v_k(j_1)| = \eta \, |v_k(j_2)|$ for all $k$, then Eq. (4.42) for $j = j_1$ and $j = j_2$ is identical up to a multiplicative factor. The linear system (4.41) is then underdetermined, so there exists a nontrivial vector $\boldsymbol{a} = \{a_k\}_{k=1}^{L^2}$ satisfying (4.41).

(ii) Assume that there exist some $j_1 \neq j_2$ and constants $r_{j_1}, r_{j_2} \geq 0$ satisfying $|G_{j_1,k}| = |v_k(j_1)| = r_{j_1}$ and $|G_{j_2,k}| = |v_k(j_2)| = r_{j_2}$ for all $k$. Equation (4.42) for $j = j_1$ and $j = j_2$ is then given by $r_{j_1} \sum_{k=1}^{L^2} a_k = 0$ and $r_{j_2} \sum_{k=1}^{L^2} a_k = 0$, respectively. Note that for each $n = 1, 2$, the equation $r_{j_n} \sum_{k=1}^{L^2} a_k = 0$ is void if $r_{j_n} = 0$ and reduces to $\sum_{k=1}^{L^2} a_k = 0$ if $r_{j_n} \neq 0$. In any case, the linear system (4.41) is underdetermined, so there exists a nontrivial vector $\boldsymbol{a} = \{a_k\}_{k=1}^{L^2}$ satisfying (4.41).

(iii) Assume that all entries of $\boldsymbol{G}$ are real-valued. Then (4.41) is simply

$$\sum_{k=1}^{L^2} a_k \, \boldsymbol{v}_k \boldsymbol{v}_k^{\mathbb{T}} = 0. \tag{4.43}$$

Since all matrices $\boldsymbol{v}_k \boldsymbol{v}_k^{\mathbb{T}} \in \mathbb{R}^{L \times L}$ are symmetric, the equation read off from the $(p, q)$-th entry of (4.43) is identical to that read off from the $(q, p)$-th entry of (4.43). This implies that the linear system (4.43) has kernel of dimension at least $L(L-1)/2 \geq 1$, so there exists a nontrivial vector $\boldsymbol{a} = \{a_k\}_{k=1}^{L^2}$ satisfying (4.43).

Now, assume that all entries of $\boldsymbol{G}$ are imaginary-valued. Writing the column vectors of $\boldsymbol{G}$ as $\boldsymbol{v}_k = i \boldsymbol{w}_k$ with $\boldsymbol{w}_k \in \mathbb{R}^L$ for $k = 1, \ldots, L^2$, Eq. (4.41) becomes

$$-\sum_{k=1}^{L^2} a_k \, \boldsymbol{w}_k \boldsymbol{w}_k^{\mathbb{T}} = 0. \tag{4.44}$$

By the same arguments as above, there exists a nontrivial vector $\boldsymbol{a} = \{a_k\}_{k=1}^{L^2}$ satisfying (4.44).

Hence, in all cases (i)–(iii), the matrix $\overline{\boldsymbol{G}} \otimes \boldsymbol{G}|_{\text{diag}} \in \mathbb{C}^{L^2 \times L^2}$ is singular.

## *Proof of Proposition 4.10*

To prove Proposition 4.10, we need the following technical lemma.

**Lemma D.7** *Let* $\boldsymbol{c} \in \mathbb{C}^L$ *be such that* $\boldsymbol{\Phi}(\boldsymbol{c}) := \overline{\boldsymbol{G}(\boldsymbol{c})} \otimes \boldsymbol{G}(\boldsymbol{c})|_{diag} \in \mathbb{C}^{L^2 \times L^2}$ *is invertible. Fix any* $\lambda \neq \lambda'$ *in* $\mathbb{Z}_L \times \mathbb{Z}_L$. *If* $\overline{\pi(\lambda)\boldsymbol{c}} \otimes \pi(\lambda')\boldsymbol{c} = \boldsymbol{\Phi}(\boldsymbol{c})\, \boldsymbol{v}$ *for some* $\boldsymbol{v} \in \mathbb{C}^{L^2}$, *then* $\overline{\pi(\lambda')\boldsymbol{c}} \otimes \pi(\lambda)\boldsymbol{c} = \boldsymbol{\Phi}(\boldsymbol{c})\, \overline{v}$.

***Proof*** Since $\boldsymbol{\Phi}(\boldsymbol{c})$ is invertible, there exist unique vectors $\boldsymbol{v}, \boldsymbol{w} \in \mathbb{C}^{L^2}$ such that $\overline{\pi(\lambda)\boldsymbol{c}} \otimes \pi(\lambda')\boldsymbol{c} = \boldsymbol{\Phi}(\boldsymbol{c})\, \boldsymbol{v}$ and $\overline{\pi(\lambda')\boldsymbol{c}} \otimes \pi(\lambda)\boldsymbol{c} = \boldsymbol{\Phi}(\boldsymbol{c})\, \boldsymbol{w}$. For any $(m, n) \in \mathbb{Z}_L \times \mathbb{Z}_L$, we have

$$\begin{aligned}
&\left(\overline{\pi(m,n)\boldsymbol{c}} \otimes \pi(m,n)\boldsymbol{c}\right)^* \overline{\pi(\lambda)\boldsymbol{c}} \otimes \pi(\lambda')\boldsymbol{c} = \langle \pi(m,n)\boldsymbol{c}, \pi(\lambda)\boldsymbol{c}\rangle \cdot \langle \pi(\lambda')\boldsymbol{c}, \pi(m,n)\boldsymbol{c}\rangle \\
&= \overline{\langle \pi(m,n)\boldsymbol{c}, \pi(\lambda')\boldsymbol{c}\rangle \cdot \langle \pi(\lambda)\boldsymbol{c}, \pi(m,n)\boldsymbol{c}\rangle} = \overline{\left(\overline{\pi(m,n)\boldsymbol{c}} \otimes \pi(m,n)\boldsymbol{c}\right)^* \overline{\pi(\lambda')\boldsymbol{c}} \otimes \pi(\lambda)\boldsymbol{c}}
\end{aligned}$$

so that

$$\boldsymbol{\Phi}(\boldsymbol{c})^*\left(\overline{\pi(\lambda)\boldsymbol{c}} \otimes \pi(\lambda')\boldsymbol{c}\right) = \overline{\boldsymbol{\Phi}(\boldsymbol{c})^*\left(\overline{\pi(\lambda')\boldsymbol{c}} \otimes \pi(\lambda)\boldsymbol{c}\right)},$$

and thus, $\boldsymbol{\Phi}(\boldsymbol{c})^*\boldsymbol{\Phi}(\boldsymbol{c})\, \boldsymbol{v} = \overline{\boldsymbol{\Phi}(\boldsymbol{c})^*\boldsymbol{\Phi}(\boldsymbol{c})\, \boldsymbol{w}}$. However, the Gram matrix of $\boldsymbol{\Phi}(\boldsymbol{c})$ given by

$$\begin{aligned}
\boldsymbol{\Phi}(c)^*\boldsymbol{\Phi}(c) &= \left[\left(\overline{\pi(m,n)\boldsymbol{c}} \otimes \pi(m,n)\boldsymbol{c}\right)^* \overline{\pi(k,\ell)\boldsymbol{c}} \otimes \pi(k,\ell)\boldsymbol{c}\right]_{(m,n),(k,\ell)\in\mathbb{Z}_L\times\mathbb{Z}_L} \\
&= \left[|\langle \pi(m,n)\boldsymbol{c}, \pi(k,\ell)\boldsymbol{c}\rangle|^2\right]_{(m,n),(k,\ell)\in\mathbb{Z}_L\times\mathbb{Z}_L}
\end{aligned}$$

is positive definite and has real-valued entries, so we obtain $\boldsymbol{v} = \overline{\boldsymbol{w}}$, that is, $\boldsymbol{w} = \overline{\boldsymbol{v}}$. □

***Proof of Proposition 4.10*** If $\boldsymbol{a} = 0$, then condition (4.36) is equivalent to having $b_1^2 \neq b_2^2$, and thus

$$\det \boldsymbol{\Psi} = \det\left(\overline{\boldsymbol{G}(\boldsymbol{c})} \otimes \boldsymbol{G}(\boldsymbol{c})|_{\text{diag}}\right) \cdot (b_1^2 - b_2^2).$$

Hence, for any vector $\boldsymbol{c}$ in the set $V_L$ from Proposition 4.9, we have that $\boldsymbol{\Psi}$ is invertible.

Now assume that $\boldsymbol{a} \neq 0$. Then by a Gaussian elimination, we see that condition (4.36) is equivalent to $b_1 \neq b_2$. With $\boldsymbol{c} \in \mathbb{C}^L$ chosen from the set $V_L$ in Proposition 4.9, and $\boldsymbol{v} = \{v(j)\}_{j=1}^{L^2} \in \mathbb{C}^{L^2}$ satisfying $\overline{\pi(\lambda)\boldsymbol{c}} \otimes \pi(\lambda')\boldsymbol{c} = \boldsymbol{\Phi}(\boldsymbol{c})\, \boldsymbol{v}$, we have

$$\boldsymbol{\Psi} \sim \begin{bmatrix} \boldsymbol{\Phi}(\boldsymbol{c}) & 0 & 0 \\ \boldsymbol{a}^{\mathbb{T}} & b_1 + b_2 - \sum_{j=1}^{L^2} a(j)\left(v(j) + \overline{v(j)}\right) & b_2 - \sum_{j=1}^{L^2} a(j)\,\overline{v(j)} \\ 0 & 0 & b_1 - b_2 \end{bmatrix},$$

which implies that $\boldsymbol{\Psi}$ is invertible if and only if

$$b_1 + b_2 \neq \sum_{j=1}^{L^2} a(j)\left(v(j) + \overline{v(j)}\right). \tag{4.45}$$

Here, the right-hand side can be written as

$$\sum_{j=1}^{L^2} a(j)\left(v(j) + \overline{v(j)}\right) = 2\,\mathrm{Re}\left(\boldsymbol{a}^{\mathbb{T}}\left(\overline{\boldsymbol{G}(\boldsymbol{c})} \otimes \boldsymbol{G}(\boldsymbol{c})|_{\mathrm{diag}}\right)^{-1}\left(\overline{\pi(\lambda)\boldsymbol{c}} \otimes \pi(\lambda')\boldsymbol{c}\right)\right).$$

It is therefore enough to show that there is a vector $\boldsymbol{c} \in V_L$ satisfying (4.45). Suppose that $\boldsymbol{d} \in V_L$ is a vector satisfying

$$b_1 + b_2 = 2\,\mathrm{Re}\left(\boldsymbol{a}^{\mathbb{T}}\left(\overline{G(\boldsymbol{d})} \otimes G(\boldsymbol{d})|_{\mathrm{diag}}\right)^{-1}\left(\overline{\pi(\lambda)\boldsymbol{d}} \otimes \pi(\lambda')\boldsymbol{d}\right)\right) \tag{4.46}$$

and

$$\boldsymbol{a}^{\mathbb{T}}\left(\overline{G(\boldsymbol{d})} \otimes G(\boldsymbol{d})|_{\mathrm{diag}}\right)^{-1}\left(\overline{\pi(\lambda)\boldsymbol{d}} \otimes \pi(\lambda')\boldsymbol{d}\right) \neq 0. \tag{4.47}$$

Using the (non-)commutation relation $\boldsymbol{M}^{\ell}\boldsymbol{T}^{k} = \omega^{k\ell}\boldsymbol{T}^{k}\boldsymbol{M}^{\ell}$ with $\omega = e^{2\pi i/L}$, we have

$$\pi(k,\ell)\boldsymbol{M}^{q}\boldsymbol{T}^{p}\boldsymbol{d} = \boldsymbol{M}^{q}\boldsymbol{T}^{p}\pi(k,\ell)\boldsymbol{d}\cdot\omega^{\ell p - kq} \quad \text{for } k,\ell,p,q \in \mathbb{Z}_L.$$

Writing $\lambda = (k,\ell)$ and $\lambda' = (k',\ell')$ in $\mathbb{Z}_L \times \mathbb{Z}_L$, we thus obtain that for any $p, q \in \mathbb{Z}_L$,

$$\begin{aligned} &\left(\overline{G(\boldsymbol{M}^{q}\boldsymbol{T}^{p}\boldsymbol{d})} \otimes G(\boldsymbol{M}^{q}\boldsymbol{T}^{p}\boldsymbol{d})|_{\mathrm{diag}}\right)^{-1} \overline{\pi(k,\ell)\boldsymbol{M}^{q}\boldsymbol{T}^{p}\boldsymbol{d}} \otimes \pi(k',\ell')\boldsymbol{M}^{q}\boldsymbol{T}^{p}\boldsymbol{d} \\ &= \left(\overline{G(\boldsymbol{d})} \otimes G(\boldsymbol{d})|_{\mathrm{diag}}\right)^{-1} \overline{\pi(k,\ell)\boldsymbol{d}} \otimes \pi(k',\ell')\boldsymbol{d}\cdot\omega^{(\ell'-\ell)p-(k'-k)q}. \end{aligned}$$

Since $\lambda = (k,\ell)$ and $\lambda' = (k',\ell')$ are distinct in $\mathbb{Z}_L \times \mathbb{Z}_L$, there is an element $(p,q) \in \mathbb{Z}_L \times \mathbb{Z}_L$ with $(\ell'-\ell)p-(k'-k)q \not\equiv 0 \mod L$, that is, $\omega^{(\ell'-\ell)p-(k'-k)q} \neq 1$. It is then easily seen that condition (4.45) holds for $\boldsymbol{c} = \boldsymbol{M}^{q}\boldsymbol{T}^{p}\boldsymbol{d} \in V_L$. Hence, we conclude that there exists a vector $\boldsymbol{c} \in V_L$ such that $\boldsymbol{\Psi}$ is invertible. □

## *Proof of Proposition 4.11*

Note that for each $\boldsymbol{c} \in U_L$, the matrix $\boldsymbol{G}(\boldsymbol{c}) \in \mathbb{C}^{L\times L^2}$ has full spark, i.e., $\mathrm{spark}(\boldsymbol{G}(\boldsymbol{c})) = L+1$ (see Definition 4.3), so its submatrix $\boldsymbol{G}(\boldsymbol{c})|_\Lambda \in \mathbb{C}^{L\times L}$ is invertible. The vector $\boldsymbol{z_c} := (\boldsymbol{G}(\boldsymbol{c})|_\Lambda)^{-1}\pi(\lambda)\boldsymbol{c} \in \mathbb{C}^L$ has no zero entries, since otherwise the equation $\boldsymbol{G}(\boldsymbol{c})|_{\Lambda\cup\{\lambda\}}\left[\begin{smallmatrix}\boldsymbol{z_c}\\1\end{smallmatrix}\right] = (\boldsymbol{G}(\boldsymbol{c})|_\Lambda)\,\boldsymbol{z_c} - \pi(\lambda)\boldsymbol{c} = 0$ would hold with $\left\|\left[\begin{smallmatrix}\boldsymbol{z_c}\\1\end{smallmatrix}\right]\right\|_0 \le L$ implying that $\mathrm{spark}(\boldsymbol{G}(\boldsymbol{c})) \le L$.

We will also need the following observation. Fix any $\boldsymbol{c} \in U_L$. Since $\boldsymbol{z_c} \in \mathbb{C}^L$ has no zero entries, the set $\{\boldsymbol{d} \in \mathbb{C}^L : (\boldsymbol{G}(\boldsymbol{d})|_\Lambda)\,\boldsymbol{z_c} = \pi(\lambda)\boldsymbol{d}\}$ is a zero measure manifold in $\mathbb{C}^L$ containing $\boldsymbol{c}$. Pick any vector $\boldsymbol{c}_1 \in \mathbb{C}^L$ outside the manifold; then clearly $\boldsymbol{z}_{\boldsymbol{c}_1} \neq \boldsymbol{z_c}$. Again, the set $\{\boldsymbol{d} \in \mathbb{C}^L : (\boldsymbol{G}(\boldsymbol{d})|_\Lambda)\,\boldsymbol{z}_{\boldsymbol{c}_1} = \pi(\lambda)\boldsymbol{d}\}$ is a zero measure manifold in $\mathbb{C}^L$ containing $\boldsymbol{c}_1$. Pick any vector $\boldsymbol{c}_2 \in \mathbb{C}^L$ outside the two manifolds; then clearly $\boldsymbol{z}_{\boldsymbol{c}_2} \neq \boldsymbol{z_c}, \boldsymbol{z}_{\boldsymbol{c}_1}$. Inductively, we obtain a sequence of vectors $\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots$ in $U_L$ such that $\boldsymbol{z_c}, \boldsymbol{z}_{\boldsymbol{c}_1}, \boldsymbol{z}_{\boldsymbol{c}_2}, \ldots$ are all distinct vectors in $\mathbb{C}^L$. Since each $\boldsymbol{z}_{\boldsymbol{c}_n} \in \mathbb{C}^L$ has no zero entries, it follows that the vectors $\overline{\boldsymbol{z_c}} \otimes \boldsymbol{z_c}, \overline{\boldsymbol{z}_{\boldsymbol{c}_1}} \otimes \boldsymbol{z}_{\boldsymbol{c}_1}, \overline{\boldsymbol{z}_{\boldsymbol{c}_2}} \otimes \boldsymbol{z}_{\boldsymbol{c}_2}, \ldots$ are all distinct in $\mathbb{C}^{L^2}$. In turn, the vectors $(\overline{\boldsymbol{z}_{\boldsymbol{c}_n}} \otimes \boldsymbol{z}_{\boldsymbol{c}_n} - \overline{\boldsymbol{z_c}} \otimes \boldsymbol{z_c}) \neq 0$ for $n = 1, 2, \ldots$ are all distinct in $\mathbb{C}^{L^2}$.

Noting that $\overline{\pi(\lambda)\boldsymbol{c}} \otimes \pi(\lambda)\boldsymbol{c} = \overline{(\boldsymbol{G}(\boldsymbol{c})|_\Lambda)\,\boldsymbol{z_c}} \otimes (\boldsymbol{G}(\boldsymbol{c})|_\Lambda)\,\boldsymbol{z_c} = \big(\overline{\boldsymbol{G}(\boldsymbol{c})|_\Lambda} \otimes \boldsymbol{G}(\boldsymbol{c})|_\Lambda\big)(\overline{\boldsymbol{z_c}} \otimes \boldsymbol{z_c}) = \big(\overline{\boldsymbol{G}(\boldsymbol{c})} \otimes \boldsymbol{G}(\boldsymbol{c})|_{\Lambda\times\Lambda}\big)(\overline{\boldsymbol{z_c}} \otimes \boldsymbol{z_c})$, we apply a column operation on $\boldsymbol{\Psi}$ to obtain

$$\boldsymbol{\Psi} \sim \begin{bmatrix} \overline{\boldsymbol{G}(\boldsymbol{c})} \otimes \boldsymbol{G}(\boldsymbol{c})|_{\Lambda\times\Lambda} & 0 \\ \boldsymbol{a}^{\mathbb{T}} & b - \boldsymbol{a}^{\mathbb{T}}(\overline{\boldsymbol{z_c}} \otimes \boldsymbol{z_c}) \end{bmatrix},$$

which gives

$$\begin{aligned} \det\boldsymbol{\Psi} &= \det\big(\overline{\boldsymbol{G}(\boldsymbol{c})} \otimes \boldsymbol{G}(\boldsymbol{c})|_{\Lambda\times\Lambda}\big)\cdot\big(b - \boldsymbol{a}^{\mathbb{T}}(\overline{\boldsymbol{z_c}} \otimes \boldsymbol{z_c})\big) \\ &= \underbrace{\big|\det(\boldsymbol{G}(\boldsymbol{c})|_\Lambda)\big|^{2L}}_{\neq 0}\cdot\big(b - \boldsymbol{a}^{\mathbb{T}}(\overline{\boldsymbol{z_c}} \otimes \boldsymbol{z_c})\big). \end{aligned}$$

**Step 1** (*Existence of a vector* $\boldsymbol{c} \in U_L$ *such that* $\det\boldsymbol{\Psi} \neq 0$).

We only need to show that there exists a vector $\boldsymbol{c} \in U_L$ satisfying $\boldsymbol{a}^{\mathbb{T}}(\overline{\boldsymbol{z_c}} \otimes \boldsymbol{z_c}) \neq b$.

(i) If $\boldsymbol{a} = 0$, then $b \neq 0$ by the assumption, so $\boldsymbol{a}^{\mathbb{T}}(\overline{\boldsymbol{z_c}} \otimes \boldsymbol{z_c}) = 0 \neq b$ for all $\boldsymbol{c} \in U_L$.

(ii) If $\boldsymbol{a} \neq 0$, then the set $\{\boldsymbol{x} \in \mathbb{C}^{L^2} : b - \boldsymbol{a}^{\mathbb{T}}\boldsymbol{x} = 0\}$ is an affine hyperplane in $\mathbb{C}^{L^2}$. We need to show that there is a vector $\boldsymbol{c} \in U_L$ such that $\overline{\boldsymbol{z_c}} \otimes \boldsymbol{z_c}$ does not belong in this affine hyperplane. Suppose to the contrary that $\{\overline{\boldsymbol{z_c}} \otimes \boldsymbol{z_c} : \boldsymbol{c} \in U_L\}$ is contained in the affine hyperplane. Since $b - \boldsymbol{a}^{\mathbb{T}}\boldsymbol{x} = \langle(\boldsymbol{x}, 1), (-\overline{\boldsymbol{a}}, \overline{b})\rangle_{\mathbb{C}^{L+1}}$ for $\boldsymbol{x} \in \mathbb{C}^{L^2}$, this means that the nonzero vector $(-\overline{\boldsymbol{a}}, \overline{b}) \in \mathbb{C}^{L+1}$ is orthogonal to $\left[\begin{smallmatrix}\overline{\boldsymbol{z_c}}\otimes\boldsymbol{z_c}\\1\end{smallmatrix}\right]$ for all $\boldsymbol{c} \in U_L$; thus, showing

$$\operatorname{span}\left\{\begin{bmatrix}\overline{z_{\boldsymbol{c}}}\otimes z_{\boldsymbol{c}}\\ 1\end{bmatrix} : \boldsymbol{c}\in U_L\right\} = \mathbb{C}^{L^2+1} \tag{4.48}$$

would give a desired contradiction.

By Gaussian elimination, we see that (4.48) holds if there is a vector $\boldsymbol{c} \in U_L$ satisfying

$$\operatorname{span}\left\{\overline{z_{\boldsymbol{c}'}}\otimes z_{\boldsymbol{c}'} - \overline{z_{\boldsymbol{c}}}\otimes z_{\boldsymbol{c}} : \boldsymbol{c}'\in U_L\right\} = \mathbb{C}^{L^2}.$$

The left-hand side is equal to

$$\begin{aligned}
&\operatorname{span}\left\{\overline{z_{\boldsymbol{c}'}}\otimes z_{\boldsymbol{c}'} - \overline{z_{\boldsymbol{c}}}\otimes z_{\boldsymbol{c}},\ \overline{z_{\boldsymbol{c}''}}\otimes z_{\boldsymbol{c}''} - \overline{z_{\boldsymbol{c}}}\otimes z_{\boldsymbol{c}} : \boldsymbol{c}',\boldsymbol{c}''\in U_L\right\}\\
&= \operatorname{span}\left\{\overline{z_{\boldsymbol{c}'}}\otimes z_{\boldsymbol{c}'} - \overline{z_{\boldsymbol{c}''}}\otimes z_{\boldsymbol{c}''},\ \overline{z_{\boldsymbol{c}''}}\otimes z_{\boldsymbol{c}''} - \overline{z_{\boldsymbol{c}}}\otimes z_{\boldsymbol{c}} : \boldsymbol{c}',\boldsymbol{c}''\in U_L\right\}\\
&\supset \operatorname{span}\left\{\overline{z_{\boldsymbol{c}'}}\otimes z_{\boldsymbol{c}'} - \overline{z_{\boldsymbol{c}''}}\otimes z_{\boldsymbol{c}''} : \boldsymbol{c}',\boldsymbol{c}''\in U_L\right\}\\
&\supset \operatorname{span}\left\{\overline{z_{\boldsymbol{c}'}}\otimes z_{\boldsymbol{c}'} - \overline{z_{\boldsymbol{c}}}\otimes z_{\boldsymbol{c}} : \boldsymbol{c}'\in U_L\right\},
\end{aligned}$$

so in order to prove (4.48), it suffices to show that

$$\operatorname{span}\left\{\overline{z_{\boldsymbol{c}'}}\otimes z_{\boldsymbol{c}'} - \overline{z_{\boldsymbol{c}''}}\otimes z_{\boldsymbol{c}''} : \boldsymbol{c}',\boldsymbol{c}''\in U_L\right\} = \mathbb{C}^{L}.$$

But this can be seen easily because $U_L$ is a dense open subset of $\mathbb{C}^L$ (see Proposition 4.1).

**Step 2** (*The set of all* $\boldsymbol{c} \in \mathbb{C}^L$ *with* $\det \boldsymbol{\Psi} \neq 0$ *is a dense open subset of* $\mathbb{C}^L$). We now write

$$\begin{aligned}
b - \boldsymbol{a}^{\mathbb{T}}(\overline{z_{\boldsymbol{c}}}\otimes z_{\boldsymbol{c}}) &= b - \boldsymbol{a}^{\mathbb{T}}\Big(\overline{(\boldsymbol{G}(\boldsymbol{c})|_\Lambda)^{-1}\pi(\lambda)\boldsymbol{c}}\otimes(\boldsymbol{G}(\boldsymbol{c})|_\Lambda)^{-1}\pi(\lambda)\boldsymbol{c}\Big)\\
&= b - \frac{1}{|\det(\boldsymbol{G}(\boldsymbol{c})|_\Lambda)|^2}\boldsymbol{a}^{\mathbb{T}}\Big(\overline{\operatorname{adj}(\boldsymbol{G}(\boldsymbol{c})|_\Lambda)\pi(\lambda)\boldsymbol{c}}\otimes \operatorname{adj}(\boldsymbol{G}(\boldsymbol{c})|_\Lambda)\pi(\lambda)\boldsymbol{c}\Big),
\end{aligned}$$

where $\operatorname{adj}(\boldsymbol{A})$ denotes the adjugate matrix of $\boldsymbol{A}$, which appears in Cramer's rule $\boldsymbol{A}^{-1} = \frac{1}{\det \boldsymbol{A}}\operatorname{adj}(\boldsymbol{A})$. Then

$$\begin{aligned}
\det\boldsymbol{\Psi} = \left|\det(\boldsymbol{G}(\boldsymbol{c})|_\Lambda)\right|^{2L-2}\Big(&b\cdot\left|\det(\boldsymbol{G}(\boldsymbol{c})|_\Lambda)\right|^2\\
&-\boldsymbol{a}^{\mathbb{T}}\Big(\overline{\operatorname{adj}(\boldsymbol{G}(\boldsymbol{c})|_\Lambda)\pi(\lambda)\boldsymbol{c}}\otimes\operatorname{adj}(\boldsymbol{G}(\boldsymbol{c})|_\Lambda)\pi(\lambda)\boldsymbol{c}\Big)\Big),
\end{aligned}$$

which is nonzero for all $\boldsymbol{c}$ in a dense open subset of $\mathbb{C}^L$, namely the subset of $U_L$ excluding a manifold of measure zero in $\mathbb{C}^L$. The manifold here is expressed by the equation $\det\boldsymbol{\Psi} = 0$, where we know from Step 1 that $\det\boldsymbol{\Psi}$ is a *nontrivial*

polynomial in the variables $c_0, \ldots, c_{L-1}$ and its complex conjugates $\overline{c_0}, \ldots, \overline{c_{L-1}}$. Hence, there exists a vector $\boldsymbol{c} \in \mathbb{C}^L$ such that $\boldsymbol{\Psi}$ is invertible. □

**Acknowledgments** This work was supported by German Research Foundation through Compressed Sensing in Information Processing Program under Grants PF 450/9-2 and PO 1347/3-2.

## References

1. Alltop, W.: Complex sequences with low periodic correlations. IEEE Trans. Inf. Theory **26**(3), 350–354 (1980)
2. Bello, P.: Characterization of randomly time-variant linear channels. IEEE Trans. Commun. Syst. **11**(4), 360–393 (1963)
3. Edelman, A.: Eigenvalues and condition numbers of random matrices. SIAM J. Matrix Anal. Appl. **9**(4), 543–560 (1988). https://doi.org/10.1137/0609045
4. Elad, M.: Sparse and Redundant Representations. Springer, New York (2010). https://doi.org/10.1007/978-1-4419-7011-4
5. Foucart, S., Rauhut, H.: A Mathematical Introduction to Compressive Sensing. Birkhäuser, Basel (2013)
6. Goldsmith, A.: Wireless Communications. Cambridge University Press, Cambridge (2005)
7. Heckel, R., Bölcskei, H.: Identification of sparse linear operators. IEEE Trans. Inf. Theory **59**(12), 7985–8000 (2013)
8. Hlawatsch, F., Matz, G.: Wireless Communications Over Rapidly Time-Varying Channels. Academic Press, Oxford (2011)
9. Kailath, T.: Measurements on time-variant communication channels. IRE Trans. Inf. Theory **8**(5), 229–236 (1962)
10. Kaplan, A., Lee, D., Pohl, V.: A message transmission scheme for linear time-varying multipath. In: 21st Intern. Workshop on Signal Process. Adv. in Wireless Commun. (SPAWC). Atlanta, USA (2020)
11. Kaplan, A., Lee, D., Pohl, V.: Message transmission through underspread time-varying linear channels. In: 45th Intern. Conf. on Acoust. Speech, and Signal Process. (ICASSP). Barcelona, Spain (2020)
12. Kozek, W., Pfander, G.E.: Identification of operators with bandlimited symbols. SIAM J. Math. Anal. **37**(3), 867–888 (2005)
13. Lawrence, J., Pfander, G.E., Walnut, D.: Linear independence of Gabor systems in finite dimensional vector spaces. J. Fourier. Anal. Appl. **11**(6), 715–726 (2005)
14. Lee, D., Pfander, G., Pohl, V.: Signal transmission through an unidentified channel. In: Proc. of 13th Intern. Conf. on Sampling Theory and Applications (SampTA). Bordeaux, France (2019)
15. Lee, D.G., Pfander, G.E., Pohl, V.: Sampling and reconstruction of multiple-input multiple-output channels. IEEE Trans. Signal Process. **67**(4), 961–976 (2019)
16. Lee, D.G., Pfander, G.E., Pohl, V., Zhou, W.: Identification of multiple-input multiple-output channels under linear side constraints. In: Proc. 43rd Intern. Conf. on Acoustics, Speech, and Signal Processing (ICASSP). Calgary, Canada (2018)
17. Lee, D.G., Pfander, G.E., Pohl, V., Zhou, W.: Identification of channels with single and multiple inputs and outputs under linear constraints. Linear Algebra Appl. **581**, 345–470 (2019). https://doi.org/10.1016/j.laa.2019.07.025
18. Malikiosis, R.D.: A note on Gabor frames in finite dimensions. Appl. Comput. Harmon. Anal. **38**(2), 318–330 (2015)

19. Matz, G., Hlawatsch, F.: Chapter 1—Fundamentals of time-varying communication channels. In: Hlawatsch, F., Matz, G. (eds.) Wireless Communications Over Rapidly Time-Varying Channels, pp. 1–63. Academic Press, Oxford (2011)
20. Pfander, G.E.: Measurement of time-varying multiple-input multiple-output channels. Appl. Comput. Harmon. Anal. **24**(3), 393–401 (2008)
21. Pfander, G.E.: Gabor frames in finite dimensions. In: Casazza, P.G., Kutyniok, G. (eds.) Finite Frames, Applied and Numerical Harmonic Analysis, pp. 193–239. Birkhäuser, Boston (2013)
22. Pfander, G.E., Walnut, D.: Measurement of time-variant channels. IEEE Trans. Inf. Theory **52**(11), 4808–4820 (2006)
23. Pfander, G.E., Walnut, D.F.: Measurement of time-variant linear channels. IEEE Trans. Inf. Theory **52**(11), 4808–4820 (2006)
24. Pfander, G.E., Walnut, D.F.: Sparse finite Gabor frames for operator sampling. In: Proc. of 10th Intern. Conf. on Sampling Theory and Applications (SampTA). Bremen, Germany (2013)
25. Pfander, G.E., Walnut, D.F.: Sampling and reconstruction of operators. IEEE Trans. Inf. Theory **62**(1), 435–458 (2016)
26. Pfander, G.E., Zheltov, P.: Identification of stochastic operators. Appl. Comput. Harmon. Anal. **26**(2), 256–279 (2014)
27. Pfander, G.E., Zheltov, P.: Sampling of stochastic operators. IEEE Trans. Inf. Theory **60**(4), 2359–2372 (2014)
28. Pfander, G.E., Zheltov, P.: Estimation of overspread scattering functions. IEEE Trans. Signal Process. **63**(10), 2451–2463 (2015)
29. Stevenhagen, P., Lenstra, H.W.: Chebotarëv and his density theorem. Math. Intelligencer **18**(2), 26–37 (1996). https://doi.org/10.1007/BF03027290
30. Tao, T.: An uncertainty principle for cyclic groups of prime order. Math. Res. Lett. **12**(1), 121–127 (2005)
31. Tong, L., Sadler, B.M., Dong, M.: Pilot-assisted wireless transmissions: general model, design criteria, and signal processing. IEEE Signal Process. Mag. **21**(6), 12–25 (2004)
32. Truong, K.T., Jr., R.W.H.: Effects of channel aging in massive mimo systems. IEEE J. Commun. Netw. **15**(4), 338–351 (2013)
33. Tsatsanis, M.K., Giannakis, G.B.: Modelling and equalization of rapidly fading channels. Int. J. Adapt. Control Signal Process. **10**(2-3), 159–176 (1996)
34. Tse, D., Viswanath, P.: Fundamentals of Wireless Communication. Cambridge University Press, Cambridge, Cambridge (2005)
35. Walnut, D., Pfander, G.E., Kailath, T.: Cornerstones of sampling of operator theory. In: Balan, R., Begué, M.J., Benedetto, J.J., Czaja, W., Okoudjou, K.A. (eds.) Excursions in Harmonic Analysis, Vol. 4, Applied and Numerical Harmonic Analysis, pp. 291–333 (2015)

# Chapter 5
# Analysis of Sparse Recovery Algorithms via the Replica Method


**Ali Bereyhi, Ralf R. Müller, and Hermann Schulz-Baldes**

## 5.1 Introduction

Statistical mechanics deals with the analysis of very large many-particle systems and seeks the following ultimate goal: Starting from the *microscopic* behavior of individual particles, it tries to find out the *macroscopic* properties of the system. The system size is, however, so large that it is not possible to solve the microscopic equations of motion. Statistical mechanics follows an alternative approach: It describes the microscopic behavior of the system particles via a stochastic model and extracts the desired *deterministic* properties via statistical analysis.

The goal and techniques of statistical mechanics are in various aspects similar to those of information theory. This connection has been widely investigated in the literature; see for instance [37]. In addition to all interesting theoretical aspects of this connection, the links between the two theories lead to a key achievement: The analytical tools of statistical mechanics can be used to address asymptotic analysis in information theory and its applications.

In this chapter, we use one particular statistical mechanical tool, namely the replica method, to investigate the asymptotic performance of a large class of sparse recovery algorithms. The interest in characterizing the asymptotic performance has several origins: The most natural one is to have an analytic bound on the performance of a given recovery scheme. This is however not the only application.

A. Bereyhi (✉) · R. R. Müller
Institute for Digital Communications, Friedrich-Alexander Universität Erlangen-Nürnberg, Erlangen, Germany
e-mail: ali.bereyhi@fau.de; ralf.r.mueller@fau.de

H. Schulz-Baldes
Department of Mathematics, Friedrich-Alexander Universität Erlangen-Nürnberg, Erlangen, Germany
e-mail: schuba@mi.uni-erlangen.de



G. Kutyniok et al. (eds.), *Compressed Sensing in Information Processing*, Applied and Numerical Harmonic Analysis, https://doi.org/10.1007/978-3-031-09745-4_5

Sparse recovery is used in several other applications in which the asymptotic performance characterization is useful for system design. In Sect. 5.7, we give two particular instances; namely, the example of deriving error bounds in distributed compressive sensing and tuning algorithms used for detection of spatially modulated signals.

The focus of this chapter is on the asymptotic analysis of a generic compressive sensing setting via the replica method. As a result, the contents of this chapter give a comprehensive overview on the replica method and its applications to asymptotic analyses in communications and signal processing. Details on most aspects of the discussion are given in [2].

## 5.2 A Multi-terminal Setting for Compressive Sensing

We consider a generic multi-terminal sensing setting. The setting includes the classical *single-terminal* compressive sensing setting, as well as other scenarios of sparse recovery.

Consider a distributed sensing network (DSN) with $J$ *correlated sparse* source signals, namely $x_j(t) \in \mathbb{X} \subseteq \mathbb{R}$ for $j \in [J]$. Here, the notation $[J]$ is defined as

$$[J] \coloneqq \{1, \ldots, J\}, \tag{5.1}$$

and is used through the chapter to shorten the presentation. The source signals are sampled at the time instances, $t = t_n$ for $n \in [N]$. Let $\boldsymbol{x}_j \in \mathbb{X}^{N\times 1}$ for $\mathbb{X} \subset \mathbb{R}$ denote the vector of samples collected from the $j$-th source signal. We assume that the sampling is performed, such that the *temporal* correlation among different time samples is negligible.[1] As the result, the sample vectors are statistically modeled as follows: $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_J$ are independent and identically distributed (i.i.d.), such that the time samples of the source signals at $t = t_n$ are *spatially* correlated.

The spatial correlation of the time samples at $t = t_n$ is modeled via the joint probability distribution $\mathrm{p}_X\left(x_n^J\right)$, where we define

$$x_n^J \coloneqq (x_{1n}, \ldots, x_{Jn}). \tag{5.2}$$

The joint distribution of the all signal samples is given by

$$\mathrm{p}\left(\boldsymbol{x}^J\right) = \prod_{n=1}^{N} \mathrm{p}_X\left(x_n^J\right), \tag{5.3}$$

where the notation $\boldsymbol{x}^J$ is defined as

[1] This is typically the case in classic signal sampling techniques.

$$\boldsymbol{x}^J \coloneqq (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_J)\,. \tag{5.4}$$

Considering source signal $j$, an individual sensing unit collects $M_j$ linear (and potentially noisy) observations of the samples. Denoting the vector of observations by $\boldsymbol{y}_j \in \mathbb{R}^{M_j \times 1}$, we can write

$$\boldsymbol{y}_j = \mathbf{A}_j \boldsymbol{x}_j + \boldsymbol{z}_j. \tag{5.5}$$

Here, $\mathbf{A}_j \in \mathbb{R}^{M_j \times N}$ denotes the *sensing matrix* of unit $j$ that describes the linear transform from the signal samples to the observations, and $\boldsymbol{z}_j \in \mathbb{R}^{M_j \times 1}$ is the measurement noise at terminal $j$.

The observations, as well as the sensing matrices, are given to a *single* data-fusion center. The data-fusion center recovers the signal samples using a *joint* recovery algorithm, i.e., it finds the estimates $\hat{\boldsymbol{x}}^J = \left(\hat{\boldsymbol{x}}_1, \ldots, \hat{\boldsymbol{x}}_J\right)$ as

$$\hat{\boldsymbol{x}}^J = \mathbf{g}\left(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_J | \mathbf{A}_1, \ldots, \mathbf{A}_J\right), \tag{5.6}$$

via some recovery algorithm $\mathbf{g}\left(\cdot | \mathbf{A}_1, \ldots, \mathbf{A}_J\right)$. At this point, we consider a generic form for the recovery algorithms. We will later focus on a specific (but broad) class of sparse recovery algorithms that use the method of least squares.

### 5.2.1 Characterization of the Recovery Performance

Before illustrating the details of the system model, let us clarify the ultimate goal of this chapter, i.e., the asymptotic analysis of sparse recovery algorithms. To this end, we first need to define a metric that characterizes the performance of a recovery algorithm $\mathbf{g}\left(\cdot | \mathbf{A}_1, \ldots, \mathbf{A}_J\right)$. This metric is defined in the following definition:

**Definition 5.1 (Average Distortion)** Consider the distortion function

$$\Delta\left(\cdot;\cdot\right) : \mathbb{R}^J \times \mathbb{R}^J \mapsto \mathbb{R}. \tag{5.7}$$

Using this function, the distortion between the source samples $\boldsymbol{x}^J$ and their corresponding estimates $\hat{\boldsymbol{x}}^J$ is determined as

$$\Delta_{\mathrm{V}}\left(\hat{\boldsymbol{x}}^J; \boldsymbol{x}^J\right) = \sum_{n=1}^{N} \Delta\left(\hat{x}_n^J; x_n^J\right). \tag{5.8}$$

The average distortion is then given by

$$D_N = \frac{1}{N}\mathbb{E}\left\{\Delta_{\mathrm{V}}\left(\hat{\boldsymbol{x}}^J; \boldsymbol{x}^J\right)\right\}, \tag{5.9}$$

where $\mathbb{E}\left\{\cdot\right\}$ indicates statistical expectation.

The average distortion describes the quality of the recovery algorithm. Depending on the choice of the distortion function, the average distortion determines different forms of estimation errors. For example, by setting

$$\Delta\left(\hat{x}_n^J; x_n^J\right) = \sum_{j=1}^{J} |\hat{x}_{jn} - x_{jn}|^2, \tag{5.10}$$

the average distortion reduces to the well-known mean squared error (MSE).

It is important to keep in mind that the average distortion explicitly depends on the recovery algorithm. In fact, depending on the choice of $\mathbf{g}(\cdot|\mathbf{A}_1, \ldots, \mathbf{A}_J)$, the estimated samples $\hat{\boldsymbol{x}}^J$ change, and consequently, $D_N$ varies.

The ultimate goal of this chapter is to find the average distortion for a large class of sparse recovery algorithms when the number of signal samples per each terminal, i.e., $N$, is very large. For most known sparse recovery algorithms, this is a hard task to do, due to reasons that we explain in Sect. 5.4.

In order to address this goal, we need to specify a model for every component of the setting. We do this in the following section.

### 5.2.2 Stochastic Model of System Components

A typical model for the noise processes is the additive white Gaussian noise (AWGN) model. This follows from the fact that noise in sensing devices is physically caused by several random independent processes whose spectral density in the bandwidth of interest is well approximated by that of AWGN. Considering the AWGN model, $\boldsymbol{z}_j$ is considered to be an i.i.d. Gaussian random vector whose entries are zero mean with variance $\sigma_j^2$.

We also model the sensing matrices as they are generated by a random process. Although stochastic modeling of noise is widely accepted, considering such a model for sensing matrices requires a bit of illustration. A stochastic model for sensing matrices assumes that each sensing matrix $\mathbf{A}_j$ is taken at random from a predefined ensemble. The logic behind considering such a model is as follows: From the compressive sensing literature, we know that sensing matrices require to satisfy some specific properties, such that a certain recovery performance is guaranteed [23]. Many random ensembles are shown to satisfy these properties. This means that by generating a sensing matrix from these ensembles at random, the anticipated recovery performance is achieved with a high probability. To incorporate this fact into the analysis, the classical approach is to assume that the sensing matrices are given by a random ensemble. From the mathematical viewpoint, such an assumption does not harm the generality of the analysis, as most structured sensing matrices can be described by a random ensemble as well.

In this chapter, we assume that the sensing matrices are *right rotationally invariant* random matrices. This is generic assumption since it includes most well-

known random ensembles, e.g., the class of i.i.d. sensing matrices. We introduce this random ensemble in the sequel. However, before defining that, let us first define the density of states for a given matrix.

**Definition 5.2 (Density of States)** Let $\mathbf{S} \in \mathbb{R}^{N \times N}$ be a self-adjoint square matrix whose eigenvalues are given by $\lambda_1, \ldots, \lambda_N \in \mathbb{R}$. The density of states for this matrix is defined as the empirical cumulative distribution function (CDF) of its eigenvalues,[2] i.e., for $\lambda \in \mathbb{R}$

$$F_{\mathbf{S}}^{N}(\lambda) = \frac{1}{N} \sum_{n=1}^{N} \mathbf{1}\{\lambda_n < \lambda\}. \tag{5.11}$$

We are now ready to define right rotationally invariant random matrices.

**Definition 5.3 (Right Rotationally Invariant Matrices)** $\mathbf{A}_j \in \mathbb{R}^{M_j \times N}$ is right rotationally invariant if its Gramian $\mathbf{J}_j = \mathbf{A}_j^{\mathsf{T}} \mathbf{A}_j$ has the decomposition

$$\mathbf{J}_j = \mathbf{U}_j \mathbf{D}_j \mathbf{U}_j^{\mathsf{T}}, \tag{5.12}$$

where $\mathbf{U}_j$ and $\mathbf{D}_j$ have the following properties:

1. The matrix $\mathbf{U}_j \in \mathbb{R}^{N \times N}$ is a Haar-distributed matrix.[3]
2. The matrix $\mathbf{D}_j \in \mathbb{R}^{N \times N}$ denotes the diagonal matrix of eigenvalues whose density of states converges as $N \to \infty$, i.e., $\lim_{N \to \infty} F_{\mathbf{D}_j}^{N}(\lambda) = F_j(\lambda)$.

The class of right rotationally invariant matrices includes the most well-known random ensembles in compressive sensing; for instance, the class of i.i.d. random matrices, i.e., random matrices whose entries are generated i.i.d. from a distribution with bounded variance.[4] Note that different forms of random matrices will have different densities of states. We clarify this point further in Sect. 5.4 when we formally formulate the asymptotic analysis of a sparse recovery algorithm.

### 5.2.3 *Stochastic Model for Jointly Sparse Signals*

As indicated, we assume that there exists a spatial correlation among the signals for various sources. Noting that the signals are sparse, we interpret this spatial coupling as *joint sparsity*. To give an intuition on joint sparsity, we focus in the sequel on a

[2] Note that for random $\mathbf{S}$, the density of state is random.

[3] A Haar matrix is a random matrix generated from the rotation-invariant measure on the set of all orthonormal matrices.

[4] Another well-known example in compressive sensing is the row-orthonormal random sensing matrix; see [39] and references therein for the exact definition and further examples.

special joint sparsity model in which the sample $n$ of signal $j$ is written as

$$x_{jn} = c_n w_{0n} + s_{0n} w_{jn} + s_{jn} u_{jn}. \tag{5.13}$$

Here, $s_{0n}$, $w_{0n}$, $c_n$, $w_{jn}$, $s_{jn}$, and $u_{jn}$ are independent i.i.d. sequences whose distributions are as follows:

1. In (5.13), the random variable $w_{0n}$, as well as the random variables $w_{jn}$ and $u_{jn}$ for $j \in [J]$, are defined to be in set $\mathbb{X}$, and their probabilities of being zero are equal to zero.
2. Random variables $s_{0n}$, $c_n$, and $s_{jn}$ are Bernoulli-distributed and

$$\Pr\{c_n = 1\} = 1 - \Pr\{c_n = 0\} = \mu_{\mathrm{c}} \tag{5.14a}$$

$$\Pr\{s_{0n} = 1\} = 1 - \Pr\{s_{0n} = 0\} = \mu_0 \tag{5.14b}$$

$$\Pr\{s_{jn} = 1\} = 1 - \Pr\{s_{jn} = 0\} = \mu_j. \tag{5.14c}$$

In this model, the samples of a terminal are given as the superposition of three sparse components. The first component, whose $n$-th entry is given by $c_n w_{0n}$, is a sparse vector that is common among all the terminals. The second sparse component, represented by $s_{0n} w_{jn}$ for $n \in [N]$, has a common support[5] across all the terminals; however, the values of the non-zero entries are drawn from independent processes. The last component contains a sparse signal whose support and non-zero entries are independently generated for each terminal.

Although the given model for joint sparsity is not the most general one, it includes the most well-known sparse recovery settings in the literature. In the sequel, we address the main settings for sparse recovery. These settings are derived from our system model as special cases.

### *5.2.4 Special Cases*

The three main settings for sparse recovery are classical compressive sensing, the problem of multiple measurement vectors (MMV), and distributed compressive sensing (DCS). In the sequel, we briefly go through these settings and illustrate how they are derived from our generic multi-terminal setting.

#### 5.2.4.1 Classical Compressive Sensing

In classical compressive sensing, also called the single measurement vector problem, a sparse signal is observed linearly via a *single* terminal and is to be recovered

[5] By support, we refer to the indices of non-zero entries in a vector.

from this underdetermined set of observations. This setting is simply derived by setting $J = 1$ in our model. As spatial correlation among terminals has no meaning in this case, one can further set $c_n = s_{0n} = 0$ for $n \in [N]$ in the sparsity model.

#### 5.2.4.2 Multiple Measurement Vectors

In MMV, multiple sparse signals are observed with a common sensing matrix and recovered at a single data-fusion center. This setting is straightforwardly derived from our model by letting

$$\mathbf{A}_1 = \ldots = \mathbf{A}_J. \tag{5.15}$$

In general, the joint sparsity model given in Sect. 5.2.3 is a valid model in MMV. Nevertheless, in many applications of MMV, it is common to assume the *common support* model for the spatial correlation. This model assumes that the samples of different terminals have common support; however, the non-zero entries are drawn from independent processes. The common support model is derived from the joint sparsity model in Sect. 5.2.3 by setting $s_{jn} = c_n = 0$ for $n \in [N]$.

#### 5.2.4.3 Distributed Compressive Sensing

DCS describes the most generic setting that fits to our model. In this problem, the jointly sparse signals of different terminals are observed with different matrices and recovered at a common fusion center. Similar to MMV, the joint sparsity model in Sect. 5.2.3 is generally valid for DCS. A common model is however the *common-innovation* model in which the signal of each terminal is given as a common sparse component superposed by an independent sparse innovation term. This model is derived from the one given in Sect. 5.2.3 by setting $s_{0n} = 0$ for $n \in [N]$.

## 5.3 Sparse Recovery via the Regularized Least-Squares Method

We focus on the class of regularized least-squares (RLS)-based recovery algorithms. These algorithms recover the signal samples by minimizing a penalized residual sum of squares. In general, an RLS-based algorithm is of the following form:

$$\mathbf{g}\left(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_J | \mathbf{A}_1, \ldots, \mathbf{A}_J\right) = \underset{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_J \in \mathbb{X}^N}{\operatorname{argmin}} \sum_{j=1}^{J} \frac{1}{2\lambda_j} \|\boldsymbol{y}_j - \mathbf{A}_j \boldsymbol{v}_j\|^2 + u_{\mathrm{V}}\left(\boldsymbol{v}^J\right). \tag{5.16}$$

Here, $\boldsymbol{v}^J \coloneqq (\boldsymbol{v}_1, \ldots, \boldsymbol{v}_J)$, and $u_{\mathrm{V}}(\cdot) : \mathbb{R}^{N\times J} \mapsto \mathbb{R}^+$ is the *regularization function* that penalizes the residual sum of squares, and $\|\cdot\|$ denotes the Euclidean norm. In the sequel, we assume that $u_{\mathrm{V}}(\cdot)$ decouples, i.e., there exists $u(\cdot) : \mathbb{R}^{1\times J} \mapsto \mathbb{R}^+$, such that

$$u_{\mathrm{V}}(\boldsymbol{v}^J) = \sum_{n=1}^{N} u\left(v_n^J\right). \tag{5.17}$$

Furthermore, $\lambda_1, \ldots, \lambda_J$ are tunable factors, referred to as *regularization parameters*.

### 5.3.1 Some Well-Known Forms

The interest in the class of RLS-based recovery schemes comes from its broadness. In fact, the recovery scheme in (5.16) includes a diverse set of sparse recovery algorithms. In the sequel, we discuss two well-known examples, namely $\ell_p$-norm minimization for classical compressive sensing and $\ell_{p,q}$-norm minimization for DCS.

#### 5.3.1.1 $\ell_p$-Norm Minimization

Most algorithms in compressive sensing with a single terminal recover the sparse signal by finding a vector of samples whose residual sum of squares is bounded, i.e., finding $\boldsymbol{v}$, such that[6]

$$\|\boldsymbol{y} - \mathbf{A}\boldsymbol{v}\|^2 \leq \epsilon \tag{5.18}$$

for some $\epsilon$, and whose $\ell_p$-norm for some $p$ is minimum. The most common choice of $p$ is $p = 1$, which results in the least absolute shrinkage and selection operator (LASSO), also called basis pursuit algorithm.

Using the method of Lagrange multipliers, it is shown that there exists a regularization parameter $\lambda$, for which the RLS-based algorithm with decoupled regularization function $u(v_n) = |v_n|^p$ performs identical to this algorithm; see, for example, [2].

[6] Note that the index $j$ is dropped, as we consider a single-terminal setting.

#### 5.3.1.2 $\ell_{p,q}$-Norm Minimization

For multi-terminal settings, the classic $\ell_p$-norm minimization techniques are often extended to $\ell_{p,q}$-norm minimization techniques. The feasible set in this case is constructed with the same approach, i.e., finding $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_J$ for which

$$\|\boldsymbol{y}_j - \mathbf{A}_j \boldsymbol{v}_j\|^2 \leq \epsilon_j \tag{5.19}$$

with some $\epsilon_j$ for $j \in [J]$. The recovered samples are then found by searching the feasible set for vectors whose $\ell_{p,q}$-norm is minimum. For a collection of $J$ vectors $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_J$, the $\ell_{p,q}$-norm is defined as

$$\|\boldsymbol{v}_1, \ldots, \boldsymbol{v}_J\|_{p,q} = \left( \sum_{n=1}^{N} \left( \sum_{j=1}^{J} |v_{jn}|^p \right)^{q/p} \right)^{1/q}. \tag{5.20}$$

The most well-known $\ell_{p,q}$-norm minimization technique is the group LASSO technique in which $p = 2$ and $q = 1$.

Similar to $\ell_p$-norm minimization, one can invoke the method of Lagrange multipliers and show that there exist regularization parameters $\lambda_1, \ldots, \lambda_J$, for which the RLS-based algorithm with decoupled regularization function $u\left(v_n^J\right) = \|v_n^J\|_p{}^q$ performs identical to $\ell_{p,q}$-norm minimization.

### 5.3.2 Bayesian Interpretation

In the Bayesian framework, an RLS-based algorithm is interpreted as a *mismatched* maximum-a-posteriori (MAP) estimator. This estimator postulates the following assumptions:

1. The *prior* joint distribution of samples at $t_n$ is proportional to $\exp\{-u(\cdot)\}$. This means that $\mathrm{p}_X\left(x_n^J\right)$ is *assumed* to be

$$\mathrm{p}_X\left(x_n^J\right) = \frac{\exp\{-u(\cdot)\}}{Z} \tag{5.21}$$

   for some normalization factor $Z$.
2. The noise processes are Gaussian.
3. The variance of noise at terminal $j$ is proportional to $\lambda_j$.

It then calculates the posterior distribution of the signal samples, i.e., it finds the conditional distribution $\mathrm{p}\left(\boldsymbol{x}^J | \boldsymbol{y}_1, \ldots \boldsymbol{y}_J, \mathbf{A}_1, \ldots, \mathbf{A}_J\right)$, and determines its maximizer as the estimate. Note that the postulated parameters are not necessarily matched to the true ones. This is in fact why the MAP estimator is called *mismatched*.

Using the Bayes' rule, it is shown that the posterior distribution, given the postulated model, is of the following form:

$$\mathrm{p}\left(\boldsymbol{x}^J|\boldsymbol{y}_1,\ldots\boldsymbol{y}_J,\mathbf{A}_1,\ldots,\mathbf{A}_J\right)=C\exp\left\{-\sum_{j=1}^{J}\frac{1}{2\lambda_j}\|\boldsymbol{y}_j-\mathbf{A}_j\boldsymbol{v}_j\|^2-u_{\mathrm{V}}\left(\boldsymbol{v}^J\right)\right\} \tag{5.22}$$

for some constant $C$. The MAP estimation hence reduces to the maximization of the exponent term that recovers the RLS-based recovery.

Given the Bayesian interpretation, one concludes that most MAP estimators used in classical signal processing and machine learning models can be reformulated as an RLS-based algorithm. As the result, the analysis in this chapter can be directly extended to Bayesian estimation[7] in other applications.

## 5.4 Asymptotic Characterization

Now that the system model and recovery algorithm are presented, we are ready to formally formulate the asymptotic performance of a sparse recovery algorithm. For the asymptotic analysis, we consider a sequence of settings. The number of signal samples $N$ and the number of measurements at terminal $j$, i.e., $M_j$, in this sequence grow large, such that $M_j$ is a deterministic function of $N$. We assume that $N$ grows unboundedly large, and $M_j$ grows with $N$ linearly. This means that there exists a fixed $\rho_j$ (typically $\rho_j \leq 1$) for each $j \in [J]$, such that

$$\rho_j \coloneqq \lim_{N\uparrow\infty}\frac{M_j}{N}<\infty. \tag{5.23}$$

We refer to $\rho_j$ as the $j$-th terminal compression ratio.

For every DSN in the sequence, we use an RLS-based algorithm to recover the signal samples. Let $D_N$ denote the average distortion between the true signal samples and their estimates in the DSN whose index is $N$; see (5.9). The asymptotic analysis intends to find the asymptotic limit of this sequence of distortions, i.e.,

$$D\coloneqq\lim_{N\uparrow\infty}D_N. \tag{5.24}$$

The derivation of $D$ and its dependence on the various model parameters are not straightforward from both analytical and computational points of view. In fact, depending on the regularization function, the derivation of $D$ deals with one or two of the following issues:

[7] Or to learning algorithms in a Bayesian framework.

- For some regularization functions, the RLS-based algorithm solves a *convex* optimization problem that can be posed as standard convex programming. Hence, it is performed in polynomial time. Although RLS-based recovery in this case is computationally tractable, there is no guarantee that the problem is also *analytically* tractable. For asymptotic analysis, one needs to determine the sequence of average distortions for any integer index $N$ and take the limit when $N$ goes to $\infty$. For some particular RLS-based algorithms, this task can be done via basic analytic tools; nevertheless, there are several forms whose limit is not known analytically via the basic tools.
- Several RLS-based algorithms are not only analytically, but also computationally intractable. An example is the $\ell_0$-norm minimization algorithm in which the regularization function is proportional to the $\ell_0$-norm. For this choice of regularization function, the recovery algorithm reduces to a decision problem that belongs to the class of nondeterministic polynomial time (NP)-complete problems and hence is NP-hard [23]. Another instance is the case in which RLS is used for recovery of a signal whose samples are drawn from a discrete support, i.e., $\mathbb{X}$ be a discrete set. Similar to $\ell_0$-norm minimization, RLS-based recovery in this case is NP-hard since it deals with integer programming.[8] Clearly, for these forms, asymptotic characterization is not computationally tractable.

The above analytical and computational issues can be addressed via the replica method. As it becomes clear later, the replica method invokes several non-rigorous tricks to bypass the analytical obstacles of the problem. The term *non-rigorous tricks* will be clarified in the next sections of this chapter, while we illustrate how the replica method exactly does that.

Now that the asymptotic analysis is formulated in principle, we can state explicitly our main purpose as follows: The main purpose is to illustrate how the asymptotic distortion $D$ is derived for an RLS-based algorithm via the replica method.

### 5.4.1 Stieltjes and R-Transforms

Before we start with the illustration of the replica method, we give some basic definitions that are used throughout the derivations via the replica method. These definitions enable us to compactly represent the statistics of the sensing matrices.

To start with the definitions, consider the sequence of DSNs indexed by $N$. For each terminal, there exists a corresponding sequence of densities $F_j^N(\lambda)$ that for a given index $N$ describes the density of states for $\mathbf{J}_j = \mathbf{A}_j^{\mathsf{T}}\mathbf{A}_j$. We assume that this sequence converges as $N \to \infty$ to a deterministic density of states $F_j(\lambda)$ for each

[8] Note that this is the case for any choice of the regularization function.

$j \in [J]$. For these asymptotic densities, the Stieltjes and R-transforms are defined as follows [53]:

**Definition 5.4 (Stieltjes Transform)** For the asymptotic CDF $F_j(\lambda)$, the Stieltjes transform is given by

$$\mathrm{G}_j(s) = \int_{-\infty}^{+\infty} \frac{1}{\lambda - s} \mathrm{d}F_j(\lambda) \tag{5.25}$$

for some complex $s$ with $\mathrm{Im}\, s \geq 0$, where $\mathrm{Im}\, s$ is the imaginary part of $s$.

**Definition 5.5 (R-Transform)** For the asymptotic density $F_j(\lambda)$, the R-transform is defined as

$$\mathrm{R}_j(w) = \mathrm{G}_j^{-1}(-w) - \frac{1}{w}, \tag{5.26}$$

where $\mathrm{G}_j^{-1}(\cdot)$ denotes the inverse of the Stieltjes transform with respect to composition. If $\mathrm{G}_j^{-1}(\cdot)$ has multiple solutions, a solution is selected whose corresponding calculation of R-transform satisfies the following conditions:

1. The following limit exists:

$$\lim_{w \to 0} \mathrm{R}_j(w) = \int_{-\infty}^{+\infty} \lambda \mathrm{d}F_j(\lambda). \tag{5.27}$$

2. $\mathrm{R}_j(w)$ is an increasing function on the real axis.[9]

The definition of the R-transform is further extended to matrix arguments: Consider a self-adjoint matrix $\mathbf{S}_{N \times N}$ with the eigendecomposition

$$\mathbf{S} = \mathbf{W} \boldsymbol{\Lambda} \mathbf{W}^{-1}. \tag{5.28}$$

For this matrix, we use the notation $\mathrm{R}_j(\mathbf{S})$ to refer to

$$\mathrm{R}_j(\mathbf{S}) \coloneqq \mathbf{W}\, \mathrm{Diag}\left\{\mathrm{R}_j(\lambda_1), \ldots, \mathrm{R}_j(\lambda_n)\right\}\, \mathbf{W}^{-1}, \tag{5.29}$$

where $\mathrm{Diag}\{a_1, \ldots, a_N\}$ denotes an $N \times N$ diagonal matrix whose diagonal entries are $a_1, \ldots, a_N$.

[9] More precisely, if $F_j(\lambda)$ is different from a step function at a single mass point, i.e., derivative of $F_j(\lambda)$ is different from a Dirac impulse at a single mass point, $\mathrm{R}_j(w)$ is *strictly* increasing; for details, see [59, Appendix E].

## 5.5 Building a Bridge to Statistical Mechanics

As mentioned before, the replica method was initially developed in statistical mechanics for the analysis of spin glasses. Nevertheless, it found its way to several other fields, such as coding, information theory, and signal processing. The key point in employing the replica method for asymptotic analysis is to make a connection between the problem at hand and the theory of spin glasses. In this section, we illustrate how this connection is made. To this end, we need first to give a quick overview on basic definitions in statistical mechanics. The contents of this section are discussed with details in [2, Chapter 3]. For further discussions on fundamentals of statistical mechanics and its connections to information theory and signal processing, see [37] and the references therein.

### *5.5.1 Introduction to Statistical Mechanics*

A thermodynamic system consists of $N$ particles with each having a microscopic parameter $v_n \in \mathbb{V}$ for $n \in [N]$ and some set $\mathbb{V}$. This parameter describes a macroscopic property of the corresponding particle, e.g., the velocity. In general, a microscopic parameter could be a vector of continuous or discrete entries. For sake of brevity, we assume that $v_n$ is a *continuous* scalar. The extension to cases with discrete $v_n$ can be followed in [2, Chapter 3]. For this system, the *microstate* is defined as a vector in $\mathbb{V}^N$ that collects microscopic parameters of all the particles, i.e.,

$$\boldsymbol{v} = [v_1, \ldots, v_N]^{\mathsf{T}}. \tag{5.30}$$

Corresponding to this system, a *Hamiltonian* $\mathcal{E}(\cdot)$ is defined, which describes the physical properties of the system. The Hamiltonian is a function that assigns to microstate $\boldsymbol{v}$ a non-negative energy level $\mathcal{E}(\boldsymbol{v})$.

*Remark 5.1* Here, we have defined the Hamiltonian in an abstract form. For a physical system, the explicit form of the Hamiltonian is derived from the physical theories that describe the interactions of microscopic parameters in the system.

For a thermodynamic system, the explicit calculation of macroscopic parameters is intractable.[10] To address this issue, statistical mechanics follows a stochastic approach. In this approach, the microstate is considered to be a random vector whose distribution depends on the *temperature*. We denote this distribution by $\mathrm{p}_\beta(\boldsymbol{v})$, where $\beta$ is the *inverse temperature*, i.e., $\beta = 1/T$ with $T$ being the temperature.

[10] This follows the same reasons given in Sect. 5.4 for the asymptotic analysis of RLS-based algorithms.

Using stochastic analysis, statistical mechanics derives physical features of the thermodynamic system from this stochastic model. These physical features are known as *macroscopic* parameters of the system. Mathematically, a thermodynamic system can be described via the following two macroscopic parameters: *entropy* and *free energy*.[11] These parameters are defined as follows:

**Definition 5.6 (Normalized Entropy)** For a given thermodynamic system with $N$ particles, the normalized entropy at inverse temperature $\beta$ is defined as

$$\mathcal{H}_N(\beta) \coloneqq -\frac{1}{N} \int_{\mathbb{V}^N} \mathrm{p}_\beta(\boldsymbol{v}) \log \mathrm{p}_\beta(\boldsymbol{v}) \, \mathrm{d}\boldsymbol{v}. \tag{5.31}$$

**Definition 5.7 (Normalized Free Energy)** Consider a thermodynamic system with $N$ particles and Hamiltonian $\mathcal{E}(\cdot)$. At inverse temperature $\beta$, the normalized free energy is defined as

$$\mathcal{F}_N(\beta) \coloneqq \frac{1}{N} \mathbb{E}\{\mathcal{E}(\boldsymbol{v})\} - \frac{1}{\beta} \mathcal{H}_N(\beta), \tag{5.32}$$

where the expectation is taken with respect to $\mathrm{p}_\beta(\boldsymbol{v})$.

#### 5.5.1.1 Second Law of Thermodynamics

The fundamental rule in stochastic analysis of thermodynamic systems is the second law of thermodynamics. This law indicates that the microstate in thermal equilibrium[12] is distributed such that the free energy is minimized. Since $\mathcal{F}_N(\beta)$ is convex with respect to $\mathrm{p}_\beta(\boldsymbol{v})$, it is concluded that the microstate is distributed with the *Boltzmann–Gibbs* distribution. This means that at thermal equilibrium

$$\mathrm{p}_\beta(\boldsymbol{v}) = \frac{\exp\{-\beta \mathcal{E}(\boldsymbol{v})\}}{\mathcal{Z}_N(\beta)} \tag{5.33}$$

at inverse temperature $\beta$. In the denominator, $\mathcal{Z}_N(\beta)$ is a normalization factor, i.e.,

$$\mathcal{Z}_N(\beta) = \int_{\mathbb{V}^N} \exp\{-\beta \mathcal{E}(\boldsymbol{v})\} \, \mathrm{d}\boldsymbol{v}, \tag{5.34}$$

[11] In fact, the main two macroscopic parameters of a thermodynamic system are entropy and energy. The free energy is derived by applying the second law of thermodynamics as the Lagrange dual function. We however use directly the free energy in our formulation, for sake of brevity.

[12] This means that there is no energy flow.

and is called the *partition function*. The distribution $\mathrm{p}_\beta(\boldsymbol{v})$ reduces to some well-known distributions for several choices of the Hamiltonian, e.g., it reduces to the Gaussian distribution when $\mathcal{E}(\boldsymbol{v}) \propto \|\boldsymbol{v}\|^2$.

*Remark 5.2* The stated form of the second law of thermodynamics is a simplified interpretation of the original form. In fact, the law states that the entropy in an isolated system grows constantly. This is interpreted as a constrained optimization problem in which the normalized entropy is maximized subject to an energy constraint. Using the method of Lagrange multipliers, the free energy is derived as the objective function of the dual unconstrained optimization. It is then shown that the Lagrange multiplier is in fact the temperature.

Substituting the Boltzmann–Gibbs distribution in the definition of the free energy, it is concluded that

$$\mathcal{F}_N(\beta) = -\frac{1}{\beta N} \log \mathcal{Z}_N(\beta). \tag{5.35}$$

This is a fundamental identity indicating that the free energy of a system in thermal equilibrium is calculated explicitly from the partition function. Starting from this equation, it is shown that all other macroscopic parameters of the system are directly derived from $\mathcal{F}_N(\beta)$. For instance,

$$\mathcal{H}_N(\beta) = \beta^2 \frac{\mathrm{d}}{\mathrm{d}\beta} \mathcal{F}_N(\beta). \tag{5.36}$$

Therefore, the partition function completely describes the macroscopic features of the system in thermal equilibrium.

#### 5.5.1.2 Spin Glasses

*Spin glasses* are thermodynamic systems whose particles choose to interact randomly. This means that the Hamiltonian of a spin glass is not only a function of the microstate, but also a randomizer. This randomizer is realized once from a random ensemble and remains fixed as the system is in thermal equilibrium.[13]

Similar to thermodynamic systems, the stochastic analysis of spin glasses follows the second law of thermodynamics. Let $\boldsymbol{\Omega}$ denote the randomizer of a spin glass. The Hamiltonian of this spin glass is given by

$$\mathcal{E}(\cdot|\boldsymbol{\Omega}) : \mathbb{V}^N \mapsto \mathbb{R}^+. \tag{5.37}$$

[13] In statistical mechanics, this randomizer is known to have *quenched* randomness. This is different from the type of randomness considered for the microstate.

In other words, for every realization of $\boldsymbol{\Omega}$, we have a specific Hamiltonian function. By the same lines of derivations explained in Sect. 5.5.1.1, one can show that, *conditioned* on the randomizer, the microstate of the spin glass in thermal equilibrium is distributed with the Boltzmann–Gibbs distribution. This means that

$$\mathrm{p}_{\beta}\left(\boldsymbol{v}|\boldsymbol{\Omega}\right)=\frac{\exp\left\{-\beta\mathcal{E}\left(\boldsymbol{v}|\boldsymbol{\Omega}\right)\right\}}{\mathcal{Z}_{N}\left(\beta|\boldsymbol{\Omega}\right)}, \tag{5.38}$$

with random partition function

$$\mathcal{Z}_{N}\left(\beta|\boldsymbol{\Omega}\right)=\int\limits_{\mathbb{V}^{N}}\exp\left\{-\beta\mathcal{E}\left(\boldsymbol{v}|\boldsymbol{\Omega}\right)\right\}\mathrm{d}\boldsymbol{v}. \tag{5.39}$$

The normalized free energy in thermal equilibrium is hence written as

$$\mathcal{F}_{N}\left(\beta|\boldsymbol{\Omega}\right)=-\frac{1}{\beta N}\log\mathcal{Z}_{N}\left(\beta|\boldsymbol{\Omega}\right), \tag{5.40}$$

and the *conditional* entropy is determined from the free energy by

$$\mathcal{H}_{N}\left(\beta|\boldsymbol{\Omega}\right)=\beta^{2}\frac{\mathrm{d}}{\mathrm{d}\beta}\mathcal{F}_{N}\left(\beta|\boldsymbol{\Omega}\right). \tag{5.41}$$

In the remaining parts of this chapter, we focus on spin glasses. This is due to the fact that our problem is formulated in terms of a spin glass.

#### 5.5.1.3 Thermodynamic Limit

Spin glasses are studied in the *thermodynamic limit*. This means that the macroscopic parameters are derived for the case, in which the number of particles tends to infinity, i.e., the asymptotic limit $N\uparrow\infty$. Suggested by physical intuition, in the thermodynamic limit, a spin glass has deterministic macroscopic parameters. This means that in the asymptotic limit, the free energy $\mathcal{F}_{N}\left(\beta|\boldsymbol{\Omega}\right)$ converges to its expected value.[14] This property of spin glasses is known as *self-averaging*; more discussions in this respect can be followed in [25, 26, 43].

Following the self-averaging property, the free energy of a spin glasses in the thermodynamic limit is calculated as follows:

1. Determining the sequence of *expected* free energies $\bar{\mathcal{F}}_{N}\left(\beta|\boldsymbol{\Omega}\right)$ indexed by $N$ as

$$\bar{\mathcal{F}}_{N}\left(\beta\right)=\mathbb{E}\left\{\mathcal{F}_{N}\left(\beta|\boldsymbol{\Omega}\right)\right\}, \tag{5.42}$$

[14] Here, the expectation is taken over the randomizer $\boldsymbol{\Omega}$.

where the expectation is taken with respect to $\boldsymbol{\Omega}$.

2. Taking the asymptotic limit of the expected sequence, i.e., calculating

$$\bar{\mathcal{F}}(\beta) = \lim_{N\uparrow\infty} \bar{\mathcal{F}}_N(\beta). \tag{5.43}$$

#### 5.5.1.4 Averaging Trick

Before we start with the derivations, let us illustrate the key *averaging trick* in statistical mechanics. Consider a function $\psi_N(\cdot)$ that for each microstate $\boldsymbol{v} \in \mathbb{V}^N$ determines a scalar parameter. The macroscopic parameter corresponding to this function is defined as

$$\bar{\psi}_N = \frac{1}{N}\mathbb{E}\{\psi_N(\boldsymbol{v})\}, \tag{5.44}$$

where the expectation is taken first with respect to the conditional Boltzmann–Gibbs distribution, i.e., $\mathrm{p}_\beta(\boldsymbol{v}|\boldsymbol{\Omega})$, and then with respect to $\boldsymbol{\Omega}$.

The classic approach for determining $\bar{\psi}_N$ in statistical mechanics is to use the averaging trick. This trick modifies the partition function with a dummy factor $h$ as follows:

$$\mathcal{Z}_N(\beta, h|\boldsymbol{\Omega}) = \int_{\boldsymbol{v}\in\mathbb{V}^N} \exp\{-\beta\mathcal{E}(\boldsymbol{v}|\boldsymbol{\Omega}) + h\psi_N(\boldsymbol{v})\}\,\mathrm{d}\boldsymbol{v}. \tag{5.45}$$

For this modified partition function, the normalized free energy, conditioned on a realization of the randomizer, is

$$\mathcal{F}_N(\beta, h|\boldsymbol{\Omega}) = -\frac{1}{\beta N}\log\mathcal{Z}_N(\beta, h|\boldsymbol{\Omega}), \tag{5.46}$$

and its expected value $\bar{\mathcal{F}}_N(\beta, h)$ is determined by calculating the expectation over $\boldsymbol{\Omega}$, i.e., as in (5.42).

By standard derivations, it is readily shown that

$$\bar{\psi}_N = -\beta\frac{\partial}{\partial h}\bar{\mathcal{F}}_N(\beta, h)\,|_{h=0}. \tag{5.47}$$

Exchanging limiting procedures, one has in the thermodynamic limit

$$\bar{\psi} \coloneqq \lim_{N\uparrow\infty} \bar{\psi}_N \tag{5.48a}$$

$$= -\beta\frac{\partial}{\partial h}\bar{\mathcal{F}}(\beta, h)\,|_{h=0}. \tag{5.48b}$$

### 5.5.2 Corresponding Spin Glass

The connection between the sparse recovery problem and the statistical mechanics is illustrated by introducing the concept of *corresponding spin glass*. In fact, for an RLS-based recovery algorithm, we can define an *imaginary* spin glass whose macroscopic parameters are the asymptotic performance metrics of the recovery algorithm. We clarify this connection in the sequel.

Remember the system model in Sect. 5.2 with sensing matrices $\mathbf{A}_1, \ldots \mathbf{A}_J$ and observation vectors $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_J$. We define the corresponding spin glass as follows:

**Definition 5.8 (Corresponding Spin Glass)** The corresponding spin glass is a spin glass whose microstate is described by $\boldsymbol{v}^J = (\boldsymbol{v}_1, \ldots, \boldsymbol{v}_J)$, where $\boldsymbol{v}_j \in \mathbb{X}^N$ for $j \in [J]$. The randomizer of this spin glass is

$$\boldsymbol{\Omega} = \left\{\mathbf{A}_1, \ldots, \mathbf{A}_J, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_J\right\}, \tag{5.49}$$

and its Hamiltonian is

$$\mathcal{E}\left(\boldsymbol{v}^J|\boldsymbol{\Omega}\right) = \sum_{j=1}^{J} \frac{1}{2\lambda_j}\|\boldsymbol{y}_j - \mathbf{A}_j\boldsymbol{v}_j\|^2 + u_{\mathrm{V}}\left(\boldsymbol{v}^J\right). \tag{5.50}$$

From our earlier discussions, we know that at inverse temperature $\beta$, the microstate in thermal equilibrium is conditionally distributed with

$$\mathrm{p}_\beta\left(\boldsymbol{v}^J|\boldsymbol{\Omega}\right) = \frac{\exp\left\{-\beta\mathcal{E}\left(\boldsymbol{v}^J|\boldsymbol{\Omega}\right)\right\}}{\mathcal{Z}_N\left(\beta|\boldsymbol{\Omega}\right)}, \tag{5.51}$$

where the partition function $\mathcal{Z}_N\left(\beta|\boldsymbol{y}, \mathbf{A}\right)$ reads

$$\mathcal{Z}_N\left(\beta|\boldsymbol{\Omega}\right) = \int\limits_{\boldsymbol{v}_j \in \mathbb{X}^N} \exp\left\{-\beta\mathcal{E}\left(\boldsymbol{v}^J|\boldsymbol{\Omega}\right)\right\} \mathrm{d}\boldsymbol{v}^J. \tag{5.52}$$

The key property of this spin glass that connects it to our sparse recovery problem is its *ground-state property*.

**Theorem 5.1 (Ground-State Property)** *For a given realization of* $\boldsymbol{\Omega}$*, assume that the Hamiltonian has a unique minimizer denoted by* $\boldsymbol{v}_\star^J\left(\boldsymbol{\Omega}\right)$*. Then, as the temperature goes to zero, i.e.,* $\beta \uparrow \infty$*, the microstate of the corresponding spin glass converges in distribution to the deterministic vector* $\boldsymbol{v}_\star^J\left(\boldsymbol{\Omega}\right)$*. This means that for every realization of* $\boldsymbol{\Omega}$

$$\lim_{\beta\uparrow\infty} p_\beta\left(\boldsymbol{v}^J|\boldsymbol{\Omega}\right) = \begin{cases} 1 & \boldsymbol{v}^J = \boldsymbol{v}_\star^J\left(\boldsymbol{\Omega}\right) \\ 0 & \boldsymbol{v}^J \neq \boldsymbol{v}_\star^J\left(\boldsymbol{\Omega}\right) \end{cases}. \tag{5.53}$$

In fact, this is a well-known property in statistical mechanics: At zero temperature, the microstate converges in distribution to a realization whose energy level is minimized. The appellation follows the fact that this realization, i.e., $\boldsymbol{v}_{\star}^{J}(\boldsymbol{\Omega})$, is called the *ground state* of the system.

The ground-state property clarifies the connection between our problem and this spin glass: In fact, the ground state is what the RLS-based algorithm recovers, i.e.,

$$\boldsymbol{v}_{\star}^{J}(\boldsymbol{\Omega}) = \hat{\boldsymbol{x}}^{J}. \tag{5.54}$$

In other words, as the temperature goes to zero, the microstate of the corresponding spin glass converges to the signal samples that are recovered via the algorithm, i.e., $\hat{\boldsymbol{x}}^{J}$. Hence, the performance metrics of this sparse recovery algorithm, e.g., the asymptotic distortion, are given as the macroscopic parameters of this spin glass at zero temperature.

The corresponding spin glass shows several other interesting properties. Interested readers are referred to [2, Chapter 3].

#### 5.5.2.1 Asymptotic Distortion as a Macroscopic Parameter

The main purpose of this chapter is to determine the asymptotic distortion. As indicated, this metric can be defined as a macroscopic parameter of the corresponding spin glass. To show that, consider the following macroscopic function:

$$\psi_N\left(\boldsymbol{v}^{J}\right) = \Delta_{\mathrm{V}}\left(\boldsymbol{v}^{J}; \boldsymbol{x}^{J}\right), \tag{5.55}$$

where $\boldsymbol{x}^{J}$ refers to the true signal samples. The macroscopic parameter defined by this function is

$$\bar{\psi} = \lim_{N\uparrow\infty} \frac{1}{N}\mathbb{E}\left\{\psi_N(\boldsymbol{v})\right\} \tag{5.56a}$$

$$= \lim_{N\uparrow\infty} \frac{1}{N}\mathbb{E}\left\{\Delta_{\mathrm{V}}\left(\boldsymbol{v}^{J}; \boldsymbol{x}^{J}\right)\right\}. \tag{5.56b}$$

As the temperature goes to zero, $\beta \uparrow \infty$, the microstate $\boldsymbol{v}^{J}$ converges to $\hat{\boldsymbol{x}}^{J}$. Hence, at zero temperature, we have

$$\bar{\psi} \to \lim_{N\uparrow\infty} \frac{1}{N}\mathbb{E}\left\{\Delta_{\mathrm{V}}\left(\hat{\boldsymbol{x}}^{J}; \boldsymbol{x}^{J}\right)\right\} \tag{5.57a}$$

$$= \lim_{N\uparrow\infty} D_N \tag{5.57b}$$

$$= D. \tag{5.57c}$$

The last equation clarifies how the asymptotic distortion is derived from the corresponding spin glass.

Using the averaging trick, we can find $D$ from the following expected modified free energy in the thermodynamic limit

$$\bar{\mathcal{F}}(\beta, h) = -\lim_{N\uparrow\infty} \frac{1}{\beta N} \mathbb{E}\left\{\log \mathcal{Z}_N\left(\beta, h|\boldsymbol{\Omega}\right)\right\}, \tag{5.58}$$

at zero temperature as

$$D = -\lim_{\beta\uparrow\infty} \beta \frac{\partial}{\partial h} \bar{\mathcal{F}}(\beta, h)\,|_{h=0}, \tag{5.59}$$

where the partition function is given by

$$\mathcal{Z}_N\left(\beta, h|\boldsymbol{\Omega}\right) = \int\limits_{\boldsymbol{v}_j \in \mathbb{X}^N} \exp\left\{-\beta \mathcal{E}\left(\boldsymbol{v}^J|\boldsymbol{\Omega}\right) + h\Delta_{\mathrm{V}}\left(\boldsymbol{v}^J; \boldsymbol{x}^J\right)\right\} \mathrm{d}\boldsymbol{v}^J. \tag{5.60}$$

### 5.5.3 The Replica Method

The variational problem derived in terms of the corresponding spin glass suffers from the same analytical intractability issue we observed in the original problem. In the original problem, we are unable to find the solution of the optimization problem in an analytical form.[15] This is now transformed to a *logarithmic expectation* in (5.58). This is not a trivial task and, hence, keeps the problem still very challenging.

One should note that from the complexity viewpoint, transforming the original problem into the variational form does not change the order of complexity. In fact, for those cases in which the RLS-based algorithm reduces to an NP-hard problem, the calculation of the corresponding free energy also lies in the class of NP-hard problems. One can check this fact by considering the simple example of using an RLS-based algorithm to recover discrete-valued signal samples, i.e., when $\mathbb{X}$ is discrete. In this case, both the original and variational problems are NP-hard. Consequently, transforming the original problem into its variational form only enables us to use the replica method that finds a *prediction* of the asymptotic performance *without* directly solving the problem.

The replica method tries to calculate this logarithmic expectation with a series of tricks. The first trick is to use the Riesz identity [45]:

**Theorem 5.2 (Riesz Identity)** *For a non-negative random variable $X$, we have*

[15] Remember that for some choices of regularization function, e.g., $\ell_0$-norm, this problem is not even numerically solvable.

$$\mathbb{E}\{\log X\} = \lim_{\theta \downarrow 0} \frac{\log \mathbb{E}\left\{X^{\theta}\right\}}{\theta}. \tag{5.61}$$

Using this identity, one can rewrite the logarithmic expectation of (5.58) as

$$\mathbb{E}\{\log \mathcal{Z}_N(\beta, h|\boldsymbol{\Omega})\} = \lim_{\theta \downarrow 0} \frac{\log \mathbb{E}\left\{\mathcal{Z}_N^{\theta}(\beta, h|\boldsymbol{\Omega})\right\}}{\theta}. \tag{5.62}$$

The right-hand side deals with the logarithm of an expectation. The problem is however still challenging since $\theta$ on the right-hand side of the identity is a *real* scalar: The intractability of logarithmic expectation is now transformed to the challenge of *calculating real moments*. Here, the second trick is applied:

**Definition 5.9 (Replica Continuity)** We assume that the moment function[16]

$$f_{\mathrm{M}}(\theta) = \mathbb{E}\left\{\mathcal{Z}_N^{\theta}(\beta, h|\boldsymbol{\Omega})\right\} \tag{5.63}$$

is analytic on the real axis and that this function is analytically continued from the set of natural numbers to the set of positive reals, i.e., $(0, \infty)$.

This second trick is *not mathematically rigorous*. This is why the replica method is often called the *replica trick*. The available results suggest that this is a valid assumption; however, the proof is still an open problem.

Assuming $\theta$ to be an integer finally resolves the intractability issue at the expense of losing mathematical rigor. We now can write the moment function as[17]

$$f_{\mathrm{M}}(\theta) = \mathbb{E}\left\{\mathcal{Z}_N^{\theta}(\beta, h|\boldsymbol{\Omega})\right\} \tag{5.64a}$$

$$= \mathbb{E}\left\{\prod_{a=1}^{\theta} \int \exp\left\{-\beta \mathcal{E}\left(\boldsymbol{v}_a^J|\boldsymbol{\Omega}\right) + h\Delta_{\mathrm{V}}\left(\boldsymbol{v}_a^J; \boldsymbol{x}^J\right)\right\} \mathrm{d}\boldsymbol{v}_a^J\right\} \tag{5.64b}$$

$$= \int \mathbb{E}\left\{\exp\left\{\sum_{a=1}^{\theta} -\beta \mathcal{E}\left(\boldsymbol{v}_a^J|\boldsymbol{\Omega}\right) + h\Delta_{\mathrm{V}}\left(\boldsymbol{v}_a^J; \boldsymbol{x}^J\right)\right\}\right\} \mathrm{d}\boldsymbol{v}_1^J \ldots \mathrm{d}\boldsymbol{v}_{\theta}^J. \tag{5.64c}$$

Note that the expectation in (5.64c) should be taken with respect to the stochastic model given in Sect. 5.2.2.

The latter integral is complicated but tractable. The main remaining task is to calculate this integral and find it as an analytic function in $\theta$. We then plug it into the Riesz identity and take the limits. In the sequel, we give a quick overview on the derivations.

[16] Note that the expectation is taken with respect to all random variables.

[17] In the notation, we drop the integration set for sake of compactness.

## 5.6 The Replica Analysis

The detailed derivation of $f_{\mathrm{M}}(\theta)$ from (5.64c) takes many pages and is out of the scope for this chapter. We hence present the derivation steps and skip the details. Interested readers are referred to [2, Appendices A-D].

We start the derivation by taking expectation with respect to noise. This task is done via basic properties of Gaussian integrals. We then use the results in [27, 32, 33] on the asymptotic limit of *spherical integrals* to calculate the expectation with respect to the sensing matrices. Some short notes on spherical integrals and their asymptotic limits are found in [2, Apendix E]. Finally, we use the law of large numbers to take the expectation with respect to the true signal samples $\boldsymbol{x}^J$.

After taking the expectations, we finally conclude that

$$f_{\mathrm{M}}(\theta) = \int \exp\left\{-N E_{\mathrm{M}}\left(\mathbf{Q}^J, \mathbf{S}^J\right) + \epsilon_N\right\} \mathrm{d}\mathbf{Q}^J \mathrm{d}\mathbf{S}^J, \tag{5.65}$$

where the exponent function $E_{\mathrm{M}}\left(\mathbf{Q}^J, \mathbf{S}^J\right)$ is defined as

$$E_{\mathrm{M}}\left(\mathbf{Q}^J, \mathbf{S}^J\right) = \sum_{j=1}^{J} \left[\mathcal{G}_j\left(\mathbf{T}_j \mathbf{Q}_j\right) + \mathrm{tr}\{\mathbf{S}_j \mathbf{Q}_j\}\right] - \mathcal{M}\left(\mathbf{S}^J\right). \tag{5.66}$$

The matrices $\mathbf{Q}^J$ and $\mathbf{S}^J$ are further defined as

$$\mathbf{Q}^J = (\mathbf{Q}_1, \ldots, \mathbf{Q}_J) \tag{5.67a}$$

$$\mathbf{S}^J = (\mathbf{S}_1, \ldots, \mathbf{S}_J) \tag{5.67b}$$

with $\mathbf{Q}_j$ and $\mathbf{S}_j$ being symmetric $\theta \times \theta$ matrices for $j \in [J]$. The exact definitions of integral measures $\mathrm{d}\mathbf{Q}_j$ and $\mathrm{d}\mathbf{S}_j$ are given in [2, Appendix A]. Moreover, $\epsilon_N$ is a bounded sequence in $N$ that converges to zero as $N$ grows large, and the matrix $\mathbf{T}_j$ is defined as

$$\mathbf{T}_j = \frac{1}{2\lambda_j}\left[\mathbf{I}_\theta - \frac{\beta\sigma_j^2}{\lambda_j + \theta\beta\sigma_j^2}\mathbf{1}_\theta\right], \tag{5.68}$$

where $\mathbf{I}_\theta$ and $\mathbf{1}_\theta$ denote $\theta \times \theta$ identity and all-one matrices, respectively. The components of the exponent function $E_{\mathrm{M}}\left(\mathbf{Q}^J, \mathbf{S}^J\right)$ are further defined as follows:

- The function $\mathcal{G}_j(\cdot)$ is given by

$$\mathcal{G}_i(\mathbf{M}) \coloneqq \int_0^\beta \mathrm{tr}\{\mathbf{M}\mathbf{R}_j(-2\mathbf{M}w)\}\mathrm{d}w \tag{5.69}$$

  for a $\theta \times \theta$ matrix $\mathbf{M}$.

- The function $\mathcal{M}\left(\mathbf{S}^J\right)$ is defined as

$$\mathcal{M}\left(\mathbf{S}^J\right) = \mathbb{E}\left\{\log \int\limits_{\mathbf{v}_j \in \mathbb{X}^\theta} \exp\left\{\Xi\left(\mathbf{v}^J, \mathbf{x}^J | \mathbf{S}^J\right) + h\Delta_{\mathrm{V}}\left(\mathbf{v}^J; \mathbf{x}^J\right)\right\} \mathrm{d}\mathbf{v}^J\right\}, \tag{5.70}$$

where the function $\Xi\left(\mathbf{v}^J, \mathbf{x}^J | \mathbf{S}^J\right)$ is given by

$$\Xi\left(\mathbf{v}^J, \mathbf{x}^J | \mathbf{S}^J\right) = \sum_{j=1}^{J} \left(\mathbf{x}_j - \mathbf{v}_j\right)^{\mathsf{T}} \mathbf{S}_j \left(\mathbf{x}_j - \mathbf{v}_j\right) - \beta u_{\mathrm{V}}\left(\mathbf{v}^J\right). \tag{5.71}$$

In these equations, the notations $\mathbf{v}^J$ and $\mathbf{x}^J$ are defined as $\mathbf{v}^J = (\mathbf{v}_1, \ldots, \mathbf{v}_J)$ and $\mathbf{x}^J = (\mathbf{x}_1, \ldots, \mathbf{x}_J)$, respectively, where $\mathbf{v}_j \in \mathbb{X}^\theta$ and $\mathbf{x}_j = \mathrm{x}_j \mathbf{1}_{\theta \times 1}$ for $j \in [J]$. The vector $\mathbf{1}_{\theta \times 1}$ denotes the $\theta \times 1$ vector of all ones, and $\mathrm{x}_1, \ldots, \mathrm{x}_J$ are correlated random variables distributed jointly with $\mathrm{p}_X(\mathrm{x}_1, \ldots, \mathrm{x}_J)$. It is worth mentioning that the term $u_{\mathrm{V}}\left(\mathbf{v}^J\right)$ decomposes as

$$u_{\mathrm{V}}\left(\mathbf{v}^J\right) = \sum_{a=1}^{\theta} u\left(\mathrm{v}_a^J\right) \tag{5.72}$$

using the decoupling property of the regularization function $u_{\mathrm{V}}(\cdot)$. Here, $\mathrm{v}_a^J = (\mathrm{v}_{1a}, \ldots, \mathrm{v}_{Ja})$ with $\mathrm{v}_{ja}$ denoting the $a$-th entry of $\mathbf{v}_j$.

*Remark 5.3* The definition of $f_{\mathrm{M}}(\theta)$ contains integrals over $N$-dimensional vectors. These integrals are transformed into integrals over $\theta$-dimensional vectors in the final expression. This transform follows several steps and assumptions, e.g., assuming limit exchange and using the asymptotic limit of spherical integrals. The detailed derivations can be followed in [2, Appendix A].

### 5.6.1 General Form of the Solution

The final form of the moment function in (5.65) enables us to apply the saddle-point method to derive the free energy in the thermodynamic limit. After some lines of derivation, we conclude that the asymptotic distortion is given by

$$D = \lim_{\theta \downarrow 0} \lim_{\beta \uparrow \infty} \int\limits_{\mathbf{v}_j \in \mathbb{X}^\theta} \mathbb{E}\left\{\Delta_{\mathrm{V}}\left(\mathbf{v}^J; \mathbf{x}^J\right) \mathrm{q}_\beta\left(\mathbf{v}^J | \mathbf{x}^J\right)\right\} \mathrm{d}\mathbf{v}^J. \tag{5.73}$$

The conditional distribution $\mathrm{q}_\beta\left(\mathbf{v}^J|\mathbf{x}^J\right)$ in this equation is a Boltzmann–Gibbs distribution over the reduced dimension and is defined as

$$\mathrm{q}_\beta\left(\mathbf{v}^J|\mathbf{x}^J\right) = \frac{\exp\left\{-\beta E_0\left(\mathbf{v}^J, \mathbf{x}^J\right)\right\}}{\int\limits_{\mathbf{v}_j \in \mathbb{X}^\theta} \exp\left\{-\beta E_0\left(\mathbf{v}^J, \mathbf{x}^J\right)\right\} \mathrm{d}\mathbf{v}^J}, \tag{5.74}$$

where the exponent function is defined as

$$E_0\left(\mathbf{v}^J, \mathbf{x}^J\right) = \sum_{j=1}^{J} \left(\mathbf{x}_j - \mathbf{v}_j\right)^{\mathsf{T}} \mathbf{R}_j \left(\mathbf{x}_j - \mathbf{v}_j\right) + u_{\mathrm{V}}\left(\mathbf{v}^J\right), \tag{5.75}$$

and the expectation is taken with respect to $\mathbf{x}^J$. The matrix $\mathbf{R}_j$ in the exponent function is further defined as

$$\mathbf{R}_j \coloneqq \mathbf{T}_j \mathrm{R}_j \left(-2\beta \mathbf{T}_j \mathbf{Q}_j^\star\right), \tag{5.76}$$

where the symmetric $\theta \times \theta$ matrix $\mathbf{Q}_j^\star$ for $j \in [J]$ is calculated from the following fixed-point equation:

$$\mathbf{Q}_j^\star = \int\limits_{\mathbf{v}_j \in \mathbb{X}^\theta} \mathbb{E}\left\{\left(\mathbf{x}_j - \mathbf{v}_j\right)\left(\mathbf{x}_j - \mathbf{v}_j\right)^{\mathsf{T}} \mathrm{q}_\beta\left(\mathbf{v}^J|\mathbf{x}^J\right)\right\} \mathrm{d}\mathbf{v}^J. \tag{5.77}$$

*Remark 5.4* To see how (5.77) describes a fixed-point equation, note that the conditional distribution $\mathrm{q}_\beta\left(\mathbf{v}^J|\mathbf{x}^J\right)$ depends on $\mathbf{Q}_j^\star$. As a result, the right-hand side of this identity is calculated as a function of $\mathbf{Q}_j^\star$, and (5.77) describes a fixed-point equation in $\mathbf{Q}_j^\star$.

### 5.6.2 Constructing Parameterized $\mathbf{Q}_j^\star$

The general solution of the replica method is given in terms of the $\theta \times \theta$ matrices $\mathbf{Q}_j^\star$. The reason for having such a solution is simply the *replica continuity* assumption. In this assumption, we postulate that $\theta$ is an *integer*. For an integer $\theta$, having a $\theta \times \theta$ matrix is completely reasonable. Nevertheless, we aim to find the final solution as an analytic function in $\theta$, so that we can use it also for *real* choices of $\theta$.

To find an analytic solution, there exists a classic trick: *Assuming a structure on* $\mathbf{Q}_j^\star$. In this trick, we limit the search to a set of parameterized matrices. The parameterization is considered such that the solution of the fixed-point equation leads to an analytic moment function in $\theta$.

In order to clarify this trick, consider the following illustration: We assume that $\mathbf{Q}_j^\star$ is a $\theta \times \theta$ matrix that is parameterized by $L$ parameters $q^{(1)}, \ldots, q^{(L)}$. This means that

$$\mathbf{Q}_j^\star = W_j\left(q^{(1)}, \ldots, q^{(L)}\right), \tag{5.78}$$

where $W_j(\cdot)$ is a deterministic function that determines a $\theta \times \theta$ matrix for given scalar arguments $q^{(1)}, \ldots, q^{(L)}$. Note that $L$ is an integer whose value is fixed and does not vary by changing $\theta$. By inserting this matrix into the fixed-point equation, a system of $L$ coupled equations in terms of $q^{(1)}, \ldots, q^{(L)}$ is derived. We insert the solution of this equation system into the replica solution and calculate the limits analytically.

With respect to this trick, the following question arises: *What is a meaningful structure for* $\mathbf{Q}_j^\star$? The answer to this question is based on physical intuitions and mathematical investigations of the energy model. These discussions are out of the scope of this overview; however, their results can be directly applied to our study. The investigations in the theory of spin glasses suggest a set of recursively extendable structures drawn from the assumption of *replica symmetry (RS)*. These structures start with a simple symmetric parameterization, known as RS, and then extend to more advanced structures by recursively perturbing RS.

#### 5.6.2.1 Replica Symmetric Solution

RS considers the most basic structure on $\mathbf{Q}_j^\star$, which depends only on two parameters $q_j$ and $\chi_j$, and is given by

$$\mathbf{Q}_j^\star = \frac{\chi_j}{\beta}\mathbf{I}_\theta + q_j \mathbf{1}_\theta. \tag{5.79}$$

A compact way to represent the RS solution is to invoke the *equivalent tunable scalar setting* that is defined below:

**Definition 5.10 (Replica Symmetric Equivalent Scalar Setting)** Let $q_j$ and $\chi_j$ be given for $j \in [J]$. For these parameters, the scalars $\xi_j^2\left(\chi_j, q_j\right)$ and $\tau_j\left(\chi_j\right)$ are defined as

$$\xi_j^2\left(\chi_j, q_j\right) = \left[\mathrm{R}_j\left(-\frac{\chi_j}{\lambda_j}\right)\right]^{-2} \frac{\partial}{\partial \chi_j}\left[\left(\sigma_j^2 \chi_j - \lambda_j q_j\right)\mathrm{R}_j\left(-\frac{\chi_j}{\lambda_j}\right)\right], \tag{5.80}$$

$$\tau_j\left(\chi_j\right) = \frac{\lambda_j}{\mathrm{R}_j\left(-\frac{\chi_j}{\lambda_j}\right)}. \tag{5.81}$$

The RS equivalent scalar setting consists of random variables $\mathrm{x}^J = (\mathrm{x}_1, \ldots, \mathrm{x}_J)$ distributed with $\mathrm{p}_X\left(\mathrm{x}^J\right)$ and their noisy observations $\mathrm{y}_j\left(\chi_j, q_j\right)$ for $j \in [J]$ that are given by

$$\mathrm{y}_j\left(\chi_j, q_j\right) = \mathrm{x}_j + \mathrm{z}_j\left(\chi_j, q_j\right) \tag{5.82}$$

with $\mathrm{z}_j\left(\chi_j, q_j\right)$ being independent Gaussian random variable with zero mean and variance $\xi_j^2\left(\chi_j, q_j\right)$. The estimation of $\mathrm{x}^J$ from its noisy observations is given by

$$\hat{\mathrm{x}}^J\left(\chi_j, q_j\right) = \underset{\mathrm{v}^J \in \mathbb{X}^J}{\operatorname{argmin}} \sum_{j=1}^{J} \frac{1}{2\tau_j\left(\chi_j\right)} \left(\mathrm{y}_j - \mathrm{v}_j\right)^2 + u\left(\mathrm{v}^J\right). \tag{5.83}$$

For this setting, the average distortion is determined as

$$D\left(\chi_j, q_j\right) = \mathbb{E}\left\{\Delta\left(\hat{\mathrm{x}}^J\left(\chi_j, q_j\right); \mathrm{x}^J\right)\right\}, \tag{5.84}$$

where the expectation is taken over all random variables.

The RS equivalent scalar setting describes a multi-terminal *scalar* setting in which the variances of noise terms are tuned by $q_j$ and $\chi_j$. The scalar samples $\mathrm{x}_1, \ldots, \mathrm{x}_J$ in this setting are estimated from the noisy observations via a single-dimension RLS-based algorithm whose regularization parameter is tuned by $\chi_j$. This means that by changing $\chi_j$ and $q_j$, the statistics of this setting and hence its average distortion $D\left(\chi_j, q_j\right)$ are changed. The RS solution states that when $\chi_j$ and $q_j$ are set to specific values, $D\left(\chi_j, q_j\right)$ determines the asymptotic average distortion of the RLS-based algorithm with decoupled regularization function $u(\cdot)$ and regularization parameters $\lambda_1, \ldots, \lambda_J$. These specific values are determined through fixed-point equations stated below:

**Proposition 5.1 (Replica Symmetric Solution)** *Consider the RS equivalent scalar system. The RS solution for asymptotic distortion is given by* $D\left(\chi_j^\star, q_j^\star\right)$, *where* $\chi_j^\star$ *and* $q_j^\star$ *satisfy the following fixed-point equations:*

$$q_j^\star = \mathbb{E}\left\{\left(\hat{x}_j\left(\chi_j^\star, q_j^\star\right) - x_j\right)^2\right\} \tag{5.85a}$$

$$\theta_j^2 \chi_j^\star = \tau_j\left(\chi_j^\star\right) \mathbb{E}\left\{\left(\hat{x}_j\left(\chi_j^\star, q_j^\star\right) - x_j\right) z_j\left(\chi_j^\star, q_j^\star\right)\right\}. \tag{5.85b}$$

*The expectation is taken over all random variables, i.e.,* $x^J$ *and* $z^J\left(\chi_j^\star, q_j^\star\right)$.

It is important to note that the right-hand side of fixed-point equations in (5.85) is *deterministic functions* of $\chi_j^\star$ and $q_j^\star$. In fact, $\hat{\mathrm{x}}_j\left(\chi_j^\star, q_j^\star\right)$ and $\mathrm{z}_j\left(\chi_j^\star, q_j^\star\right)$ are random variables whose statistics are specified by $\chi_j^\star$ and $q_j^\star$. As a result, after taking the expectation, the remaining terms are *deterministic* expressions containing $\chi_j^\star$ and $q_j^\star$.

The RS solution is calculated readily. In fact, (5.83) is a $J$-dimensional optimization that can be solved analytically in various cases. For most well-known RLS-based recovery algorithms, such as *convex* $\ell_p$ and $\ell_{p,q}$-norm minimizations,

the RS solution gives a valid prediction of the asymptotic distortion.[18] Nevertheless, there are a few particular cases in which the RS solution is invalid.[19] This inconsistency is due to the simplicity of the RS structure. For those cases, one needs to break RS.

#### 5.6.2.2 Replica Symmetry Breaking

For scenarios in which the RS solution is not valid, e.g., $\ell_0$-norm minimization, the search for $\mathbf{Q}_j^\star$ is extended to a wider set of parameterized matrices via the replica symmetry breaking (RSB) scheme. This scheme was introduced by Parisi in [42]. The scheme perturbs the RS gradually via a recursive technique. This perturbation is called *breaking*.

**Definition 5.11 (Replica Symmetry Breaking)** Let $\theta$ be an integer multiple of an integer $\zeta$ and $\mathbf{Q}_\ell$ represent a $\zeta \times \zeta$ matrix. RSB finds the new $\theta \times \theta$ matrix $\mathbf{Q}_{\ell+1}$ as

$$\mathbf{Q}_{\ell+1} = \mathbf{I}_{\frac{\theta}{\zeta}} \otimes \mathbf{Q}_\ell + q_{\ell+1}\mathbf{1}_\theta \tag{5.86}$$

for some real scalar $q_{\ell+1}$. Here, $\otimes$ denotes the Kronecker product.

By letting $\mathbf{Q}_0$ be an RS matrix, the RSB structures are recursively generated. The RSB solutions are of more complicated form. We hence skip them and refer interested readers to [2, Chapter 4].

## 5.7 Applications and Numerical Results

The asymptotic characterization of RLS-based recovery algorithms enables us to address several tasks that rise in various applications of sparse recovery. In this section, we briefly go through a few of them. The scope of these applications however is not limited to these instances. We have given more discussions in this respect in [1, 2, 4–16, 46–48].

[18] There are in general various ways to test the validity. The most common test is the *zero-temperature entropy test*; see [2]. For computationally feasible approaches, one can compare the given solution with large-dimensional (but still finite-dimensional) simulations; for instance, see the consistency the RS solution with numerical simulations in [2, Chapter 6] for $\ell_1$-norm minimization.

[19] The invalidity of the solution in these cases is shown by the zero-temperature entropy test. For some particular cases, the RS solution violates the known rigorous bounds.

### 5.7.1 Performance Analysis of Sparse Recovery

The most relevant application of the results is to employ them for asymptotic investigation of sparse recovery algorithms. A long discussion in this respect is found in [2, Chapter 6], as well as [5, 8, 13]. As a particular instance, we employ the asymptotic results to study the impact of spatial correlation in multi-terminal compressive sensing.

For sake of visualization, we consider a simple setting with two terminals. These terminals observe signals $x_1(t)$ and $x_2(t)$ that are jointly sparse. We assume that the joint sparsity follows the *common-innovation* model; see Sect. 5.2.4.3.

The fusion center can recover the sparse signal via two alternative approaches:

1. Since each signal is sparse, the fusion center can use two separate sparse recovery algorithms to recover each sparse signal *individually*.
2. A *joint* recovery algorithm can be used to take into account the spatial correlation among the terminals.

The Slepian–Wolf theorem suggests that joint recovery outperforms an individual scheme [22]. This is in fact a well-known behavior that has been observed in several respects in the context of compressive sensing; see, for example, [18, 21, 24, 31]. To investigate this issue, we consider a sample RLS-based recovery algorithm for each approach and compare their performances using the asymptotic characterization. For the individual approach, we consider the well-known LASSO algorithm. This algorithm is realized by setting the regularization function to

$$u_{\mathrm{V}}(\boldsymbol{v}_1, \boldsymbol{v}_2) = \|\boldsymbol{v}_1\|_1 + \|\boldsymbol{v}_2\|_1. \tag{5.87}$$

As a comparable joint recovery, one can use an RLS-based joint recovery scheme with convex utility, e.g., the group LASSO algorithm in which

$$u_{\mathrm{V}}(\boldsymbol{v}_1, \boldsymbol{v}_2) = \|\boldsymbol{v}_1, \boldsymbol{v}_2\|_{2,1}. \tag{5.88}$$

In the sequel, we use the two-dimensional LASSO technique proposed initially in [8]. This algorithm extends the individual LASSO recovery approach by modifying the regularization function as

$$u_{\mathrm{V}}(\boldsymbol{v}_1, \boldsymbol{v}_2) = \|\boldsymbol{v}_1\|_1 + \|\boldsymbol{v}_2\|_1 + \phi\|\boldsymbol{v}_1 + \alpha\boldsymbol{v}_2\|_1 \tag{5.89}$$

for some scalars $\phi$ and $\alpha$. The intuition behind this algorithm is that any linear combination of jointly sparse signals is also sparse, and its sparsity level depends on the spatial correlation. The study in [8] has shown that this approach outperforms the classic group LASSO technique for the common-innovation joint sparsity model.

Using the RS solution, we can calculate the asymptotic MSE for both approaches. The asymptotic MSE is determined from the RS solution by setting the distortion function to the squared Euclidean distance between the true and recovered pairs.

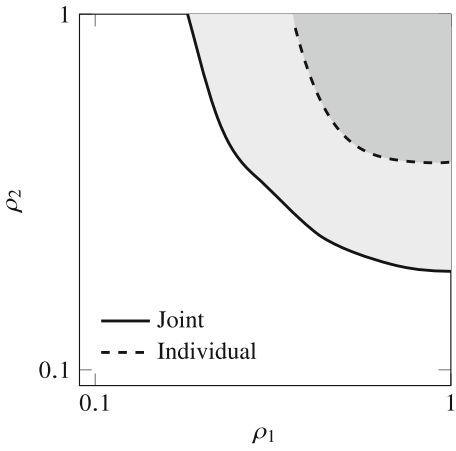



**Fig. 5.1** Rate-distortion region for both joint and individual LASSO schemes, i.e., (5.87) and (5.89), respectively

Using the asymptotic MSE, we plot the *rate-distortion* region for both schemes. It is found by fixing a threshold MSE and finding all pairs of compression rates, i.e., $(\rho_1, \rho_2)$ for which the achievable MSE is smaller than the threshold. This region is shown in Fig. 5.1 for a particular example in which the common part is 30% sparse and each terminal has a 10% sparse innovation component. The tunable factors in both algorithms are optimized to achieve minimal MSE. As the figure shows, using a spatially coupled regularization improves the recovery performance significantly. The Bayesian viewpoint illustrates this observation as follows: The postulated prior distribution of an RLS-based algorithm with spatially coupled regularization takes into account the spatial correlation and hence outperforms the individual approach.

### 5.7.2 Tuning RLS-Based Algorithms

Compressive sensing is not the only application of sparse recovery. In fact, sparse recovery is used in various applications, such as communications, networking, and machine learning; see some instances in [1, 2, 6, 7, 9–12, 14–16, 40, 46–48]. In these applications, there is often a tuning task: Find the *regularization parameters* of an RLS-based algorithm, such that the performance is optimized. This task is readily addressed via the asymptotic characterization of the RLS-based recovery algorithms.

We can illustrate this application by considering a simple example of *spatial modulation*. The details on this example can be followed in [11, 16]. In spatial modulation, the information is encoded in the support of the transmit signal: In each symbol interval, based on the data bits, a subset of available transmit antennas is set on and the remaining are turned off. As a result, the transmit signal is sparse,

and hence, an effective detection scheme at the receiver is to use a sparse recovery algorithm.[20]

The common sparse recovery algorithms used in *spatial modulation* are formulated as RLS-based recovery schemes. Examples are the classic LASSO and *box-LASSO* techniques. We already know the classic LASSO scheme from the previous section. The *box-LASSO* technique is moreover an extension of LASSO in which the set $\mathbb{X}$ is restricted to a box, e.g., $\mathbb{X} = [-B, B]$ for some real $B$. This box restriction is shown to enhance the performance, when we detect discrete-valued signals [34, 51].

One of the challenges in these techniques is to find the optimal regularization parameters, which result in minimum bit error rate. Such a task is usually addressed via iterative tuning techniques. Nevertheless, in high data rates, the tuning techniques impose extra processing load on the system. The asymptotic characterization enables us to address this task analytically and hence avoid the extra load. An instance of tuning via the asymptotic characterization is shown in Fig. 5.2. In this figure, a multiuser uplink scenario is considered in which the LASSO and box-LASSO techniques are used for detection. Here, $P$ denotes the transmit power and $\sigma^2$ is the noise variance at the receiver. The sparsity of the transmit signal is assumed to be 12.5%. The figure shows the optimal regularization parameter, denoted by $\lambda^\star$, against $\log P/\sigma^2$. Although these results are derived via the asymptotic characterization, the study in [16] shows that they closely track the simulation results. Further discussions regarding the tuning of RLS-based algorithms via asymptotic results can be followed in [2, Chapters 6 and 7], as well as [15].

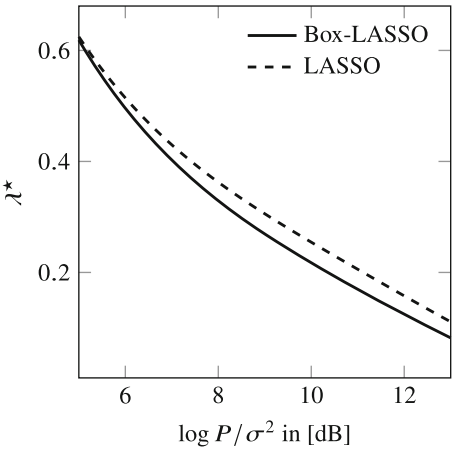



**Fig. 5.2** Optimal regularization parameter for LASSO and box-LASSO

[20] For sake of brevity, we skip the detailed system model. Interested readers are referred to [11, 16] and the references therein.

## 5.8 Summary and Final Discussions

The replica method is a powerful tool for large system analysis, as seen in this chapter. Following the prescription suggested by the replica method, we have found an analytic expression for the asymptotic distortion. The result could not be derived via basic analytical tools. This demonstrates the power of the *replica method*. To keep the contents of this chapter straightforward, we have dropped the detailed derivations and only presented the major steps. The details can be found in [2].

The presented analysis is extendable in various respects and results in various further interesting conclusions. Going through all of these extensions and conclusions is not possible within a single chapter. We hence skip them here and refer interested readers to [2] and the references therein. Nevertheless, to give you a flavor, we conclude this chapter with a few highlights.

### *5.8.1 Decoupling Principle*

Although this chapter focused on the derivation of *asymptotic distortion*, the result can be further used to prove the so-called *decoupling principle*. This principle indicates that in the asymptotic regime the joint distribution of $x_n^J$ and $\hat{x}_n^J$ converges to the one described via an *equivalent* scalar setting, often called the *decoupled setting*. This decoupled setting is shown to consist of an *equivalent additive noise term* and a *decoupled recovery scheme*; see [2, Chapter 5]. The interesting point is that the decoupled recovery scheme remains the same for all solutions, i.e., the RS and RSB solutions, and it is only the distribution of the equivalent noise term that changes. A comprehensive illustration of the decoupling principle and its detailed derivations are given in [2, Chapter 5].

### *5.8.2 Nonuniform Sparsity Patterns*

In various applications, the sparsity of signals varies over time. This form of sparsity is often called *nonuniform*, whereas the normal form is considered *uniform*. For nonuniform sparse signals, the stochastic model of samples is not i.i.d. anymore. They are still independent;[21] however, the joint distribution changes through time. The analysis in this chapter extends to nonuniform patterns by some modifications. Some results in this direction can be followed in [3, 9].

[21] Since temporal correlation is usually avoided by classic sampling approaches.

### 5.8.3 Extensions to Bayesian Estimation

In the Bayesian framework, the considered RLS-based algorithms are seen as MAP estimators. This is however not the only approach for Bayesian inference. In many other applications, e.g., signal processing and machine learning, other forms of Bayesian inference are used, e.g., the minimum MSE estimator or more generally, estimators with minimal posterior distortion; see, for example, [40].

The replica-based analysis in this chapter is readily extended to these estimators as well. The derivations follow the same steps as illustrated in this chapter, i.e., finding a corresponding spin glass and interpreting the desired metrics as its macroscopic parameters. The key difference here is that for other estimators, the desired metrics might be a macroscopic parameter at a *non-zero* temperature.

## 5.9 Bibliographical Notes

Asymptotic analysis of signal recovery schemes roots back to early studies on linear recovery techniques, e.g., studies in [28, 49]. The findings indicate that the asymptotic properties of linear recovery schemes are equivalently described by a simple scalar setting. Müller and Gerstacker conjectured later that similar behavior extends to most nonlinear schemes, as well [38]. This conjecture was originated from the analytic results reported in a series of studies that employed the replica method to derive the asymptotic performance of multiuser detectors. This series of works start with the study by Tanaka in [52]. A key milestone in this direction is achieved in [29], where the authors determine the asymptotic performance metrics of a mismatched minimum MSE recovery scheme. This result is extended to MAP estimators in [44] using standard large deviations techniques.

Early analytic investigations in compressive sensing follow rigorous approaches. An instance is the studies by Donoho and Tanner in [19, 20], in which random geometry is utilized to show the phase transition of linear programming when it is used to perform sparse recovery. A similar approach is taken [17, 50] to study the performance of $\ell_1$-norm minimization for sparse recovery. To address the fundamental limits in compressive sensing, an alternative information-theoretic approach is followed by Wu in [57]; see also [58] and the references therein.

The strong connection between sparse recovery and multiuser detection was initially illustrated in several lines of work; see for example [30]. The study in [44] further extends the replica-based characterizations to address MAP-type sparse recovery algorithms. These initial asymptotic analyses rely on the earlier derivations and hence enclose restricted system models, e.g., single terminal and i.i.d. sensing matrix. These restrictions are addressed in the later lines of work by two different approaches: They either develop a framework by which the results are extended to wider system models, e.g., university laws [41], or they deviate from

the earlier derivations and use the replica method explicitly to derive the asymptotic characteristics; see for instance [35, 36, 54–56].

These analyses were however limited to RS investigations. The complete replica analysis of RLS-based algorithms was given in a series of works in [2, 4, 5, 13] providing both the RS and RSB solutions.

## References


1. Asaad, S., Bereyhi, A., Müller, R.R., Schaefer, R.F.: Joint user selection and precoding in multiuser MIMO systems via group LASSO. In: Proceedings of the IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), Istanbul (2019)
2. Bereyhi, A.: Statistical Mechanics of Regularized Least Squares. Ph.D. Dissertation, Friedrich-Alexander University (2020)
3. Bereyhi, A., Müller, R.R.: Maximum-a-posteriori signal recovery with prior information: applications to compressive sensing. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, pp. 4494–4498 (2018)
4. Bereyhi, A., Müller, R., Schulz-Baldes, H.: RSB decoupling property of MAP estimators. In: Proceedings of the IEEE Information Theory Workshop (ITW),Cambridge, pp. 379–383 (2016)
5. Bereyhi, A., Müller, R.R., Schulz-Baldes, H.: Replica symmetry breaking in compressive sensing. In: Proceedings of the IEEE Information Theory and Applications Workshop (ITA), San Diego, pp. 1–7 (2017)
6. Bereyhi, A., Sedaghat, M.A., Asaad, S., Müller, R.: Nonlinear precoders for massive MIMO systems with general constraints. In: Proceedings of the VDE 21st International ITG Workshop on Smart Antennas (WSA), Berlin, pp. 1–8 (2017)
7. Bereyhi, A., Sedaghat, M.A., Müller, R.R.: Asymptotics of nonlinear LSE precoders with applications to transmit antenna selection. In: Proceedings of the IEEE International Symposium on Information Theory (ISIT), Aachen, pp. 81–85 (2017)
8. Bereyhi, A., Haghighatshoar, S., Müller, R.R.: Theoretical bounds on MAP estimation in distributed sensing networks. In: Proceedings of the IEEE International Symposium on Information Theory (ISIT), Vail (2018)
9. Bereyhi, A., Sedaghat, M.A., Müller, R.R.: RLS recovery with asymmetric penalty: fundamental limits and algorithmic approaches. In: Proceedings of the 2nd International Balkan Conference on Communications and Networking, Podgorica (2018)
10. Bereyhi, A., Sedaghat, M.A., Müller, R.R.: Precoding via approximate message passing with instantaneous signal constraints. In: Proceedings of the International Zurich Seminar on Information and Communication (IZS), Zürich, pp. 128–132 (2018)
11. Bereyhi, A., Asaad, S., Gäde, B., Müller, R.R.: RLS-based detection for massive spatial modulation MIMO. In: Proceedings of the IEEE International Symposium on Information Theory (ISIT), Paris, pp. 1167–1171 (2019)
12. Bereyhi, A., Asaad, S., Müller, R.R., Chatzinotas, S.: RLS precoding for massive MIMO systems with nonlinear front-end. In: Proceedings of the IEEE 20th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), Cannes (2019)
13. Bereyhi, A., Müller, R.R., Schulz-Baldes, H.: Statistical mechanics of MAP estimation: general replica ansatz. IEEE Trans. Inf. Theory **65**(12), 7896–7934 (2019)
14. Bereyhi, A., Jamali, V., Müller, R.R., Fischer, G., Schober, R., Tulino, A.M.: PAPR-limited precoding in massive MIMO systems with reflect- and transmit-array antennas. In: Proceedings of the Asilomar Conference on Signals, Systems, and Computers, Pacific Grove (2019). https://ieeexplore.ieee.org/abstract/document/9048889

15. Bereyhi, A., Sedaghat, M.A., Müller, R.R., Fischer, G.: GLSE precoders for massive MIMO systems: analysis and applications. IEEE Trans. Wirel. Commun. **18**(9), 4450–4465 (2019)
16. Bereyhi, A., Asaad, S., Gäde, B., Müller, R.R., Poor, H.V., "Detection of Spatially Modulated Signals via RLS: Theoretical Bounds and Applications," in IEEE Transactions on Wireless Communications, vol. 21, no. 4, pp. 2291–2304, (2022). https://doi.org/10.1109/TWC.2021.3110839.
17. Chandrasekaran, V., Recht, B., Parrilo, P.A., Willsky, A.S.: The convex geometry of linear inverse problems. Found. Comput. Math. **12**(6), 805–849 (2012)
18. Davies, M.E., Eldar, Y.C.: Rank awareness in joint sparse recovery. IEEE Trans. Inf. Theory **58**(2), 1135–1146 (2012)
19. Donoho, D.L., Tanner, J.: Neighborliness of randomly projected simplices in high dimensions. Proc. Natl. Acad. Sci. **102**(27), 9452–9457 (2005)
20. Donoho, D., Tanner, J.: Counting faces of randomly projected polytopes when the projection radically lowers dimension. J. Am. Math. Soc. **22**(1), 1–53 (2009)
21. Eldar, Y.C., Rauhut, H.: Average case analysis of multichannel sparse recovery using convex relaxation. IEEE Trans. Inf. Theory **56**(1), 505–519 (2009)
22. El Gamal, A., Kim, Y.H.: Network Information Theory. Cambridge University Press, Cambridge (2011)
23. Foucart, S., Rauhut, H.: A Mathematical Introduction to Compressive Sensing. Springer, New York (2013)
24. Gribonval, R., Rauhut, H., Schnass, K., Vandergheynst, P.: Atoms of all channels, unite! average case analysis of multi-channel sparse recovery using greedy algorithms. J. Fourier Anal. Appl. **14**(5–6), 655–687 (2008)
25. Guerra, F., Toninelli, F.L.: The thermodynamic limit in mean field spin glass models. Commun. Math. Phys. **230**(1), 71–79 (2002)
26. Guerra, F., Toninelli, F.L.: The infinite volume limit in generalized mean field disordered models. Markov Processes Relat. Fields **9**, 195–207 (2003)
27. Guionnet, A., Zeitouni, O.: Large deviations asymptotics for spherical integrals. J. Funct. Anal. **188**(2), 461–515 (2002)
28. Guo, D., Rasmussen, L.K., Lim, T.J.: Linear parallel interference cancellation in long-code CDMA multiuser detection. IEEE J. Sel. Areas Commun. **17**(12), 2074–2081 (1999)
29. Guo, D., Verdú, S.: Randomly spread CDMA: asymptotics via statistical physics. IEEE Trans. Inf. Theory **51**(6), 1983–2010 (2005)
30. Guo, D., Baron, D., Shamai, S.: A single-letter characterization of optimal noisy compressed sensing. In: 47th Annual Allerton Conference on Communication, Control, and Computing (Allerton), pp. 52–59 (2009)
31. Haghighatshoar, S.: Multi terminal probabilistic compressed sensing. In: 2014 IEEE International Symposium on Information Theory, pp. 221–225 (2014)
32. Harish-Chandra: Differential operators on a semisimple Lie algebra. Am. J. Math., 87–120 (1957)
33. Itzykson, C., Zuber, J.B.: The planar approximation. II. J. Math. Phys. **21**(3), 411–421 (1980)
34. James, G.M., Paulson, C., Rusmevichientong, P.: The constrained lasso. Proc. Refereed Conf. Proc. **31**, 4945–4950 (2012)
35. Kabashima, Y., Wadayama, T., Tanaka, T.: A typical reconstruction limit for compressed sensing based on $\ell_p$-norm minimization. J. Stat. Mech: Theory Exp. **2009**(09), L09003 (2009)
36. Kabashima, Y., Wadayama, T., Tanaka, T.: Statistical mechanical analysis of a typical reconstruction limit of compressed sensing. In: Proceedings of IEEE International Symposium on Information Theory (ISIT), pp. 1533–1537 (2010)
37. Merhav, N.: Statistical physics and information theory. Found. Trends Commun. Inf. Theory **6**(1–2), 1–212 (2010)
38. Müller, R.R., Gerstacker, W.H.: On the capacity loss due to separation of detection and decoding. IEEE Trans. Inf. Theory **50**(8), 1769–1778 (2004)
39. Müller, R.R., Alfano, G., Zaidel, B.M., de Miguel, R.: Applications of large random matrices in communications engineering. Preprint. arXiv:1310.5479 (2013)

40. Müller, R.R., Bereyhi, A., Mecklenbräuker, C.F.: Oversampled adaptive sensing with random projections: Analysis and algorithmic approaches. In: Proceedings of the IEEE International Symposium on Signal Processing and Information Technology (ISSPIT), Louisville, pp. 336–341 (2018)
41. Oymak, S., Tropp, J.A.: Universality laws for randomized dimension reduction, with applications. Inf. Infer. J IMA **7**(3), 337–446 (2018)
42. Parisi, G.: A sequence of approximated solutions to the SK model for spin glasses. J. Phys. A Math. Gen. **13**(4), L115 (1980)
43. Pastur, L., Shcherbina, M.: Absence of self-averaging of the order parameter in the Sherrington-Kirkpatrick model. J. Stat. Phys. **62**(1–2), 1–19 (1991)
44. Rangan, S., Fletcher, A.K., Goyal, V.: Asymptotic analysis of MAP estimation via the replica method and applications to compressed sensing. IEEE Trans. Inf. Theory **58**(3), 1902–1923 (2012)
45. Riesz, F.: Sur les valeurs moyennes des fonctions. J. Lond. Math. Society **1**(2), 120–121 (1930)
46. Schram, V., Bereyhi, A., Zaech, J.N., Müller, R.R., Gerstacker, W.H.: Approximate message passing for indoor THz channel estimation. In: Proceedings of the 3rd International Balkan Conference on Communications and Networking, Skopje (2019)
47. Sedaghat, M.A., Bereyhi, A., Müller, R.: A new class of nonlinear precoders for hardware efficient massive MIMO systems. In: Proceedings of the IEEE International Conference on Communications (ICC), Paris (2017)
48. Sedaghat, M.A., Bereyhi, A., Müller, R.R.: Least square error precoders for massive MIMO with signal constraints: fundamental limits. IEEE Trans. Wireless Commun. **17**(1), 667–679 (2018)
49. Shamai, S., Verdú, S.: The impact of frequency-flat fading on the spectral efficiency of CDMA. IEEE Trans. Inf. Theory **47**(4), 1302–1327 (2001)
50. Stojnic, M.: Various thresholds for $\ell_1$-optimization in compressed sensing. Preprint. arXiv:0907.3666 (2009)
51. Stojnic, M.: Recovery thresholds for $\ell_1$ optimization in binary compressed sensing. In: Proceedings of the IEEE International Symposium on Information Theory (ISIT), pp. 1593–1597 (2010)
52. Tanaka, T.: A statistical-mechanics approach to large-system analysis of CDMA multiuser detectors. IEEE Trans. Inf. Theory **48**(11), 2888–2910 (2002)
53. Tulino, A.M., Verdú, S., Verdu, S.: Random matrix theory and wireless communications. In: Foundations and Trends$^{TM}$ in Communications and Information Theory, Now Publishers, Boston (2004)
54. Tulino, A.M., Caire, G., Verdu, S., Shamai, S.: Support recovery with sparsely sampled free random matrices. IEEE Trans. Inf. Theory **59**(7), 4243–4271 (2013)
55. Vehkaperä, M., Kabashima, Y., Chatterjee, S.: Analysis of regularized LS reconstruction and random matrix ensembles in compressed sensing. IEEE Trans. Inf. Theory **62**(4), 2100–2124 (2016)
56. Wen, C.K., Zhang, J., Wong, K.K., Chen, J.C., Yuen, C.: On sparse vector recovery performance in structurally orthogonal matrices via lasso. IEEE Trans. Signal Proces. **64**(17), 4519–4533 (2016)
57. Wu, Y.: Shannon theory for compressed sensing. Ph.D. Dissertation, Princeton University (2011)
58. Wu, Y., Verdú, S.: Optimal phase transitions in compressed sensing. IEEE Trans. Inf. Theory **58**(10), 6241–6263 (2012)
59. Zaidel, B.M., Müller, R.R., Moustakas, A.L., de Miguel, R.: Vector precoding for Gaussian MIMO broadcast channels: impact of replica symmetry breaking. IEEE Trans. Inf. Theory **58**(3), 1413–1440 (2012)

# Chapter 6
# Unbiasing in Iterative Reconstruction Algorithms for Discrete Compressed Sensing


**Robert F. H. Fischer and Carmen Sippel**

## 6.1 Introduction

Undoubtedly, *compressed sensing (CS)* [12, 15, 19] is meanwhile a well-established and widespread method in various fields of mathematics, signal processing, and engineering. Thereby, the interest in compressed sensing first arose from a theoretical (mathematical) point of view, but this line of research was closely followed by the study of the use and of the performance of compressed sensing in various practical applications. In this chapter, we study reconstruction algorithms from a communications theory point of view and for the use in communication scenarios.

### *6.1.1 Compressed Sensing Problem and Reconstruction Algorithms*

We consider the following compressed sensing problem: based on the observation $\boldsymbol{y} = [y_1, y_2, \ldots, y_M]^\mathsf{T} \in \mathbb{R}^M$, which is obtained via the known sensing matrix $\boldsymbol{A} = [a_{ji}] \in \mathbb{R}^{M\times N}$, $M < N$, by[1]

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{n}\,, \tag{6.1}$$

[1] Notation: Random variables and random vectors are typeset in sans-serif font; realizations in conventional italic (math) font. Vectors are displayed in bold lower-case letters, matrices in bold upper-case letters. The transpose and the inverse of $\boldsymbol{A}$ are denoted by $\boldsymbol{A}^\mathsf{T}$ and $\boldsymbol{A}^{-1}$, respectively. A diagonal matrix of appropriate size with the entries of the vector $\boldsymbol{a}$ as diagonal elements is denoted by $\mathbf{diag}(\boldsymbol{a})$. $\boldsymbol{I}$ is the identity matrix. The $\ell_p$ norm is written as $||\cdot||_p$. $\mathrm{E}_{\boldsymbol{x}}\{\cdot\}$: (element-wise) expectation w.r.t. random vector $\boldsymbol{x}$. $\mathrm{f}_x(x)$: probability density function of random variable $x$.

R. F. H. Fischer (✉) · C. Sippel
Universität Ulm, Institut für Nachrichtentechnik, Ulm, Germany
e-mail: robert.fischer@uni-ulm.de; carmen.sippel@uni-ulm.de



G. Kutyniok et al. (eds.), *Compressed Sensing in Information Processing*, Applied and Numerical Harmonic Analysis, https://doi.org/10.1007/978-3-031-09745-4_6

the vector $\boldsymbol{x} = [x_1, x_2, \ldots, x_N]^\mathsf{T} \in \mathbb{R}^N$ should be recovered. Thereby, it is assumed that $\boldsymbol{x}$ is sparse, meaning that only a few non-zero components are present. The elements are drawn i.i.d. from a known marginal *probability density function (pdf)* $\mathrm{f}_x(x)$, i.e., $\mathrm{f}_{\boldsymbol{x}}(\boldsymbol{x}) = \prod_{i=1}^{N} \mathrm{f}_x(x_i)$; a Dirac component at $x = 0$ accounts for the sparsity.

In practice, no noise-free measurements will be available. This fact is modeled by the additive noise term $\boldsymbol{n} \in \mathbb{R}^M$. We follow the usual presumptions that the elements of $\boldsymbol{n}$ are i.i.d. Gaussian (marginal pdf $\mathrm{f}_n(n)$) with mean zero, variance $\sigma_n^2$ per component, and that they are independent of the signal $\boldsymbol{x}$.

Although the standard compressed sensing problem is non-convex due to its sparsity constraint, it can be relaxed to an $\ell_1$-based problem [13], which can efficiently be solved by convex optimization techniques, see [9].

Apart from $\ell_1$-based optimization, in the literature, there is a vast amount of algorithms for signal recovery in compressed sensing, such as *orthogonal matching pursuit (OMP)* [49], *compressive sampling matching pursuit (CoSaMP)* [43, 44], *iterative hard thresholding (IHT)* [5, 6], *iterative soft thresholding (IST)* [14], *approximate message passing (AMP)* [3, 17], and *vector AMP (VAMP)* [53, 54] (similar, but not identical approaches are *orthogonal AMP (OAMP)* [38] and *iterative MMSE estimation and soft feedback (IMS)* [63]) to mention only the most prominent ones.

### 6.1.2 Discrete Setting

In the vast majority of the literature on compressed sensing the non-zero elements of $\boldsymbol{x}$ are drawn from the real numbers. However, in a number of communication applications the non-zero elements are deliberately, by design, drawn from a finite set with real-valued elements (e.g., an amplitude-shift keying constellation [50]).

We will discuss iterative reconstruction algorithms and the required processing steps for the general setting, i.e., arbitrary marginal pdfs $\mathrm{f}_x(x)$. However, we will eventually give the respective cost functions and show results from extensive numerical simulations for the particular discrete setting where $x_i \in \{-1,\ 0,\ +1\}$ with probabilities $\{p_1,\ p_0,\ p_1\}$ ($2p_1 + p_0 = 1$ and $p_1 = s/(2N)$ when $s$ denotes the sparsity). The signal pdf is hence given by ($\delta(x)$ denotes the Dirac function)

$$\mathrm{f}_x(x) = p_1\,\delta(x+1) + p_0\,\delta(x) + p_1\,\delta(x-1)\ . \tag{6.2}$$

Notice that for real-valued sensing matrices $\boldsymbol{A} \in \mathbb{R}^{M\times N}$, the measurements $\boldsymbol{y} \in \mathbb{R}^M$ are still real-valued.

Particular examples where discrete-valued sparse signals may beneficially be exploited in communications are *sensor networks*, where $N$ low-activity sensors independently transmit binary data and a fusion center with $M$ antennas has to reconstruct which sensors were active and which data has been transmitted [68].

Further applications are *peak-to-average power reduction* in orthogonal frequency-division multiplexing [23], the detection of pulse-width-modulated signals in *radar* [20], code-book excited linear prediction (CELP) *source coding* [18], and compressed-sensing-based *cryptography* [21].

This *discrete compressed sensing* is related to *model-based compressed sensing* [1]; the signal model is given in form of the distribution of the discrete signal elements. However, it should not be confused with *1-bit compressed sensing* where the elements of the *measurement vector* are quantized, e.g., $\boldsymbol{y} \in \{\pm 1\}^M$, but still $\boldsymbol{x} \in \mathbb{R}^N$, e.g., [8, 31, 70]. This is of particular interest when cheap (one-bit) analog-to-digital (A/D) converters are employed in the acquisition of measurements.

Naturally, the knowledge on the discrete nature of the signal should be utilized in the signal reconstruction. Meanwhile most classical recovery algorithms have been adapted for discrete compressed sensing, e.g., [26, 34, 60–63]. The estimation of a discrete-valued vector is a combinatorial problem in general; it is non-convex, even if the $\ell_0$ constraint is relaxed to an $\ell_1$ one. In [47], an extension of the simplex algorithm, called branch-and-cut algorithm, has been proposed. Unfortunately, these algorithms have a prohibitively high computational complexity and in their analysis typically bounded noise is assumed, for a detailed discussion see [45].

The signal recovery problem in compressed sensing has also been tackled from a *channel coding* perspective, e.g., [11, 65, 69]. The relations are particularly obvious when dealing with discrete compressed sensing. AMP [3, 16, 17, 40] is derived from the generic concept of *message passing*, which, in the form of the *sum-product algorithm*, is very successfully utilized for the decoding of *low-density parity-check codes* [35] (or other sparse graphical models, e.g., [29]). The message-passing approach can be adapted to the situation where an a-priori distribution of the sparse vector is known, cf. [17, 36]. This resulting algorithm is often denoted as Bayesian AMP (BAMP) [2, 51], cf. also generalized AMP (GAMP) [52]. BAMP/GAMP can be used straightforwardly for the discrete scenario.

Typically, in communication scenarios, no perfect signal reconstruction is required but only some *tolerable error ratio* should not be exceeded. Consequently, in the numerical examples we assess the error rate; as typical in digital communications, the order of magnitude which can be achieved is relevant.

### 6.1.3 Outline of the Chapter

In this chapter, we discuss *iterative algorithms* for compressed sensing. We give an overview over the relevant approaches available in the literature and introduce improved processing steps—which we show to be *unbiasing* operations—for the information exchange between the building blocks of the iterative schemes. Unless otherwise stated, the exposition is valid for general a-priori pdfs but we will give numerical results for the discrete case. The comparison of the continuous and discrete case is beyond the scope of this chapter.

The chapter is organized as follows. We review message-passing approaches from the literature in Sect. 6.2 and have a look at how the problem can be separated into feasible parts. In Sect. 6.3, this leads us to known iterative ("turbo") algorithms, where two problems are alternatingly solved. Thereby, the information exchange between both parts is of importance. In Sect. 6.4, this step is discussed and we represent it as an *unbiasing* operation. Using this knowledge, we propose improved VAMP-type algorithms and assess them in Sect. 6.5 by means of numerical simulations. The characterization of the reliability by an average variance or by individual variances is studied.

## 6.2 Problem Statement and Iterative Algorithms

The task of reconstruction algorithms is to produce an *estimate* $\hat{\boldsymbol{x}}$ given the observation $\boldsymbol{y}$, i.e., to *infer* $\boldsymbol{x}$ from $\boldsymbol{y}$, knowing the sensing matrix $\boldsymbol{A}$. The optimum estimate in the *minimum mean-square error (MMSE)* sense is given by the *conditional mean*[2] [33, 48]

$$\hat{\boldsymbol{x}} = \mathrm{E}\{\boldsymbol{x} \mid \boldsymbol{y}\} = \int \boldsymbol{x}\, \mathrm{f}_{\boldsymbol{x}|\boldsymbol{y}}(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x} \,, \tag{6.3}$$

where $\mathrm{f}_{\boldsymbol{x}|\boldsymbol{y}}(\boldsymbol{x})$ is the posterior (conditional) pdf induced by model (6.1). Alternatively, the *maximum-a-posteriori (MAP)* estimate may be sought, which is given by

$$\hat{\boldsymbol{x}} = \operatorname*{argmax}_{\boldsymbol{x}} \mathrm{f}_{\boldsymbol{x}|\boldsymbol{y}}(\boldsymbol{x}) \,. \tag{6.4}$$

Since $\mathrm{f}_{\boldsymbol{x}|\boldsymbol{y}}(\boldsymbol{x}) \geq 0$, w.l.o.g. we can write ($\beta$ is a positive constant)

$$\mathrm{f}_{\boldsymbol{x}|\boldsymbol{y}}(\boldsymbol{x}) = \tfrac{1}{Z}\, \mathrm{e}^{-\beta\, E_{\boldsymbol{y}}(\boldsymbol{x})} \,, \tag{6.5}$$

where $Z = \int \mathrm{e}^{-\beta\, E_{\boldsymbol{y}}(\boldsymbol{x})}\, \mathrm{d}\boldsymbol{x}$ is the so-called *partition function*, which normalizes the distribution. The MAP estimate is then equivalently given as the minimization of some energy function

$$\hat{\boldsymbol{x}} = \operatorname*{argmin}_{\boldsymbol{x}} E_{\boldsymbol{y}}(\boldsymbol{x}) \,. \tag{6.6}$$

In this section, we review message-passing approaches available in the literature and the underlying factorization of the problem at hand in order to solve (6.3)

[2] If no limits are given for integrals, the lower and upper limits are $-\infty$ and $\infty$, respectively.

or (6.4) in practice. The concept of exponential families, which is required in the following sections, is also briefly summarized.

### 6.2.1 Factorization and Message-Passing Approaches

Problems (6.3) and (6.4) cannot be solved straightforwardly if the dimensions $M$ and $N$ are large. Reasonable approaches can be derived when considering the structure of the problem more closely.

Due to the above assumptions (i.i.d. data, additive i.i.d. noise, independent of the data), the conditional pdf can be written as

$$\begin{aligned} \mathrm{f}_{\boldsymbol{x}|\boldsymbol{y}}(\boldsymbol{x}) &= \frac{1}{\mathrm{f}_{\boldsymbol{y}}(\boldsymbol{y})}\, \mathrm{f}_{\boldsymbol{y}|\boldsymbol{x}}(\boldsymbol{y})\, \mathrm{f}_{\boldsymbol{x}}(\boldsymbol{x}) \\ &= c \cdot \mathrm{f}_{\boldsymbol{n}}(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x})\, \mathrm{f}_{\boldsymbol{x}}(\boldsymbol{x}) \\ &= c \cdot \prod\nolimits_{j=1}^{M} \mathrm{f}_{n}(y_j - \boldsymbol{a}_j^{\mathsf{T}}\boldsymbol{x}) \cdot \prod\nolimits_{i=1}^{N} \mathrm{f}_{x}(x_i)\ , \end{aligned} \tag{6.7}$$

where $c$ is a constant and $\boldsymbol{a}_j^{\mathsf{T}}$ is the $j^{\text{th}}$ row of the sensing matrix $\boldsymbol{A} = \left[\boldsymbol{a}_1 \cdots \boldsymbol{a}_M\right]^{\mathsf{T}}$.

Moreover, assuming Gaussian noise, i.e., $\mathrm{f}_{\boldsymbol{n}}(\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}) = \frac{1}{\sqrt{2\pi\sigma_n^2}^M}\, \mathrm{e}^{-||\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}||_2^2/(2\sigma_n^2)}$, the MAP estimate is equivalently given by

$$\hat{\boldsymbol{x}} \;=\; \underset{\boldsymbol{x}}{\operatorname{argmin}} \left( \tfrac{1}{2\sigma_n^2} ||\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}||_2^2 - \log(\mathrm{f}_{\boldsymbol{x}}(\boldsymbol{x})) \right) . \tag{6.8}$$

When a Laplacian prior pdf is assumed $-\log(\mathrm{f}_{\boldsymbol{x}}(\boldsymbol{x})) = \text{const} + \lambda\, ||\boldsymbol{x}||_1$ and (6.8) is specialized to *LASSO (least absolute shrinkage and selection operator)* [66]. Only when assuming a Gaussian prior pdf, (6.8) is a least-squares problem with *Tikhonov regularization* [10], or a *linear MMSE equalization* problem, which can be solved analytically.

#### 6.2.1.1 Message-Passing Approaches

The factorization (6.7) into $M + N$ factors immediately leads to approaches widely used in practice. We now give a brief overview over the main ideas; for the details the reader is referred to the literature.

*Pdf-Based Message Passing* In [39] it is shown that when neglecting the dependencies of one element $x_i$ on the other elements of $\boldsymbol{x}$ (and $y_j$ on $\boldsymbol{y}$), the problem can be dissected into two coupled equations for updating (conditional) pdfs. This ***first step*** establishes a *pdf-based* message passing which is tackled with the *sum-product algorithm* [35, 37]: the nodes are the elements of $\boldsymbol{x}$ accompanied by a

"variable node" update and the elements of $\boldsymbol{y}$ accompanied by a "measurement node" update, respectively. Noteworthy, at all variable nodes, for each measurement node an individual message (pdf) is calculated and sent back (and vice versa) leading to a huge complexity. This approach is still impractical but it leads to interpretations which can subsequently be exploited.

In [67] it is shown that (on cycle-free graphs) the sum-product algorithm converges to a solution $\hat{\boldsymbol{x}}$ which corresponds to *stationary points* of the *(Bethe) free energy* in an associated system ($\boldsymbol{x}$ is the state of $N$ particles, $E_{\boldsymbol{y}}(\boldsymbol{x})$ is the corresponding Hamiltonian). Alternatively, the *Helmholtz free energy* $F_{\mathrm{H}} = -\log(Z)$, where $Z$ is the partition function in (6.5), may be considered which describes the problem from a different point of view, cf. [32]. Having tractable approximations of these energy quantities may, thus, provide approximations of the initial problem.

*Mean and Variance-Based Message Passing* When (implicitly, in the large-system limit) assuming Gaussian random variables, within the iterations only (conditional) means and variances have to be updated. Moreover, for Gaussian pdfs MMSE and MAP criterion coincide; the sum-product algorithm coincides with the *max-product algorithm* [37]. Defining the edge-dependent *residuum* $r_{j,i} \overset{\text{def}}{=} y_j - \sum_{l \neq i} a_{j,l} x_l$, the problem is dissected into "variable nodes" and "residuum nodes" (instead of measurement nodes) [39]. This procedure still has high complexity as individual messages (mean and variance) per edge in the factor graph have to be calculated. These updates follow the philosophy of message passing, where each node passes *extrinsic information* back, i.e., the information *gained* via the other messages.

This ***second step*** is the core idea of many practical inference techniques, such as *expectation propagation (EP)* [41] or *expectation-consistent (EC) approximate inference* [46]. On the one hand, the pdfs to be handled are replaced by pdfs from some family; then only parameters representing the *sufficient statistics* have to be specified. Of special interest are *exponential families* (cf. Sect. 6.2.2), since Gaussian pdfs are special cases thereof. On the other hand, all factors in (6.7), except the currently (in the respective node in the message-passing algorithm) considered one, are replaced by a pdf from the chosen family. Thereby, the local calculations become feasible.

*Approximate Message Passing* Finally, two main modifications lead to the practical algorithm of *"approximate" message passing (AMP)*. First, the edge-dependent messages are written as *node-dependent* (averaged) versions plus some deviation. Second, approximating these deviations via a first-order Taylor series expansion, simple update equations are obtained. For details see [39]. Now, iterations between approximative "variable nodes" and approximative "residuum nodes" are carried out. Only averaged (node-, not edge-dependent) message are passed and only average reliabilities (variances) are tracked. The messages are no longer exact extrinsics but averaged and approximated versions and also not the posteriors of the nodes. AMP is a well performing, low-complexity algorithm; its convergence is well understood via *state evolution*. However, since only approximate quantities are

tracked, no intuitive understanding is possible, cf. [55]. In particular, the *Onsager term* [39], known from statistical physics, has no direct interpretation.

Although being derived from the "message-passing" philosophy—where the processing is done fine-grained and based on a very local view of the node—, AMP iterates between two parts in a "turbo" fashion—a very global view on the vectors $\boldsymbol{x}$ and $\boldsymbol{r}$ is taken; no per-edge messages are calculated. This ***third step*** is viewed in a more general way in the following.

### 6.2.1.2 Partitioning of the Problem

The above discussion reveals the general principle that an intractable problem is transformed into a tractable one by (i) treating groups of factors in the factorization (6.7) jointly and (ii) substituting such groups by a pdf from a chosen family. By iterating over the factors in a message-passing approach, the desired solution is found iteratively.

Besides treating all factors individually (as done in mean- and variance-based message passing), the most obvious partitioning of the problem is to combine either all $M$ factors belonging to the observations into $\mathrm{f}_{\boldsymbol{n}}(\boldsymbol{n})$ or all $N$ factors pertaining to the variables into $\mathrm{f}_{\boldsymbol{x}}(\boldsymbol{x})$. The extreme case of considering both combinations and resorting to the two-factor representation (6.7) lead to the "turbo" view and is studied in detail subsequently. The corresponding factor graphs are depicted in Fig. 6.1. These graphical tools visualize the dependencies of the factors (rectangular nodes) of the variables (circle nodes); an edge symbolizes that the variable is an argument of the factor [35]. The factors corresponding to the prior knowledge of the signal (signal constraints) are shown on the top of the figures, whereas the factors corresponding to the observations (channel constraints) are shown at the bottom.

The "mixed" approaches, not shown in the figure and not discussed in this chapter, enable some degree of freedom in the order the factors are processed (scheduling). This can be utilized for an optimized sequential update.

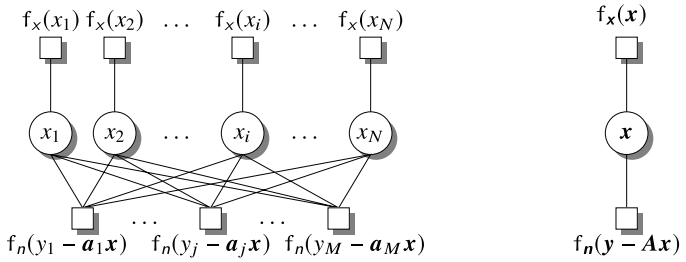



**Fig. 6.1** Factor graphs corresponding to the message-passing view (left: all factors are treated individually) and to the turbo view (right: the factors corresponding to signal and channel constraints, respectively, are combined)

For example, when the prior factors are kept separately while combining the factors belonging to the observations, a sequential processing of the variables $x_i$ is enabled, e.g., [58]. Compared to the factorization with only two factors, where a single variable $x_i$ is processed at the same time as all other variables, this has the advantage that the reliability of this variable can benefit from insight into previously considered variables leading to faster convergence. This also means that the variance cannot be tracked on average for the signal vector, but, instead, individually per $x_i$.

In cases where the prior pdf is not completely factorizable, see, e.g., [4], this factorization does not go down to the individual variables; nevertheless, it can be applied to the respective compounds of variables.

### 6.2.2 Exponential Families

We now give a brief review of exponential families, which are well-suited for the use as substitute pdfs in iterative schemes.

A pdf of an $N$-dimensional random vector $\boldsymbol{x}$ is member of an *exponential family* if it can be written as [42]

$$\mathrm{f}_{\boldsymbol{x}}(\boldsymbol{x}) = \frac{1}{Z(\boldsymbol{\theta})}\, f(\boldsymbol{x})\, \mathrm{e}^{\boldsymbol{\theta}^{\mathsf{T}} \boldsymbol{g}(\boldsymbol{x})} \,. \tag{6.9}$$

Thereby, $f(\boldsymbol{x})$ can be any non-negative real-valued function, $\boldsymbol{\theta} \in \mathbb{R}^n$ represents the *natural parameters*, $\boldsymbol{g}(\boldsymbol{x}) \in \mathbb{R}^n$ is a vector-valued function of $\boldsymbol{x}$ which reflects the *sufficient statistics* of $\boldsymbol{x}$, and $Z(\boldsymbol{\theta})$ is the so-called *partition function* which normalizes the pdf. This important class of pdfs encompasses a wide range of common distributions, in particular, the Gaussian one.

If we specify first- and second-order moments by choosing ($\Lambda > 0$)

$$\boldsymbol{g}(\boldsymbol{x}) = \left[x_1,\ \ldots,\ x_N,\ -\tfrac{1}{2}\textstyle\sum_i x_i^2\right]^{\mathsf{T}} , \tag{6.10}$$

$$\boldsymbol{\theta} = \left[\lambda_1,\ \ldots,\ \lambda_N,\quad \Lambda \quad\right]^{\mathsf{T}} , \tag{6.11}$$

for $f(\boldsymbol{x}) = 1$ a Gaussian pdf which is rotationally invariant about the mean is specified; the $N$-dimensional mean and a single (average) variance characterize the pdf—we call this case *average variance (AvgV)*. Alternatively, we can choose ($\Lambda_i > 0$)

$$\boldsymbol{g}(\boldsymbol{x}) = \left[x_1,\ \ldots,\ x_N,\ -\tfrac{1}{2}x_1^2,\ \ldots,\ -\tfrac{1}{2}x_N^2\right]^{\mathsf{T}} , \tag{6.12}$$

$$\boldsymbol{\theta} = \left[\lambda_1,\ \ldots,\ \lambda_N,\quad \Lambda_1\ ,\ \ldots,\quad \Lambda_N\ \right]^{\mathsf{T}} ; \tag{6.13}$$

here (for $f(\boldsymbol{x}) = 1$) a Gaussian pdf with individual variances per dimension is specified; the $N$-dimensional mean and the $N$ individual variances characterize the pdf—we call this case *individual variances (IndV)*. Conveniently, we define

$$\boldsymbol{\lambda} = \left[\lambda_1, \ \ldots, \ \lambda_N\right]^{\mathsf{T}}, \qquad \begin{array}{ll} \boldsymbol{\Lambda} = \Lambda \boldsymbol{I}\,, & \text{AvgV} \\ \boldsymbol{\Lambda} = \mathbf{diag}(\Lambda_1, \ldots, \Lambda_N)\,, & \text{IndV} \end{array} . \tag{6.14}$$

Please note that exponential families have the convenient property that [42]

$$\boldsymbol{\mu} \overset{\text{def}}{=} \mathrm{E}_{\boldsymbol{x}}\{\boldsymbol{g}(\boldsymbol{x})\} = \left.\frac{\partial \log(Z(\boldsymbol{\chi}))}{\partial \boldsymbol{\chi}}\right|_{\boldsymbol{\chi}=\boldsymbol{\theta}} . \tag{6.15}$$

Hence, the vector $\boldsymbol{\mu}$ contains the means $m_i = \mathrm{E}\{x_i\}$, $i = 1, \ldots, N$, and quantities from which either the average variance $\sigma^2_{\text{avg}} = \frac{1}{N}\sum_{i=1}^{N} \mathrm{E}\{(x_i - m_i)^2\}$ or the individual variances $\sigma_i^2 = \mathrm{E}\{(x_i - m_i)^2\}$ can be deduced. Moreover, natural parameters and variances are connected by

$$\begin{array}{lll} \lambda_i = \dfrac{m_i}{\Lambda}\,, & \Lambda = \dfrac{1}{\sigma^2_{\text{avg}}}\,, & \text{AvgV} \\ \lambda_i = \dfrac{m_i}{\Lambda_i}\,, & \Lambda_i = \dfrac{1}{\sigma_i^2}\,, & \text{IndV} \end{array} . \tag{6.16}$$

Remarkably, for Gaussian pdfs $\mathrm{f}_{\boldsymbol{x}|\boldsymbol{y}}(\boldsymbol{x})$, i.e., pdfs from an exponential family with parameterization (6.10), (6.11) or (6.12), (6.13), MMSE and MAP criterion coincide as

$$\mathrm{E}_{\boldsymbol{x}}\{\boldsymbol{x}|\boldsymbol{y}\} = \underset{\boldsymbol{x}}{\operatorname{argmax}}\, \mathrm{f}_{\boldsymbol{x}|\boldsymbol{y}}(\boldsymbol{x}) \,. \tag{6.17}$$

Finally, we note that a given pdf can be *projected* onto an exponential family; the pdf is assumed to be of the form $\mathrm{e}_{\boldsymbol{x}}(\boldsymbol{x}) = \frac{1}{Z(\boldsymbol{\theta})} h(\boldsymbol{x}) \mathrm{e}^{\boldsymbol{\theta}^{\mathsf{T}} \boldsymbol{g}(\boldsymbol{x})}$. Thereby, the projection is done such that the *Kullback–Leibler divergence* $D(\cdot||\cdot)$ between $\mathrm{f}_{\boldsymbol{x}|\boldsymbol{y}}(\boldsymbol{x})$ and $\mathrm{e}_{\boldsymbol{x}}(\boldsymbol{x})$ is minimized, i.e.,

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmin}}\, D(\mathrm{f}_{\boldsymbol{x}|\boldsymbol{y}}(\boldsymbol{x})||\mathrm{e}_{\boldsymbol{x}}(\boldsymbol{x})) = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \int \mathrm{f}_{\boldsymbol{x}|\boldsymbol{y}}(\boldsymbol{x}) \log\left(\frac{\mathrm{f}_{\boldsymbol{x}|\boldsymbol{y}}(\boldsymbol{x})}{\mathrm{e}_{\boldsymbol{x}}(\boldsymbol{x})}\right) \mathrm{d}\boldsymbol{x} \,. \tag{6.18}$$

It is straightforward to show that this minimization is equivalent to adjusting $\boldsymbol{\theta}$ such that the moments (defined by $\boldsymbol{g}(\boldsymbol{x})$) of $\mathrm{e}_{\boldsymbol{x}}(\boldsymbol{x})$ coincide with the ones of the initial pdf $\mathrm{f}_{\boldsymbol{x}|\boldsymbol{y}}(\boldsymbol{x})$. This fact also holds the other way round: if the moments match, the Kullback–Leibler divergence between the two involved pdfs is minimized [56]. Exponential families thus give the best representation under a moment constraint.

## 6.3 Expectation-Consistent Approximate Inference

As shown in the last section, the simplest way of factorizing the posterior pdf is given by separating the "signal" and the "channel" part, i.e., (6.7) can be rewritten as ($\frac{1}{Z} = c\, \exp\left(||\boldsymbol{y}||_2^2/(2\sigma_n^2)\right)$)

$$\mathrm{f}_{\boldsymbol{x}|\boldsymbol{y}}(\boldsymbol{x}) = \frac{1}{Z} \underbrace{\prod_{i=1}^{N} \mathrm{f}_X(x_i)}_{f_\mathrm{s}(\boldsymbol{x})} \underbrace{\exp\left(\boldsymbol{x}^\mathsf{T}\boldsymbol{F}\boldsymbol{x} + \boldsymbol{f}_y^\mathsf{T}\boldsymbol{x}\right)}_{f_\mathrm{c}(\boldsymbol{x})}, \tag{6.19}$$

where $\boldsymbol{f}_y \overset{\text{def}}{=} \frac{1}{\sigma_n^2}\boldsymbol{A}^\mathsf{T}\boldsymbol{y}$ and $\boldsymbol{F} \overset{\text{def}}{=} -\frac{1}{2\sigma_n^2}\boldsymbol{A}^\mathsf{T}\boldsymbol{A}$, and $Z$ is the partition function. For a factorization into two general factors, Opper and Winther [46] proposed a framework called *expectation-consistent (EC) approximate inference*; subsequently it was generalized in [25]. Applying this framework to our compressed sensing problem, diverse practical recovery algorithms result.

In this section, we briefly review the derivation of the algorithms and address the consequences. Two classes of recovery schemes which emerge from the framework are discussed in more detail. The exposition is valid for any prior pdf $\mathrm{f}_X(x)$; however, the associated cost functions are eventually stated for the particular discrete setting (6.2). Results from numerical simulations are postponed to Sect. 6.5.

### 6.3.1 Derivation and Optimization Procedure

As motivated in the above review of message-passing approaches, instead of calculating $\mathrm{E}_{\boldsymbol{x}}\{\boldsymbol{x} \mid \boldsymbol{y}\}$ directly, the partition function $Z$ is often considered. To that end, we choose a vector-valued function $\boldsymbol{g}(\boldsymbol{x})$ which represents the moments (usually mean and variance) we want to estimate/track within the algorithm (cf. Sect. 6.2.2). Then, by expanding with 1, the partition function can be written as

$$\begin{aligned} Z &= \int f_\mathrm{s}(\boldsymbol{x})\, f_\mathrm{c}(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x} \\ &= Z_\mathrm{s}(\boldsymbol{\theta}_\mathrm{s}) \int f_\mathrm{c}(\boldsymbol{x})\, \mathrm{e}^{-\boldsymbol{\theta}_\mathrm{s}^\mathsf{T}\boldsymbol{g}(\boldsymbol{x})} \frac{1}{Z_\mathrm{s}(\boldsymbol{\theta}_\mathrm{s})} f_\mathrm{s}(\boldsymbol{x})\, \mathrm{e}^{\boldsymbol{\theta}_\mathrm{s}^\mathsf{T}\boldsymbol{g}(\boldsymbol{x})}\, \mathrm{d}\boldsymbol{x}\,. \end{aligned} \tag{6.20}$$

As $Z$ is also intractable, a manageable approximation is desired. To that end, the *signal part* $\frac{1}{Z_\mathrm{s}(\boldsymbol{\theta}_\mathrm{s})} f_\mathrm{s}(\boldsymbol{x})\mathrm{e}^{\boldsymbol{\theta}_\mathrm{s}^\mathsf{T}\boldsymbol{g}(\boldsymbol{x})}$ of the integrand is replaced by $\frac{1}{Z_\mathrm{o}(\boldsymbol{\theta}_\mathrm{o})}\mathrm{e}^{\boldsymbol{\theta}_\mathrm{o}^\mathsf{T}\boldsymbol{g}(\boldsymbol{x})}$, which is called the *overlap* [46] (both are valid pdfs). This leads to

$$Z \approx Z_{\mathrm{EC,s}}(\boldsymbol{\theta}_\mathrm{s}, \boldsymbol{\theta}_\mathrm{o}) = Z_\mathrm{s}(\boldsymbol{\theta}_\mathrm{s}) \int f_\mathrm{c}(\boldsymbol{x})\mathrm{e}^{-\boldsymbol{\theta}_\mathrm{s}^\mathsf{T}\boldsymbol{g}(\boldsymbol{x})} \frac{1}{Z_\mathrm{o}(\boldsymbol{\theta}_\mathrm{o})}\mathrm{e}^{\boldsymbol{\theta}_\mathrm{o}^\mathsf{T}\boldsymbol{g}(\boldsymbol{x})}\, \mathrm{d}\boldsymbol{x}$$

with $\boldsymbol{\theta}_\mathrm{c} \overset{\mathrm{def}}{=} \boldsymbol{\theta}_\mathrm{o} - \boldsymbol{\theta}_\mathrm{s}$

$$= \frac{Z_\mathrm{s}(\boldsymbol{\theta}_\mathrm{s})}{Z_\mathrm{o}(\boldsymbol{\theta}_\mathrm{o})} \int f_\mathrm{c}(\boldsymbol{x}) \mathrm{e}^{\boldsymbol{\theta}_\mathrm{c}^\mathsf{T} \boldsymbol{g}(\boldsymbol{x})} \, \mathrm{d}\boldsymbol{x} = \frac{Z_\mathrm{s}(\boldsymbol{\theta}_\mathrm{s}) \, Z_\mathrm{c}(\boldsymbol{\theta}_\mathrm{c})}{Z_\mathrm{o}(\boldsymbol{\theta}_\mathrm{o})} \,, \tag{6.21}$$

where $\frac{1}{Z_\mathrm{c}(\boldsymbol{\theta}_\mathrm{c})} f_\mathrm{c}(\boldsymbol{x}) \mathrm{e}^{\boldsymbol{\theta}_\mathrm{c}^\mathsf{T} \boldsymbol{g}(\boldsymbol{x})}$ is the *channel part*. The partition functions of the three involved pdfs (which are all members of an exponential family and, thus, characterized by the parameter vectors $\boldsymbol{\theta}_\mathrm{s}$, $\boldsymbol{\theta}_\mathrm{c}$, and $\boldsymbol{\theta}_\mathrm{o}$, respectively) are given by

$$Z_\mathrm{s}(\boldsymbol{\theta}_\mathrm{s}) = \int f_\mathrm{s}(\boldsymbol{x}) \, \mathrm{e}^{\boldsymbol{\theta}_\mathrm{s}^\mathsf{T} \boldsymbol{g}(\boldsymbol{x})} \, \mathrm{d}\boldsymbol{x} \,, \tag{6.22}$$

$$Z_\mathrm{c}(\boldsymbol{\theta}_\mathrm{c}) = \int f_\mathrm{c}(\boldsymbol{x}) \, \mathrm{e}^{\boldsymbol{\theta}_\mathrm{c}^\mathsf{T} \boldsymbol{g}(\boldsymbol{x})} \, \mathrm{d}\boldsymbol{x} \,, \tag{6.23}$$

$$Z_\mathrm{o}(\boldsymbol{\theta}_\mathrm{o}) = \int \mathrm{e}^{\boldsymbol{\theta}_\mathrm{o}^\mathsf{T} \boldsymbol{g}(\boldsymbol{x})} \, \mathrm{d}\boldsymbol{x} \,. \tag{6.24}$$

Instead of treating the partition function, the *negative log-partition function*

$$-\log(Z_{\mathrm{EC,s}}(\boldsymbol{\theta}_\mathrm{s}, \, \boldsymbol{\theta}_\mathrm{o})) = -\log\big(Z_\mathrm{s}(\boldsymbol{\theta}_\mathrm{s})\big) - \log\big(Z_\mathrm{c}(\boldsymbol{\theta}_\mathrm{o} - \boldsymbol{\theta}_\mathrm{s})\big) + \log\big(Z_\mathrm{o}(\boldsymbol{\theta}_\mathrm{o})\big) \tag{6.25}$$

may be considered. Since $\boldsymbol{\theta}_\mathrm{c} = \boldsymbol{\theta}_\mathrm{o} - \boldsymbol{\theta}_\mathrm{s}$, only two free parameters are present.

Noteworthy, instead of approximating the signal part in the integrand, one can alternatively replace the channel part by the overlap. This leads to the adequate expression

$$-\log(Z_{\mathrm{EC,c}}(\boldsymbol{\theta}_\mathrm{c}, \, \boldsymbol{\theta}_\mathrm{o})) = -\log\big(Z_\mathrm{s}(\boldsymbol{\theta}_\mathrm{o} - \boldsymbol{\theta}_\mathrm{c})\big) - \log\big(Z_\mathrm{c}(\boldsymbol{\theta}_\mathrm{c})\big) + \log\big(Z_\mathrm{o}(\boldsymbol{\theta}_\mathrm{o})\big) \,. \tag{6.26}$$

Obviously, $Z_{\mathrm{EC,s}}$ and $Z_{\mathrm{EC,c}}$—and thus the therefrom calculated estimate—are only sensible approximations if the parameters $\boldsymbol{\theta}_\mathrm{s}$, $\boldsymbol{\theta}_\mathrm{o}$ (or $\boldsymbol{\theta}_\mathrm{c}$, $\boldsymbol{\theta}_\mathrm{o}$) are tuned suitably. In [46] it is argued that the parameters should be adjusted such that $-\log(Z_{\mathrm{EC,s}})$ ($-\log(Z_{\mathrm{EC,c}})$) is stationary.

A practical approach is to do this optimization iteratively. First, given $\boldsymbol{\theta}_\mathrm{o}$, as the negative log-partition function is a concave function in $\boldsymbol{\theta}_\mathrm{s}$ (or $\boldsymbol{\theta}_\mathrm{c}$), a *maximization* w.r.t. $\boldsymbol{\theta}_\mathrm{s}$ (or $\boldsymbol{\theta}_\mathrm{c}$) has to be performed—the unique *maximizer* is searched (subsequently we consider $Z_{\mathrm{EC,s}}$; for $Z_{\mathrm{EC,c}}$ the procedure is the same)

$$\begin{aligned} \boldsymbol{\theta}_\mathrm{s}^* &= \operatorname*{argmax}_{\boldsymbol{\theta}_\mathrm{s}} \Big\{ \underbrace{-\log\big(Z_\mathrm{s}(\boldsymbol{\theta}_\mathrm{s})\big) - \log\big(Z_\mathrm{c}(\boldsymbol{\theta}_\mathrm{o} - \boldsymbol{\theta}_\mathrm{s})\big)}_{L_\mathrm{s}(\boldsymbol{\theta}_\mathrm{s})} + \log\big(Z_\mathrm{o}(\boldsymbol{\theta}_\mathrm{o})\big) \Big\} \\ &= \operatorname*{argmax}_{\boldsymbol{\theta}_\mathrm{s}} \big\{ L_\mathrm{s}(\boldsymbol{\theta}_\mathrm{s}) \big\} \,, \end{aligned} \tag{6.27}$$

with the obvious definition of the cost function $L_\mathrm{s}(\boldsymbol{\theta}_\mathrm{s})$. A necessary condition is $\frac{\partial}{\partial \boldsymbol{\theta}_\mathrm{s}} L_\mathrm{s}(\boldsymbol{\theta}_\mathrm{s}) \stackrel{!}{=} \mathbf{0}$, which, considering the property (6.15) of exponential families, actually requires

$$\begin{aligned}\frac{\partial}{\partial \boldsymbol{\theta}_\mathrm{s}} L_\mathrm{s}(\boldsymbol{\theta}_\mathrm{s}) &= \frac{\partial}{\partial \boldsymbol{\theta}_\mathrm{s}} \Big( -\log\big(Z_\mathrm{s}(\boldsymbol{\theta}_\mathrm{s})\big) - \log\big(Z_\mathrm{c}(\boldsymbol{\theta}_\mathrm{o} - \boldsymbol{\theta}_\mathrm{s})\big)\Big) \\ &= -\boldsymbol{\mu}_\mathrm{s}(\boldsymbol{\theta}_\mathrm{s}) + \boldsymbol{\mu}_\mathrm{c}(\boldsymbol{\theta}_\mathrm{o} - \boldsymbol{\theta}_\mathrm{s}) \stackrel{!}{=} \mathbf{0}\,. \end{aligned} \tag{6.28}$$

Hence, the optimization problem (6.27) is equivalent to matching the moments of the signal and channel part. Contrary to what is stated in [46], the moment matching is not an additional constraint. In summary, in the first step, $\boldsymbol{\theta}_\mathrm{s}$ (or $\boldsymbol{\theta}_\mathrm{c}$) is adjusted either such that $L_\mathrm{s}(\boldsymbol{\theta}_\mathrm{s})$ (analogously $L_\mathrm{c}(\boldsymbol{\theta}_\mathrm{c})$) is maximized or, alternatively, such that the moments of the signal and channel part match.

Second, given $\boldsymbol{\theta}_\mathrm{s}^*$ (or $\boldsymbol{\theta}_\mathrm{c}^*$), the parameter $\boldsymbol{\theta}_\mathrm{o}$ has to be adjusted such that $-\log(Z_{\mathrm{EC,s}})$ (or $-\log(Z_{\mathrm{EC,c}})$) is stationary. This leads to

$$\begin{aligned}\frac{\partial}{\partial \boldsymbol{\theta}_\mathrm{o}} \Big( -\log\big(Z_{\mathrm{EC,s}}(\boldsymbol{\theta}_\mathrm{s}^*, \boldsymbol{\theta}_\mathrm{o})\big)\Big) &= \frac{\partial}{\partial \boldsymbol{\theta}_\mathrm{o}} \Big(\mathrm{const} - \log\big(Z_\mathrm{c}(\boldsymbol{\theta}_\mathrm{o} - \boldsymbol{\theta}_\mathrm{s}^*)\big) + \log\big(Z_\mathrm{o}(\boldsymbol{\theta}_\mathrm{o})\big)\Big) \\ &= \mathbf{0} - \boldsymbol{\mu}_\mathrm{c}(\boldsymbol{\theta}_\mathrm{o} - \boldsymbol{\theta}_\mathrm{s}^*) + \boldsymbol{\mu}_\mathrm{o}(\boldsymbol{\theta}_\mathrm{o}) \\ \text{and obeying (6.28)} \qquad &= -\boldsymbol{\mu}_\mathrm{s}(\boldsymbol{\theta}_\mathrm{s}^*) + \boldsymbol{\mu}_\mathrm{o}(\boldsymbol{\theta}_\mathrm{o}) \stackrel{!}{=} \mathbf{0}\,. \end{aligned} \tag{6.29}$$

Again, as above, the optimization problem is equivalent to matching the moments of the signal part and the overlap. Since $\boldsymbol{\mu}_\mathrm{s}$ is given and the expectation parameters and the natural parameters of the overlap have a simple connection (cf. (6.16)), this can be done immediately. For example, when $\boldsymbol{g}(\boldsymbol{x})$ and $\boldsymbol{\theta}_\mathrm{o}$ are chosen according to (6.10) and (6.11), we have

$$\Lambda_\mathrm{o} = \frac{1}{\sigma_\mathrm{s}^2}\,, \qquad \lambda_{\mathrm{o},i} = \frac{m_{\mathrm{s},i}}{\sigma_\mathrm{s}^2}\,, \qquad i = 1, \ldots, N\,. \tag{6.30}$$

Finally, the means give the desired estimate

$$\hat{\boldsymbol{x}} = [m_{\mathrm{s},1},\ \ldots,\ m_{\mathrm{s},N}]^\mathsf{T}\,. \tag{6.31}$$

Noteworthy, any function $\boldsymbol{g}(\boldsymbol{x})$ which defines the exponential family can be used in principle. Of special interest are the parameterizations (6.10) and (6.12), which in [46] are called "uniform diagonalization" and "vector-valued diagonalization," respectively. Here, Gaussian pdfs are utilized and either an average variance $\sigma_\mathrm{avg}^2$ or individual variances $\sigma_1^2, \ldots, \sigma_N^2$ are tracked to characterize reliabilities.

### 6.3.2 Algorithms

The steps for adjusting the parameters $\boldsymbol{\theta}_\mathrm{s}$ (or $\boldsymbol{\theta}_\mathrm{c}$) and $\boldsymbol{\theta}_\mathrm{o}$ directly lead to two classes of algorithms which are subsequently discussed in more detail and which are specialized to the discrete setting.

#### 6.3.2.1 Optimization: $\mathrm{ECopt_s}$, $\mathrm{ECopt_c}$ and $\mathrm{ECseq_s}$, $\mathrm{ECseq_c}$

The maximization of the concave cost function $L_\mathrm{s}(\boldsymbol{\theta}_\mathrm{s})$ (or $L_\mathrm{c}(\boldsymbol{\theta}_\mathrm{c})$) may be replaced by a minimization of the *convex function* $-2\,L_\mathrm{s}(\boldsymbol{\theta}_\mathrm{s})$ (the scaling by the factor 2 is introduced for convenience). To that end, any convex optimization algorithm can be applied, see, e.g., [9]. For model (6.1)/pdf (6.19), the function to be minimized in the first step either reads

$$\begin{aligned} -2\,L_\mathrm{s}(\boldsymbol{\theta}_\mathrm{s}) \quad &= 2\,\log\big(Z_\mathrm{s}(\boldsymbol{\theta}_\mathrm{s})\big) - \log\Big(\det(\boldsymbol{\Lambda}_\mathrm{o}-\boldsymbol{\Lambda}_\mathrm{s}+\boldsymbol{F})\Big) \\ &\quad + (\boldsymbol{\lambda}_\mathrm{o}-\boldsymbol{\lambda}_\mathrm{s}+\boldsymbol{f}_{\boldsymbol{y}})^\mathsf{T}(\boldsymbol{\Lambda}_\mathrm{o}-\boldsymbol{\Lambda}_\mathrm{s}+\boldsymbol{F})^{-1}(\boldsymbol{\lambda}_\mathrm{o}-\boldsymbol{\lambda}_\mathrm{s}+\boldsymbol{f}_{\boldsymbol{y}})\,, \end{aligned} \tag{6.32}$$

$$\begin{aligned} \text{or} \quad -2\,L_\mathrm{c}(\boldsymbol{\theta}_\mathrm{c}) &= 2\,\log\big(Z_\mathrm{s}(\boldsymbol{\theta}_\mathrm{o}-\boldsymbol{\theta}_\mathrm{c})\big) - \log\Big(\det(\boldsymbol{\Lambda}_\mathrm{c}+\boldsymbol{F})\Big) \\ &\quad + (\boldsymbol{\lambda}_\mathrm{c}+\boldsymbol{f}_{\boldsymbol{y}})^\mathsf{T}(\boldsymbol{\Lambda}_\mathrm{c}+\boldsymbol{F})^{-1}(\boldsymbol{\lambda}_\mathrm{c}+\boldsymbol{f}_{\boldsymbol{y}})\,. \end{aligned} \tag{6.33}$$

Since gradients (w.r.t. $\boldsymbol{\theta}_\mathrm{s}$ and $\boldsymbol{\theta}_\mathrm{c}$, respectively) can easily be calculated, *first-order minimization algorithms* are preferable over *zeroth-order algorithms* (gradient-free optimization). However, this optimization step has significant numerical complexity.

For the discrete compressed sensing setup with prior pdf (6.2), after some manipulations, the signal-pdf-dependent term on the right-hand side of (6.32) and (6.33) specializes to

$$\log\big(Z_\mathrm{s}(\boldsymbol{\theta}_\mathrm{s})\big) = \sum_{i=1}^{N} \log\Big(p_0 + 2p_1 \mathrm{e}^{-\Lambda_{\mathrm{s},i}/2}\cosh(\lambda_{\mathrm{s},i})\Big) \tag{6.34}$$

$$\text{or} \quad \log\big(Z_\mathrm{s}(\boldsymbol{\theta}_\mathrm{o}-\boldsymbol{\theta}_\mathrm{c})\big) = \sum_{i=1}^{N} \log\Big(p_0 + 2p_1 \mathrm{e}^{-(\Lambda_{\mathrm{o},i}-\Lambda_{\mathrm{c},i})/2}\cosh(\lambda_{\mathrm{o},i}-\lambda_{\mathrm{c},i})\Big)\,. \tag{6.35}$$

Having $\boldsymbol{\theta}_\mathrm{s}$, in the second step

$$\boldsymbol{\mu}_\mathrm{s} \;=\; \mathrm{E}_{\boldsymbol{x}}\{\boldsymbol{g}(\boldsymbol{x})\} = \int \boldsymbol{g}(\boldsymbol{x})\,\tfrac{1}{Z_\mathrm{s}(\boldsymbol{\theta}_\mathrm{s})}\mathrm{e}^{\boldsymbol{\theta}_\mathrm{s}^\mathsf{T}\boldsymbol{g}(\boldsymbol{x})}\,f_\mathrm{s}(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} \tag{6.36}$$

has to be calculated and (6.30) is evaluated to obtain the overlap (equivalent calculations are carried out when having $\boldsymbol{\theta}_\mathrm{c}$). As $f_\mathrm{s}(\boldsymbol{x})$ factorizes, only one-dimensional integrals have to be solved. Hence, this step has only minor numerical complexity. These two steps are then iterated.

In [46], this strategy is called *"double-loop" algorithm*. We prefer the denominations $\mathrm{ECopt}_\mathrm{s}$ (when using $L_\mathrm{s}(\boldsymbol{\theta}_\mathrm{s})$) and $\mathrm{ECopt}_\mathrm{c}$ (when using $L_\mathrm{c}(\boldsymbol{\theta}_\mathrm{c})$), respectively.

The costly numerical minimization can approximately be done coordinate-wise, i.e., by adjusting only the pair $\lambda_{\mathrm{s},i}$, $\Lambda_{\mathrm{s},i}$ belonging to the variable $x_i$ and going over the variables $i = 1, \ldots, N$ (maybe in some optimized ordering). The $2N$-dimensional optimization problem is broken down to $N$ two-dimensional ones. This procedure is only possible if individual variances are treated (vector-valued diagonalization). The cost function $-2\,L_\mathrm{s}(\boldsymbol{\theta}_\mathrm{s})$ in (6.32) can be reduced after some manipulations to a function for the variable $x_i$ only (for details see [46]), and reads

$$\begin{aligned} -2L_{\mathrm{s},i}(\lambda_{\mathrm{s},i}, \Lambda_{\mathrm{s},i}) \overset{\mathrm{def}}{=} \mathrm{const} - 2\log\Big(p_0 + 2p_1\mathrm{e}^{-\Lambda_{\mathrm{s},i}/2}\cosh(\lambda_{\mathrm{s},i})\Big) \\ + \log\big(\Lambda_{\mathrm{o,s},i} - (\Lambda_{\mathrm{s},i} - \Lambda^\circ_{\mathrm{s},i})\big) - \frac{\big(\lambda_{\mathrm{o,s},i} - (\lambda_{\mathrm{s},i} - \lambda^\circ_{\mathrm{s},i})\big)^2}{\big(\Lambda_{\mathrm{o,s},i} - (\Lambda_{\mathrm{s},i} - \Lambda^\circ_{\mathrm{s},i})\big)}\,, \end{aligned} \tag{6.37}$$

where $\lambda^\circ_{\mathrm{s},i}$ and $\Lambda^\circ_{\mathrm{s},i}$ have to be understood as the current (obsolete, non-optimized) values and $\lambda_{\mathrm{o,c},i}$ and $\Lambda_{\mathrm{o,c},i}$ correspond to $\boldsymbol{\mu}_\mathrm{c}$. Alternatively, the function $-2\,L_\mathrm{c}(\boldsymbol{\theta}_\mathrm{c})$ in (6.33) reduces to

$$\begin{aligned} -2L_{\mathrm{c},i}(\lambda_{\mathrm{c},i}, \Lambda_{\mathrm{c},i}) \overset{\mathrm{def}}{=} \mathrm{const} - 2\log\Big(p_0 + 2p_1\mathrm{e}^{-(\Lambda_{\mathrm{o,c},i} - \Lambda_{\mathrm{c},i})/2}\cosh(\lambda_{\mathrm{o,c},i} - \lambda_{\mathrm{c},i})\Big) \\ + \log\big(\Lambda_{\mathrm{o,c},i} - (\Lambda^\circ_{\mathrm{c},i} - \Lambda_{\mathrm{c},i})\big) - \frac{\big(\lambda_{\mathrm{o,c},i} - (\lambda^\circ_{\mathrm{c},i} - \lambda_{\mathrm{c},i})\big)^2}{\big(\Lambda_{\mathrm{o,c},i} - (\Lambda^\circ_{\mathrm{c},i} - \Lambda_{\mathrm{c},i})\big)}\,, \end{aligned} \tag{6.38}$$

where $\lambda^\circ_{\mathrm{c},i}$ and $\Lambda^\circ_{\mathrm{c},i}$ are the current values. In (6.37) and (6.38) only the first term on the right-hand side is specific for the discrete case; in the general case the first term in (6.37) would read $\log\big(\int \mathrm{f}_x(x_i)\,\mathrm{e}^{\lambda_{\mathrm{s},i}x_i - \Lambda_{\mathrm{s},i}x_i^2/2}\,\mathrm{d}x_i\big)$.

We denote these strategies by $\mathrm{ECseq}_\mathrm{s}$ and $\mathrm{ECseq}_\mathrm{c}$, respectively.

#### 6.3.2.2 Vector Approximate Message Passing: VAMP

As we will see later on in the numerical examples, the optimization procedure leads to very good performance, however, at the cost of numerical complexity. A much simpler strategy can be derived from the fact that the moments $\boldsymbol{\mu}_\mathrm{s}$ and $\boldsymbol{\mu}_\mathrm{c}$ have to match. In [46] this procedure is called *"single-loop" algorithm*.

Here, in the first step

$$\boldsymbol{\mu}_\mathrm{c} \;=\; \mathrm{E}_{\boldsymbol{x}}\{\boldsymbol{g}(\boldsymbol{x})\} = \int \boldsymbol{g}(\boldsymbol{x})\,\tfrac{1}{Z_\mathrm{c}(\boldsymbol{\theta}_\mathrm{c})}\,f_\mathrm{c}(\boldsymbol{x})\,\mathrm{e}^{\boldsymbol{\theta}_\mathrm{c}^\mathrm{T}\boldsymbol{g}(\boldsymbol{x})}\,\mathrm{d}\boldsymbol{x} \tag{6.39}$$

is calculated; then, via (6.30), the overlap $\boldsymbol{\theta}_{\mathrm{o,c}}$ is obtained. Since the distribution parameters are coupled, one can calculate

$$\boldsymbol{\theta}_{\mathrm{s}} = \boldsymbol{\theta}_{\mathrm{o,c}} - \boldsymbol{\theta}_{\mathrm{c}} \,. \tag{6.40}$$

This value is then used in the second step, which is identical to (6.36) in the above approach. Again, having $\boldsymbol{\mu}_{\mathrm{s}}$ the overlap parameter $\boldsymbol{\theta}_{\mathrm{o,s}}$ is calculated using (6.30). Then,

$$\boldsymbol{\theta}_{\mathrm{c}} = \boldsymbol{\theta}_{\mathrm{o,s}} - \boldsymbol{\theta}_{\mathrm{s}} \tag{6.41}$$

is updated and the two steps are iterated. This algorithms coincides with VAMP proposed in [54].

Noteworthy, in the calculation of $\boldsymbol{\mu}_{\mathrm{c}}$, the overlap takes the role of the *prior pdf* $f_{\mathrm{s}}(\boldsymbol{x})$ and in the calculation of $\boldsymbol{\mu}_{\mathrm{s}}$ it takes the role of the *channel pdf* $f_{\mathrm{c}}(\boldsymbol{x})$. These approximations make the calculation of the means computable at all. For the compressed sensing setup,

- the calculation of (6.39) amounts to a *joint linear MMSE (LMMSE)* estimator treating the action of the channel but ignoring the prior pdf—we abbreviate this operation by "LE" (for linear estimator)—, whereas
- the calculation of (6.36) amounts to *individual non-linear MMSE (NLMMSE)* estimators obeying the signal pdf but ignoring the coupling via the sensing matrix—we abbreviate this operation by "NLE" (for non-linear estimator).

These two steps are dual w.r.t. even more aspects; for more details see [59]. For the LMMSE step (6.39) and for a large class of prior pdfs $\mathrm{f}_x(x)$ in the NLMMSE step (6.36) analytic expressions can be given, cf. [4, 54, 59].

These steps can also be interpreted as a projection of the pdfs onto exponential families. The pdf $\mathrm{o}(\boldsymbol{x}) = \frac{1}{Z_{\mathrm{o}}(\boldsymbol{\theta}_{\mathrm{o}})} \mathrm{e}^{\boldsymbol{\theta}_{\mathrm{o}}^{\mathsf{T}} \boldsymbol{g}(\boldsymbol{x})}$ shall approximate the pdf $\mathrm{f}_{\boldsymbol{x}|\boldsymbol{y}}(\boldsymbol{x})$. In the first step, the pdf

$$\mathrm{o}(\boldsymbol{x}) \, \frac{\frac{1}{Z_{\mathrm{c}}(\boldsymbol{\theta}_{\mathrm{c}})} f_{\mathrm{c}}(\boldsymbol{x})}{\mathrm{e}^{\boldsymbol{\theta}_{\mathrm{s}}^{\mathsf{T}} \boldsymbol{g}(\boldsymbol{x})}} = \frac{1}{Z_{\mathrm{c}}(\boldsymbol{\theta}_{\mathrm{c}})} f_{\mathrm{c}}(\boldsymbol{x}) \, \mathrm{e}^{\boldsymbol{\theta}_{\mathrm{c}}^{\mathsf{T}} \boldsymbol{g}(\boldsymbol{x})} \tag{6.42}$$

is projected onto $\mathrm{o}(\boldsymbol{x})$. Using (6.40), $\boldsymbol{\theta}_{\mathrm{s}}$ is calculated and, in the second step, the pdf

$$\mathrm{o}(\boldsymbol{x}) \, \frac{\frac{1}{Z_{\mathrm{s}}(\boldsymbol{\theta}_{\mathrm{s}})} f_{\mathrm{s}}(\boldsymbol{x})}{\mathrm{e}^{\boldsymbol{\theta}_{\mathrm{c}}^{\mathsf{T}} \boldsymbol{g}(\boldsymbol{x})}} = \frac{1}{Z_{\mathrm{s}}(\boldsymbol{\theta}_{\mathrm{s}})} f_{\mathrm{s}}(\boldsymbol{x}) \, \mathrm{e}^{\boldsymbol{\theta}_{\mathrm{s}}^{\mathsf{T}} \boldsymbol{g}(\boldsymbol{x})} \tag{6.43}$$

is projected onto $\mathrm{o}(\boldsymbol{x})$. Then, $\boldsymbol{\theta}_{\mathrm{c}}$ is calculated via (6.41). Since the moments are matched, the projection is done in such a way that the Kullback–Leibler divergence between the involved pdfs is minimized. The projections are iterated until convergence is reached. This is the main approach of expectation propagation [41].

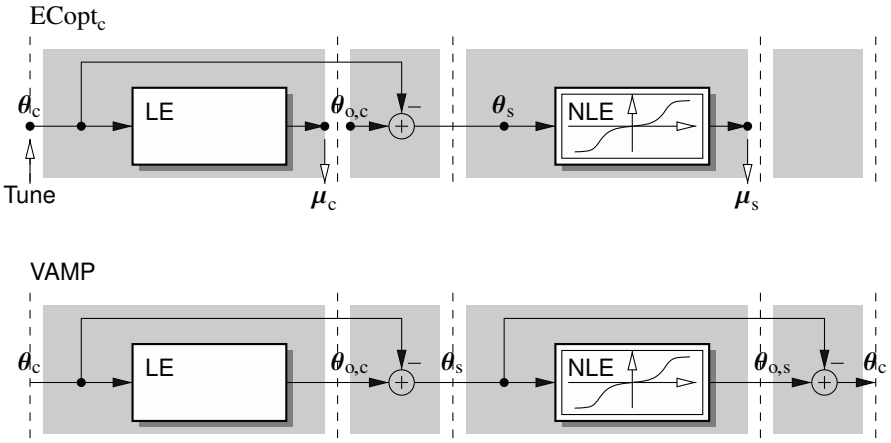



**Fig. 6.2** Partitioning of the function blocks in iterative algorithms derived from the EC framework. Top: Optimization approach (double-loop algorithm); Bottom: VAMP

#### 6.3.2.3 Discussion

Figure 6.2 shows the conceptual splitting of the function blocks of EC-based iterative algorithms for CS. In the top row the calculation steps of one iteration of the optimization approach ECopt$_c$ are visualized. Given $\boldsymbol{\theta}_{o,c}$, this algorithm optimizes (tunes) $\boldsymbol{\theta}_c$ such that $\boldsymbol{\mu}_c = \boldsymbol{\mu}_s$ (via the minimization of (6.33) or in the sequential way employing (6.38)). If a new parameter vector $\boldsymbol{\theta}_c$ is obtained, the corresponding $\boldsymbol{\mu}_c$ determines the new $\boldsymbol{\theta}_{o,c}$. The alternative approach ECopt$_s$ is obvious and not shown.

In the second row, the calculation steps of one iteration of VAMP are shown. In sequence, $\boldsymbol{\mu}_c$ and $\boldsymbol{\mu}_s$ are calculated; both blocks are separated by the updates (6.40) and (6.41).

### 6.3.3 *Alternative Partitioning of the Problem*

Up to now, we have considered the obvious partitioning of the conditional pdf and, thus, of the problem into a "signal" and a "channel" part. In [57], a different, very flexible factorization has been proposed. Here, (6.19) is written as

$$\mathrm{f}_{\boldsymbol{x}|\boldsymbol{y}}(\boldsymbol{x}) = \frac{1}{Z} \underbrace{\prod_{i=1}^{N} \mathrm{f}_x(x_i) \exp\left((1-\gamma) f_{\boldsymbol{y},i} x_i\right)}_{f_\mathrm{s}(\boldsymbol{x})} \underbrace{\exp\left(\boldsymbol{x}^\mathsf{T} \boldsymbol{F} \boldsymbol{x} + \gamma \boldsymbol{f}_{\boldsymbol{y}}^\mathsf{T} \boldsymbol{x}\right)}_{f_\mathrm{c}(\boldsymbol{x})}, \quad (6.44)$$

where $\boldsymbol{f}_{\boldsymbol{y}}^\mathsf{T} = [f_{\boldsymbol{y},1}, \ldots, f_{\boldsymbol{y},N}]$ and $\gamma \in [0,\ 1]$ is a trade-off parameter. For $\gamma = 1$, the above separation (used in VAMP) is obtained. Noteworthy, for $\gamma = 0$, the influence of the observations $\boldsymbol{y}$ is completely taken into account in the signal part.

The surprising, analytic result of [57] is that using an appropriate initialization (depending on $\gamma$), the performance of the algorithm is independent of the choice of $\gamma$. This enables some degree of freedom in the implementations of the estimators (6.36) and (6.39). The details can be found in [57].

## 6.4 Unbiasing of MMSE Estimators

In iterative schemes, i.e., feedback loops, the passing of the results of one processing/decoding block to the next is a crucial point. In principle, suited processing of the results has to ensure that positive feedback is avoided and thus amplification and instable behavior are circumvented. This basic principle emerges in different settings and is known under various denominations. In connection with (V)AMP it is termed *Onsager correction* [54] or *decoupling* [7]. In iterative (turbo) algorithms (e.g., for channel decoding) this is called the calculation of *extrinsic information* [35], which means that only the information gained in the respective step has to be passed on. Here, we will pursue the signal processing/estimation view that a systematic offset in an estimate, a *bias*, has to be removed, i.e., *unbiasing* has to be performed [64]. In [38], bias-free estimators are called *divergence-free*.

In this section, we derive our view of unbiasing; parts have been published in [24, 64]. In view of the functions blocks in EC-based iterative algorithms (see Sect. 6.3), we will address two settings: *joint linear* estimators (the "LE" block in Fig. 6.2) and *scalar non-linear* estimators (the "NLE" block in Fig. 6.2), respectively. First the basic principles and conditions of unbiasing are studied separately, then, the unbiasing procedures are applied to a VAMP-type recovery scheme.

### *6.4.1 Joint Linear Estimators*

We first consider the joint linear estimation, i.e., the block treating the "channel," part. To that end we follow the observation model

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{n}\,, \tag{6.45}$$

where the measurement (channel) matrix $\boldsymbol{A}$ is known. Typically, the *MMSE* estimate is desired—in general it is given by [33, 48] (the index "B" indicates that the estimate is biased, see below)

$$\hat{\boldsymbol{x}}_{\mathrm{c,B}} = \mathrm{E}_{\boldsymbol{x}}\{\boldsymbol{x} \mid \boldsymbol{y}\}\,. \tag{6.46}$$

In case the random vectors are (assumed to be) jointly Gaussian, this *conditional mean estimator* reduces to a linear (affine) one. If $\boldsymbol{x}$ is i.i.d. Gaussian with mean $\boldsymbol{x}_{\mathrm{c}}$ and variance (per component) $\sigma_{\mathrm{c}}^2$ and the noise $\boldsymbol{n}$ is zero-mean i.i.d. Gaussian with

variance (per component) $\sigma_n^2$, independent of $\boldsymbol{x}$, the estimate specializes to[3] [33]

$$\hat{\boldsymbol{x}}_{\mathrm{c,B}} = \boldsymbol{x}_{\mathrm{c}} + \Big(\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A} + \frac{\sigma_n^2}{\sigma_{\mathrm{c}}^2}\boldsymbol{I}\Big)^{-1}\boldsymbol{A}^{\mathsf{T}}\big(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}_{\mathrm{c}}\big) \tag{6.47}$$

$$= \Big(\frac{1}{\sigma_n^2}\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A} + \frac{1}{\sigma_{\mathrm{c}}^2}\boldsymbol{I}\Big)^{-1}\Big(\frac{1}{\sigma_n^2}\boldsymbol{A}^{\mathsf{T}}\boldsymbol{y} + \frac{1}{\sigma_{\mathrm{c}}^2}\boldsymbol{x}_{\mathrm{c}}\Big)\,. \tag{6.48}$$

Per construction—as the *orthogonality principle* is obeyed—the estimation error $\boldsymbol{e}_{\mathrm{c,B}} \overset{\mathrm{def}}{=} \hat{\boldsymbol{x}}_{\mathrm{c,B}} - \boldsymbol{x}$ is orthogonal to the observation $\boldsymbol{y}$ and the error covariance matrix reads

$$\boldsymbol{\Phi}_{\mathrm{c,B}} = \sigma_n^2\Big(\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A} + \frac{\sigma_n^2}{\sigma_{\mathrm{c}}^2}\boldsymbol{I}\Big)^{-1} = \sigma_{\mathrm{c}}^2\big(\boldsymbol{I} - \boldsymbol{K}\big)\,, \tag{6.49}$$

where we have used the *end-to-end cascade* (channel + estimator)

$$\boldsymbol{K} = [K_{ij}] \overset{\mathrm{def}}{=} \Big(\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A} + \frac{\sigma_n^2}{\sigma_{\mathrm{c}}^2}\boldsymbol{I}\Big)^{-1}\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A}\,. \tag{6.50}$$

The average error variance is given by

$$\sigma_{\mathrm{c,B}}^2 = \frac{1}{N}\operatorname{tr}(\boldsymbol{\Phi}_{\mathrm{B,c}}) = \sigma_{\mathrm{c}}^2\Big(1 - \frac{1}{N}\operatorname{tr}(\boldsymbol{K})\Big)\,. \tag{6.51}$$

Like all MMSE estimates, $\hat{\boldsymbol{x}}_{\mathrm{c,B}}$ is *biased* (hence, the index "B") in the sense that—as per basic principle the error is orthogonal to the observation [48]—part of the useful signal is accounted to the error [27]. Unbiasing leads to an error that is orthogonal to the desired quantity; it may be done by scaling the second part of the estimate $\hat{\boldsymbol{x}}_{\mathrm{c,B}}$ in (6.47) suitably, i.e.,

$$\hat{\boldsymbol{x}}_{\mathrm{s}} = \boldsymbol{x}_{\mathrm{c}} + \boldsymbol{C}\Big(\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A} + \frac{\sigma_n^2}{\sigma_{\mathrm{c}}^2}\boldsymbol{I}\Big)^{-1}\boldsymbol{A}^{\mathsf{T}}\big(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}_{\mathrm{c}}\big)\,. \tag{6.52}$$

For the joint linear estimator, we have two main principles how to adjust the scaling matrix $\boldsymbol{C}$, *average* and *individual* unbiasing.

#### 6.4.1.1 Average Unbiasing

For an *average unbiasing* we restrict ourselves to $\boldsymbol{C} = c_{\mathrm{c}}\,\boldsymbol{I}$ and demand

[3] We continue the notation of Fig. 6.2. The biased estimate (corresponding to the "overlap") is denoted by the respective estimation step. The unbiased estimate—which is the input to the other estimator—is not denoted by the block where it is produced, but by the block where it is input.

$$\mathrm{tr}(\boldsymbol{C}\,\boldsymbol{K}) \;=\; \mathrm{tr}(c_\mathrm{c}\,\boldsymbol{K}) \stackrel{!}{=} N\;, \tag{6.53}$$

which is achieved by

$$c_\mathrm{c} = \frac{N}{\mathrm{tr}(\boldsymbol{K})} \;=\; \frac{\sigma_\mathrm{c}^2}{\sigma_\mathrm{c}^2 - \sigma_\mathrm{B,c}^2}\;, \tag{6.54}$$

where the last form follows from (6.51).

It can be shown that data and unbiased error $\boldsymbol{e}_\mathrm{s} \stackrel{\mathrm{def}}{=} \boldsymbol{x}_\mathrm{s} - \boldsymbol{x}$ are *orthogonal on average*, i.e.,

$$\frac{1}{N}\mathrm{E}\{\boldsymbol{x}^\mathsf{T}\boldsymbol{e}_\mathrm{s}\} = \frac{1}{N}\sum\nolimits_{i=1}^{N}\mathrm{E}\{x_i\,e_{\mathrm{s},i}\} \;=\; 0\;, \tag{6.55}$$

and, after some manipulations, that the average variance of $\boldsymbol{e}_\mathrm{s}$ is given by

$$\sigma_\mathrm{s}^2 = \frac{1}{N}\sum\nolimits_{i=1}^{N}\mathrm{E}\{e_{\mathrm{s},i}^2\} \;=\; \Big(\frac{1}{\sigma_\mathrm{B,c}^2} - \frac{1}{\sigma_\mathrm{c}^2}\Big)^{-1} \;=\; \sigma_\mathrm{c}^2\Big(\frac{1}{\mathrm{M_A}(K_{i,i})} - 1\Big)\;, \tag{6.56}$$

where $\mathrm{M_A}(\cdot)$ denotes the *arithmetic mean*. The unbiased estimate (6.52) can be written as (cf. also [28])

$$\hat{\boldsymbol{x}}_\mathrm{s} = c_\mathrm{c}\hat{\boldsymbol{x}}_\mathrm{c,B} - (c_\mathrm{c}-1)\boldsymbol{x}_\mathrm{c} \;=\; \sigma_\mathrm{s}^2\Big(\frac{\hat{\boldsymbol{x}}_\mathrm{c,B}}{\sigma_\mathrm{c,B}^2} - \frac{\boldsymbol{x}_\mathrm{c}}{\sigma_\mathrm{c}^2}\Big)\;. \tag{6.57}$$

#### 6.4.1.2 Individual Unbiasing

Alternatively, the components may be scaled individually such that the components of the unbiased error are *individually orthogonal* to the data, i.e., $\mathrm{E}\{x_i\,e_{\mathrm{s},i}\} = 0,\ \forall i$. This is achieved when choosing

$$\boldsymbol{C} = \mathbf{diag}(1/K_{11},\ldots,1/K_{NN})\;. \tag{6.58}$$

It can be shown that the individual variances and the average variance of $\boldsymbol{e}_\mathrm{s}$ amount to [22]

$$\sigma_{\mathrm{s},i}^2 = \mathrm{E}\{e_{\mathrm{s},i}^2\} \;=\; \sigma_\mathrm{c}^2\frac{1-K_{i,i}}{K_{i,i}} \tag{6.59}$$

$$\sigma_\mathrm{s}^2 = \frac{1}{N}\sum\nolimits_{i=1}^{N}\mathrm{E}\{e_{\mathrm{s},i}^2\} = \frac{1}{N}\sum\nolimits_{i=1}^{N}\sigma_\mathrm{c}^2\frac{1-K_{i,i}}{K_{i,i}} = \sigma_\mathrm{c}^2\Big(\frac{1}{\mathrm{M_H}(K_{i,i})}-1\Big)\;, \tag{6.60}$$

where $\mathrm{M_H}(\cdot)$ denotes the *harmonic mean*.

Noteworthy, the two unbiasing strategies (average unbiasing vs. individual unbiasing) are not identical. Since the elements $K_{i,i}$ of the end-to-end cascade are all positive and real, the relation $\mathrm{M_A}(K_{i,i}) > \mathrm{M_H}(K_{i,i})$ holds. This means that the average estimation variance is smaller if only orthogonality on average is demanded; the more strict demand of individual orthogonality of the error components and data leads to a (somewhat) larger average variance. However, the individual variances can be utilized profitably in the next processing step of an iterative algorithm leading finally to a gain.

### 6.4.2 Scalar Non-linear Estimators

We now consider the individual non-linear estimation, i.e., the block treating the "signal," part. In this scalar case, the channel model is given by

$$y = x + w\,, \tag{6.61}$$

where $x$ is drawn according to some known prior pdf $\mathrm{f}_x(x)$ (with mean zero and variance $\sigma_x^2$), and the noise (disturbance) $w$ is zero-mean Gaussian with variance $\sigma_w^2$ and independent of the data $x$; hence, $x$ and $w$ are orthogonal, i.e., $\mathrm{E}\{xw\} = 0$.

Again, we are interested in an estimate $\hat{x}$ which is calculated such that the mean-squared error is minimized. The corresponding *conditional mean estimator*

$$\hat{x}_{\mathrm{s,B}} = \mathrm{E}_x\{x \mid y\} \overset{\mathrm{def}}{=} S(y) \tag{6.62}$$

is the optimum solution [33, 48]. Sometimes we may explicitly indicate the dependency of the estimate on the observation, i.e., write $\hat{x}_{\mathrm{s,B}}(y)$. The *conditional variance* of the estimation error $e_{\mathrm{s,B}}(y) \overset{\mathrm{def}}{=} x - \hat{x}_{\mathrm{s,B}}(y)$, and the *mean-squared error* calculate to

$$\varsigma^2_{\mathrm{s,B}}(y) = \mathrm{E}_x\{e^2_{\mathrm{s,B}}(y) \mid y\} = \sigma_w^2 \frac{\mathrm{d}}{\mathrm{d}y} S(y)\,, \tag{6.63}$$

$$\sigma^2_{\mathrm{s,B}} = \mathrm{E}_y\{\varsigma^2_{\mathrm{s,B}}(y)\}\,. \tag{6.64}$$

The right expression in (6.63) holds since the noise is Gaussian and, thus, a member of an exponential family [42].

Whenever $\boldsymbol{x}$ is not Gaussian, the characteristic curve of the estimator is not a linear function, $S(y) \neq a \cdot y$, and the conditional variance is dependent on the observation $y$. Only for linear estimators, conditional variance and mean-squared error coincide.

The considered setting is depicted as a block diagram in Fig. 6.3. The observation $y$ (given by model (6.61)) is fed to the estimation function $S(y)$ which provided the biased estimate $\hat{x}_{\mathrm{s,B}}$. The processing for the subsequently discussed unbiasing strategies (6.67) and (6.71) is also shown.

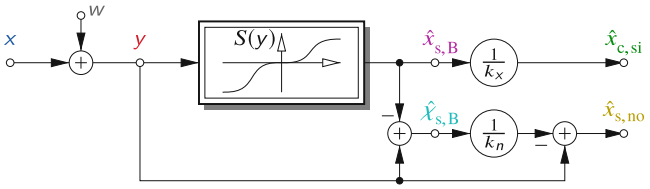



**Fig. 6.3** Situation of scalar non-linear estimation and unbiasing (reproduced from [24])

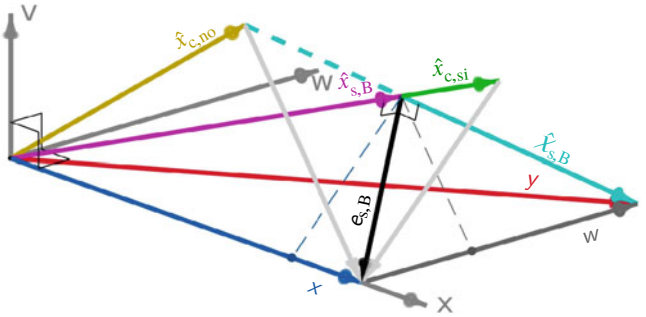



**Fig. 6.4** Visualization of the relations of the random variables in three-dimensional vector space; orthogonal dimensions $x$, $w$, and $v$ (reproduced from [24])

The relation between random variables can conveniently be visualized through vectors in a vector space [48]. Thereby, the lengths of the vectors correspond to the standard deviations of the random variables and the angles to the covariances—uncorrelated random variables correspond to perpendicular vectors. A visualization of the present situation is given in Fig. 6.4 in a three-dimensional space. Baseline is the horizontal $x$-$w$-plane; since $x$ (blue) and $w$ (gray) are uncorrelated we have $\sigma_y^2 = \sigma_x^2 + \sigma_w^2$. A third dimension $v$ is required to represent the action of the non-linear device $S(y)$. Per basic estimation principle, the error $e_{s,B}$ (black) is orthogonal to the observation $y$ and to the estimate $\hat{x}_{s,B}(y)$. As the error is not orthogonal to the data $x$, a bias is present.

For the scalar non-linear estimator, since we operate in a three-dimensional vector space, two main principles how to do the unbiasing are possible, *signal-oriented* and *noise-oriented* unbiasing.

#### 6.4.2.1 Signal-Oriented Unbiasing

The general principle to derive the bias compensation is the decomposition of the estimate (the output of the non-linear estimator device $S(y)$) into a scaled (scaling factor $k_x$) version of $x$ plus an uncorrelated distortion $w_x$, i.e.,

$$\hat{x}_{s,B}(y) = S(y) \stackrel{!}{=} k_x x + w_x \,, \tag{6.65}$$

where the gain $k_x$ is adjusted such that $x$ and $w_x$ are orthogonal, i.e., $\mathrm{E}\{x\, w_x\} = 0$. This is obtained by

$$k_x = \frac{\mathrm{E}_x\{S(y)\, x\}}{\mathrm{E}_x\{x^2\}} = \frac{\sigma_x^2 - \sigma_{\mathrm{s,B}}^2}{\sigma_x^2} = \frac{\sigma_x^2}{\sigma_x^2 + \sigma_w^2} \,. \tag{6.66}$$

Noteworthy, the same result is achieved from geometric considerations: the scaling factor $k_x$ corresponds to the projection of $\hat{x}_{\mathrm{s,B}}$ onto $x$ (blue dashed line and blue dot in Fig. 6.4). Using basic geometry of right triangles, (6.66) is obtained.

When performing "signal-oriented" unbiasing (SoU) (upper branch in Fig. 6.3), the estimate (green) is then given by [64]

$$\hat{x}_{\mathrm{c,si}} = \frac{1}{k_x}\, \hat{x}_{\mathrm{s,B}} = \frac{\sigma_x^2}{\sigma_x^2 - \sigma_{\mathrm{s,B}}^2}\, \hat{x}_{\mathrm{s,B}} \,. \tag{6.67}$$

Here, the error $e_{\mathrm{c,si}} \overset{\mathrm{def}}{=} x - \hat{x}_{\mathrm{c,si}}$ (light gray) is orthogonal to the data $x$; it lies parallel to the $w$-$v$-plane. It can be shown that conditional variance and MSE calculate to

$$\varsigma_{\mathrm{c,si}}^2(y) = \mathrm{E}_x\{e_{\mathrm{c,si}}(y) \mid y\} = \varsigma_{\mathrm{s,B}}^2(y) + \Big(\frac{\sigma_{\mathrm{s,B}}^2}{\sigma_x^2 - \sigma_{\mathrm{s,B}}^2}\Big)^2 \hat{x}_{\mathrm{s,B}}^2(y) \,, \tag{6.68}$$

$$\sigma_{\mathrm{s,si}}^2 = \mathrm{E}_y\{\varsigma_{\mathrm{c,si}}^2(y)\} = \Big(\frac{1}{\sigma_{\mathrm{s,B}}^2} - \frac{1}{\sigma_x^2}\Big)^{-1} \,. \tag{6.69}$$

#### 6.4.2.2 Noise-Oriented Unbiasing

In case of non-linear estimators, alternatively, a "noise-oriented" unbiasing (NoU) can be performed [64]. Here, the noise estimate $\hat{\chi}_{\mathrm{s,B}} = \mathrm{E}_w\{w \mid y\} = \mathrm{E}_x\{y - x \mid y\} = y - \hat{x}_{\mathrm{s,B}}$ (cyan in Fig. 6.4) is considered. By basic geometry, the scaling factor $k_w$ corresponding to the projection of $\hat{\chi}_{\mathrm{s,B}}$ onto $w$ (gray dashed line and gray dot) is given by

$$k_w = \frac{\sigma_w^2 - \sigma_{\mathrm{s,B}}^2}{\sigma_w^2} = \frac{\sigma_w^2}{\sigma_w^2 + \sigma_x^2} \,. \tag{6.70}$$

We are finally interested in the unbiased estimate $\hat{x}_{\mathrm{c,no}}$ (golden), given by

$$\hat{x}_{\mathrm{c,no}} = y - \frac{1}{k_w}\, \hat{\chi}_{\mathrm{s,B}} = y + \frac{\sigma_w^2}{\sigma_w^2 - \sigma_{\mathrm{s,B}}^2}\, (\hat{x}_{\mathrm{s,B}}(y) - y) \,. \tag{6.71}$$

Here, the error $e_{\mathrm{c,no}}(y) \overset{\mathrm{def}}{=} x - \hat{x}_{\mathrm{c,no}}(y)$ (light gray) is orthogonal to $w$ and lies in the $x$-$v$-plane. Direct calculations reveal that conditional variance and MSE calculate to

$$\varsigma^2_{\mathrm{s,no}}(y) = \mathrm{E}_x\{e_{\mathrm{c,no}}(y) \mid y\} = \varsigma^2_{\mathrm{s,B}}(y) + \left(\frac{\sigma^2_{\mathrm{s,B}}}{\sigma^2_w - \sigma^2_{\mathrm{s,B}}}\right)^2 (\hat{x}_{\mathrm{s,B}}(y) - y)^2 \,, \quad (6.72)$$

$$\sigma^2_{\mathrm{s,no}} = \mathrm{E}_y\{\varsigma^2_{\mathrm{c,no}}(y)\} = \left(\frac{1}{\sigma^2_{\mathrm{s,B}}} - \frac{1}{\sigma^2_w}\right)^{-1} . \quad (6.73)$$

### 6.4.3 Iterative Schemes with Individual and Average Variances

Having derived the unbiasing approaches, we now apply them to iterative VAMP-type algorithms. Thereby, either an average variance or individual variances can be employed to represent the reliability of the estimates. Noteworthy, the biased estimates treated in this section are synonymous to the overlap treated in Sect. 6.2.

For the *joint linear estimator*, unbiasing causes no problems. If an average variance is desired, unbiasing on average (Sect. 6.4.1.1) is performed and the average variance calculates to (6.56). When individual variances are desired, individual unbiasing (Sect. 6.4.1.2) is carried out and the individual variances (6.59) are passed to the next stage in the iterative algorithm (from "LE" to "NLE" in Fig. 6.2).

In terms of normalized mean and precision (natural parameters), the unbiasing operations are simply given by

$$\boldsymbol{\lambda}_{\mathrm{s}} = \boldsymbol{\lambda}_{\mathrm{o,c}} - \boldsymbol{\lambda}_{\mathrm{c}} \,, \quad \begin{array}{ll} \Lambda_{\mathrm{s}} = \Lambda_{\mathrm{o,c}} - \Lambda_{\mathrm{c}} \,, & \mathrm{AvgV} \\ \boldsymbol{\Lambda}_{\mathrm{s}} = \boldsymbol{\Lambda}_{\mathrm{o,c}} - \boldsymbol{\Lambda}_{\mathrm{c}} \,, & \mathrm{IndV} \end{array} \,, \quad (6.74)$$

which is nothing else than the subtraction between the "LE" and the "NLE" block in Fig. 6.2 (cf. the definition of $\boldsymbol{\theta}$ in Sect. 6.2.2).

The situation for the parallel *scalar non-linear estimators* is more involved. When an average variance is desired, first the biased MSE $\sigma^2_{\mathrm{s,B}}$ has to be calculated. Thereby, the statistical expectation in (6.64) is replaced by the empirical average over the parallel branches. Then, the unbiased variances $\sigma^2_{\mathrm{s,si}}$ or $\sigma^2_{\mathrm{s,no}}$ are obtained via (6.69) or (6.73). Finally, the unbiasing of the elements is done using (6.67) or (6.71) where the scaling factors can be written compactly as $\sigma^2_x/(\sigma^2_x - \sigma^2_{\mathrm{s,B}}) = \sigma^2_{\mathrm{s,si}}/\sigma^2_{\mathrm{s,B}}$ and $\sigma^2_w/(\sigma^2_w - \sigma^2_{\mathrm{s,B}}) = \sigma^2_{\mathrm{s,no}}/\sigma^2_{\mathrm{s,B}}$, respectively [64].

When individual variances are passed to and should be produced by the parallel scalar non-linear estimators, an additional problem occurs. The biased MSE $\sigma^2_{\mathrm{s,B}}$ cannot be calculated as an empirical average. As this quantity (given $\mathrm{f}_x(x)$) is a function of the noise variance $\sigma^2_w$ only, it may be precalculated and tabulated or approximated by simple functions (e.g., a polynomial). Then, unbiasing is performed according to (6.67) or (6.71) and the conditional variances $\varsigma^2_{\mathrm{c,si}}(y)$ or $\varsigma^2_{\mathrm{s,no}}(y)$ according to (6.68) or (6.72) are passed individually per branch to the next stage.

When using

$$\begin{aligned} &\lambda_{\mathrm{o,s},i} \overset{\text{def}}{=} \hat{x}_{\mathrm{s,B},i}/\varsigma^2_{\mathrm{s,B},i}\ , \lambda_{\mathrm{b},i} \overset{\text{def}}{=} \hat{x}_{\mathrm{s,B},i}/\sigma^2_{\mathrm{s,B},i}\ , \lambda_{\mathrm{s},i} \overset{\text{def}}{=} y/\sigma^2_w\ , \\ &\Lambda_{\mathrm{o,s},i} \overset{\text{def}}{=} 1/\varsigma^2_{\mathrm{s,B},i}\ , \quad \Lambda_{\mathrm{b},i} \overset{\text{def}}{=} 1/\sigma^2_{\mathrm{s,B},i}\ , \quad \Lambda_{\mathrm{s},i} \overset{\text{def}}{=} 1/\sigma^2_w\ , \quad \Lambda_{\mathrm{x}} \overset{\text{def}}{=} 1/\sigma^2_x\ , \end{aligned} \tag{6.75}$$

the signal-oriented unbiasing operations (per element $i$) read in terms of the natural parameters

$$\begin{aligned} &\lambda_{\mathrm{c},i} = \lambda_{\mathrm{o,s},i}\ , \qquad \Lambda_{\mathrm{c}} = \Lambda_{\mathrm{o,s}} - \Lambda_{\mathrm{x}}\ , && \text{AvgV} \\ &\lambda_{\mathrm{c},i} = \tfrac{\Lambda_{\mathrm{c},i}}{\Lambda_{\mathrm{b},i},-\Lambda_{\mathrm{x}}}\lambda_{\mathrm{b},i}, \quad \Lambda_{\mathrm{c},i} = \left(\frac{1}{\Lambda_{\mathrm{o,s},i}} + \left(\frac{\Lambda_{\mathrm{x}}}{\Lambda_{\mathrm{b},i} - \Lambda_{\mathrm{x}}}\,\frac{\lambda_{\mathrm{b},i}}{\Lambda_{\mathrm{b},i}}\right)^2\right)^{-1}\ , && \text{IndV} \end{aligned} \tag{6.76}$$

and for noise-oriented unbiasing

$$\begin{aligned} &\lambda_{\mathrm{c},i} = \lambda_{\mathrm{o,s},i} - \lambda_{\mathrm{s},i}\ , && \Lambda_{\mathrm{c}} = \Lambda_{\mathrm{o,s}} - \Lambda_{\mathrm{s}}\ , && \text{AvgV} \\ &\lambda_{\mathrm{c},i} = \tfrac{\Lambda_{\mathrm{c},i}}{\Lambda_{\mathrm{b},i}-\Lambda_{\mathrm{s},i}}\left(\lambda_{\mathrm{b},i} - \lambda_{\mathrm{s},i}\right)\ , && \Lambda_{\mathrm{c},i} = \left(\tfrac{1}{\Lambda_{\mathrm{o,s},i}} + \left(\tfrac{\Lambda_{\mathrm{s},i}}{\Lambda_{\mathrm{b},i}-\Lambda_{\mathrm{s},i}}\left(\tfrac{\lambda_{\mathrm{b},i}}{\Lambda_{\mathrm{b},i}} - \tfrac{\lambda_{\mathrm{s},i}}{\Lambda_{\mathrm{s},i}}\right)\right)^2\right)^{-1}\ . && \text{IndV} \end{aligned} \tag{6.77}$$

As can be seen, only the noise-oriented average unbiasing corresponds to the conventional update in the EC approach—the simple subtraction between the "NLE" and the "LE" block in Fig. 6.2. Using this straightforwardly for individual variances is not optimum. Indeed, as shown in [24], $\Lambda_{\mathrm{c},i}$ will become frequently negative leading to unusable results which have to be clipped and, thus, to non-optimum performance of the reconstruction algorithm. The present new derivation from estimation theory, however, guarantees meaningful parameters and improved performance.

## 6.5 Numerical Results and Discussion

The discussed iterative signal recovery approaches are now assessed and compared by means of numerical simulations. We thereby restrict ourselves to discrete compressed sensing; specifically, the signal pdf (6.2) is employed. The elements of the sensing matrix $\boldsymbol{A}$ are assumed to be i.i.d. unit-variance Gaussian. The columns of $\boldsymbol{A}$ are then scaled to unit $\ell_2$ norm, which in communications corresponds to a transmitter-side power control. Two different dimensionalities of the problem are investigated: the choice $N = 258$, $M = 129$ with sparsity $s = 12$ (Scenario A), and $N = 64$, $M = 32$ with sparsity $s = 4$ (Scenario B).

We first consider schemes where the reliability over the iterations is characterized by an average variance. Then, schemes utilizing individual variances are studied.

### 6.5.1 Average Variance

For assessing the schemes, we plot the *symbol error ratio (SER)*, i.e., the ratio of erroneously recovered signal elements $x_i$ and total number of symbols, over the iterations. These plots cover the convergence behavior of the algorithm and the steady-state performance. As usual in communications, the SER is displayed in logarithmic scale because we are interested in the order of magnitude of the residual error ratio.

In Fig. 6.5, the SER is shown for Scenario A and for algorithms utilizing an average variance. AMP [17, 39] is compared with VAMP [54] (which employs noise-oriented average unbiasing according to (6.77)); in addition VAMP using signal-oriented average unbiasing according to (6.76) is shown. For stability reasons, the precision parameters $\Lambda$ are clipped to the interval $\Lambda \in [10^{-8},\ 10^{8}]$. The signal-to-noise ratio is adjusted to $10\log_{10}(1/\sigma_n^2) \mathrel{\widehat{=}} 17$ dB.

It can be seen that VAMP outperforms AMP slightly; essentially a somewhat faster convergence is achieved. For the present setting of sufficiently large dimensions (even still short for a number of applications) the steady-state performance is reached after a few iterations and differs not too much.

It is apparent that noise-oriented unbiasing outperforms the signal-oriented variant. This is explained by the dual operations in the two estimation steps in VAMP (see Fig. 6.2). The one block performs joint linear estimation (concentrating on the channel action), the other block performs individual non-linear estimation (taking only the signal statistics into account). Dual to signal-oriented (average) unbiasing after the linear estimation, noise-oriented (average) unbiasing after the non-linear estimation should be used. For more details see [59].

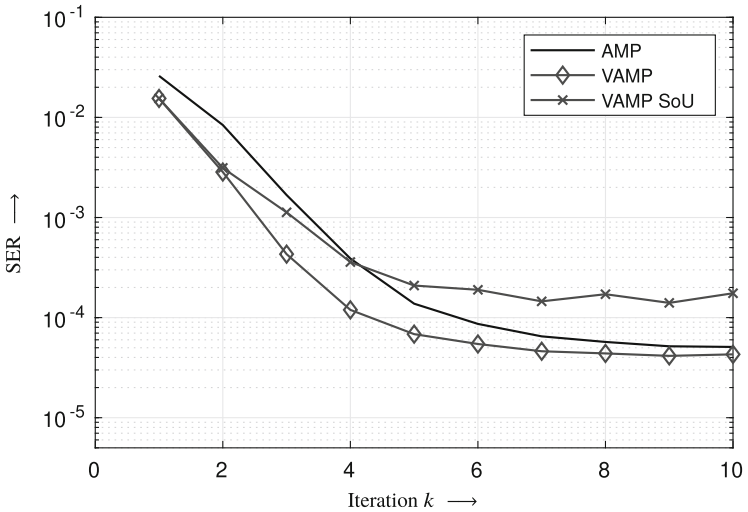



**Fig. 6.5** Symbol error ratio (SER) over the iterations. $10\log_{10}(1/\sigma_n^2) \mathrel{\widehat{=}} 17$ dB, $N = 258$, $M = 129$, $s = 12$. Clipping of $\Lambda$ to $[10^{-8},\ 10^{8}]$

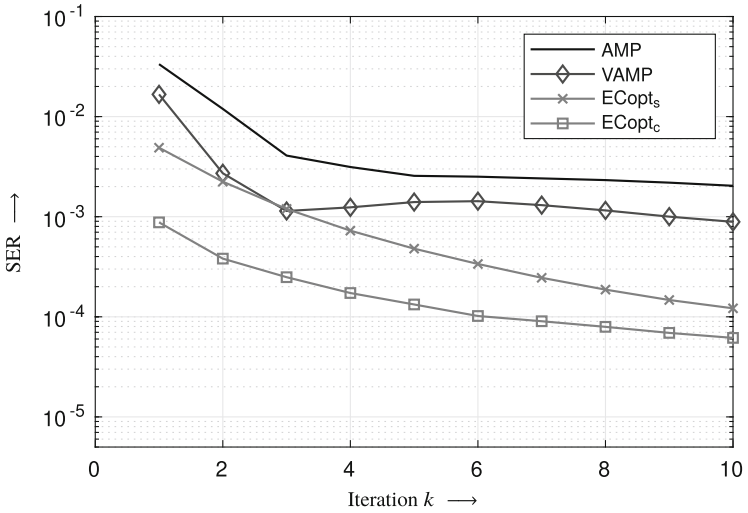



**Fig. 6.6** Symbol error ratio (SER) over the iterations. $10\log_{10}(1/\sigma_n^2) \mathrel{\widehat{=}} 20$ dB, $N = 64$, $M = 32$, $s = 4$. Clipping of $\Lambda$ to $[10^{-8},\ 10^{8}]$

The results for Scenario B are displayed in Fig. 6.6. Again, the SER is shown over the iterations for AMP and VAMP. In addition, the results of ECopt$_s$ and ECopt$_c$ are given. Here, the signal-to-noise ratio is adjusted to 20 dB.

In case of small dimensions, AMP and VAMP converge poorly; when carrying out some hundred iterations, some improvements for both algorithms are possible; in the present case, both converge to error ratios around $2 \cdot 10^{-4}$. ECopt$_s$ and, in particular, ECopt$_c$ offer a much better performance. Note that the all-zero vector is chosen as starting point for ECopt$_s$, whereas in ECopt$_c$ the joint linear estimate is calculated first and used as initialization for $\boldsymbol{\theta}_{o,c}$ (cf. Fig. 6.2). This "warm start" leads to the advantage (horizontal shift of the curve) of ECopt$_c$ over the other variant; except this fact both versions perform similar.

Noteworthy, the improvement in SER comes at the cost of increased complexity. Even though the curves are plotted over the iteration number, significant differences in the complexity per iteration are present. AMP and VAMP perform simple matrix operations and scalar non-linear estimation only, whereas ECopt uses a convex optimization algorithm on a $(N + 1)$-dimensional cost function (6.33) or (6.32) per iteration.

### *6.5.2 Individual Variances*

We now turn to the case of using individual variances within the algorithms to characterize the reliabilities. Thereby, a more fine-grained knowledge on the

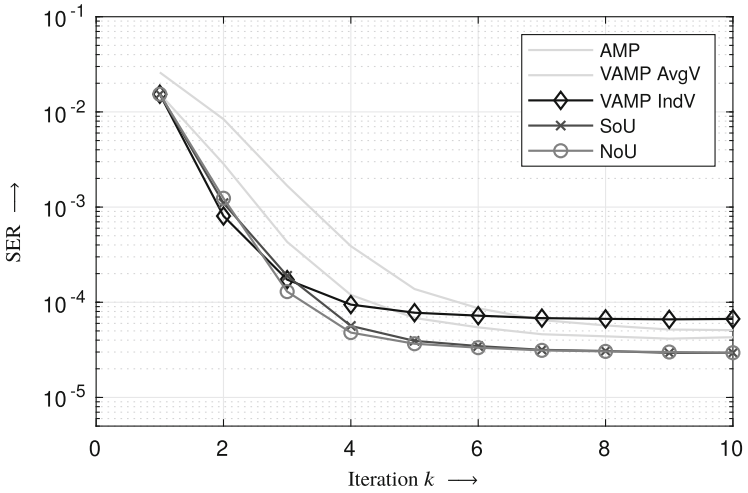



**Fig. 6.7** Symbol error ratio (SER) over the iterations. $10\log_{10}(1/\sigma_n^2) \mathrel{\widehat{=}} 17$ dB, $N = 258$, $M = 129$, $s = 12$. Clipping of $\Lambda$ to $[10^{-8},\ 10^{8}]$

estimation quality of the symbols may be exploited, potentially leading to a better performance. However, it has to be admitted that the numerical complexity increases. For average variances, where $\boldsymbol{\Lambda}$ in (6.14) is a diagonal matrix, the inverse $(\boldsymbol{\Lambda} + \boldsymbol{F})^{-1}$ in (6.32), (6.33) can be efficiently calculated using a singular-value decomposition, cf. [54]. This is not possible for general diagonal matrices.

The results for Scenario A are collected in Fig. 6.7. As above, the SER is shown over the iterations. For reference, the results for AMP and VAMP from above (average variance) are repeated in gray. We compare VAMP with individual variance, where the update after the non-linear estimators is done in the straightforward way (analogously to (6.74), which, in each case, is employed after the linear estimator), with the unbiasing strategies derived in Sect. 6.4.2.

Apparently, straightforwardly applying the EC framework with vector-valued diagonalization to the compressed sensing setup does not lead to satisfactory performance. Even AMP and VAMP with average variance outperform this variant. However, when employing the unbiasing rules derived in Sect. 6.4.2, an improvement over classical VAMP can be achieved. Here, noise-oriented unbiasing performs only slightly better than the signal-oriented variant.

Noteworthy, for the unbiasing rules (6.76) and (6.77), the biased mean-squared error $\sigma_{\mathrm{s,B}}^2$ (which depends on $N/s$, i.e., $p_1$ in (6.2), only), or equivalently the precision $\Lambda_{\mathrm{b}} = 1/\sigma_{\mathrm{s,B}}^2$, is required; it cannot be calculated by averaging within the algorithm. Hence, either $\Lambda_{\mathrm{b}}$ is precalculated (as a function of $\Lambda_{\mathrm{s}}$) and tabulated or an approximation is used. The SER curves when using a fine-grained table cannot be distinguished from that when using the simple approximation $\Lambda_{\mathrm{b}} = \exp(0.1330\Lambda_{\mathrm{s}} + 2.754)$ which holds for the present ratio $N/s$.

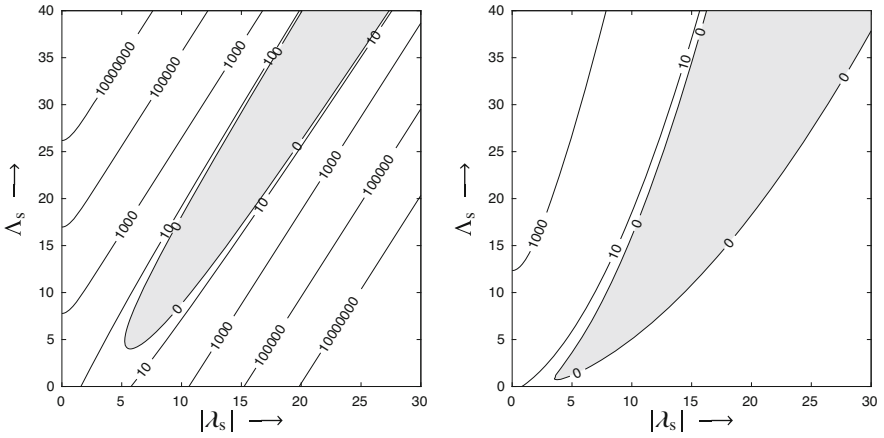



**Fig. 6.8** Contour plot of $\Lambda_c$ over $|\boldsymbol{\lambda}_s|$ and $\Lambda_s$. Left: discrete pdf according to (6.2), $s = 12$; Right: Bernoulli–Gauss pdf; the non-zero elements are zero-mean, unit-variance Gaussian, $s = 12$

The poor performance of the straightforward approach can be explained by studying the precision $\Lambda_c$ (after the non-linear estimator and the update $\Lambda_c = \Lambda_{o,s} - \Lambda_s$) as a function of $|\boldsymbol{\lambda}_s|$ and $\Lambda_s$ (cf. Fig. 6.2). To that end, a contour plot of $\Lambda_c$ is shown in Fig. 6.8. Within the gray-shaded area $\Lambda_c$ becomes negative and, thus, it does not have any sensible meaning. This circumstance can only be handled by clipping. This effect has already been observed in [54]—however, when employing average variances, negative precision parameters occur very rarely and clipping does (almost) not hurt. In case of individual variances negative quantities occur much more often; in [24] we have shown that up to 5 % of the components of $\boldsymbol{x}$ are affected in case of the discrete prior (6.2). In [30] this problem is treated for ECseq by incorporating additional constraints in the optimization. In order to show that this is not an effect of the discrete prior, on the right-hand side the contour plot is shown for a Bernoulli–Gauss pdf; there the effect is even more pronounced.

The respective results for Scenario B are depicted in Fig. 6.9. The curves for AMP, VAMP, and $\text{ECopt}_c$ employing average variances are repeated in gray for reference.

First, we note that in case of small dimensions, the use of individual variances leads to more performance gains than in case of larger dimension. However, the straightforward application in the EC framework is outperformed by the derived unbiasing from Sect. 6.4.2.

Here, $\text{ECopt}_s$ does not perform as well as $\text{ECopt}_c$. This effect is again explained by the incorrect unbiasing. In $\text{ECopt}_s$ the parameter vector $\boldsymbol{\theta}_s$ is tuned and $\boldsymbol{\theta}_c = \boldsymbol{\theta}_{o,s} - \boldsymbol{\theta}_s$ is employed within the cost function; see also Fig. 6.2. However, this is exactly the stage where negative precisions occur very likely. Clipping these negative values deteriorates the performance. In contrast, in $\text{ECopt}_c$ only the unbiasing/update $\boldsymbol{\theta}_s = \boldsymbol{\theta}_{o,c} - \boldsymbol{\theta}_c$ is required, which is the correct procedure. As a consequence, this version outperforms the alternative version. Noteworthy, contrary

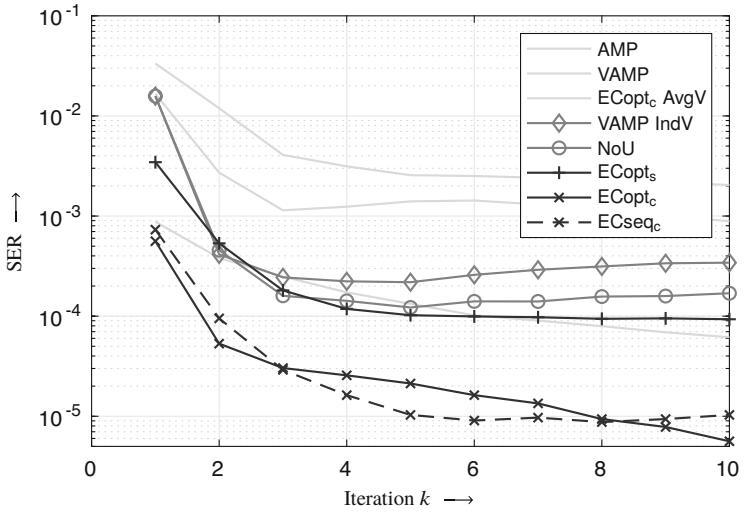



**Fig. 6.9** Symbol error ratio (SER) over the iterations. $10\log_{10}(1/\sigma_n^2) \mathrel{\widehat{=}} 20$ dB, $N = 64$, $M = 32$, $s = 4$. Clipping of $\Lambda$ to $[10^{-8},\ 10^{8}]$

to the statement in [46], the partitioning of the problem is not symmetric and the factors are not equivalent.

Finally, the sequential optimization approach ECseq$_c$ almost performs the same as when performing full optimization. However, only $N$ two-dimensional optimizations per iteration instead of a $2N$-dimensional one have to be carried out leading to a much smaller complexity. The alternative approach ECseq$_s$ does not perform well (not shown) due to the reasons discussed above. For sensing matrices of moderate sizes, ECseq$_c$ is an interesting scheme offering very good performance at manageable numerical complexity.

**Acknowledgments** This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) — Grants Fi 982/8-1 and Fi 982/16-1.

## References

1. Baraniuk, R.G., Cevher, V., Duarte, M.F., Hegde, C.: Model-based compressive sensing. IEEE Trans. Inf. Theory **56**(4), 1982–2001 (2010)
2. Baron, D., Sarvotham, S., Baraniuk, R.G.: Bayesian compressive sensing via belief propagation. IEEE Trans. Signal Process. **58**(1), 269–280 (2010)
3. Bayati, M., Montanari, A.: The dynamics of message passing on dense graphs, with applications to compressed sensing. IEEE Trans. Inf. Theory **57**(2), 764–785 (2011)
4. Birgmeier, S.: Message passing for multidimensional inverse problems. Ph.D. thesis, Technical University Vienna (2019)

5. Blumensath, T., Davies, M.E.: Iterative thresholding for sparse approximations. J. Fourier Anal. Appl. **14**(5), 629–654 (2008)
6. Blumensath, T., Davies, M.E.: Iterative hard thresholding for compressed sensing. Appl. Comput. Harmon. Anal. **27**(3), 265–274 (2009)
7. Borgerding, M., Schniter, P.: Onsager-corrected deep learning for sparse linear inverse problems. In: IEEE Global Conference on Signal and Information Processing (GlobalSIP), pp. 227–231 (2016)
8. Boufounos, P.T., Baraniuk, R.G.: 1-bit compressive sensing. In: Annual Conference Information Sciences and Systems (CISS), pp. 16–21 (2008)
9. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press, New York (2004)
10. Boyd, S., Vandenberghe, L.: Introduction to Applied Linear Algebra: Vectors, Matrices, and Least Squares. Cambridge University Press, New York (2018)
11. Candès, E.J., Tao, T.: Decoding by linear programming. IEEE Trans. Inf. Theory **51**(12), 4203–4215 (2005)
12. Candès, E.J., Wakin, M.B.: An introduction to compressive sampling. IEEE Signal Process. Mag. **25**(2), 21–30 (2008)
13. Candès, E.J., Romberg, J., Tao, T.: Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. IEEE Trans. Inf. Theory **52**(2), 489–509 (2006)
14. Daubechies, I., Fornasier, M., Loris, I.: Accelerated projected gradient method for linear inverse problems with sparsity constraints. J. Fourier Anal. Appl. **14**(5), 764–792 (2008)
15. Donoho, D.L.: Compressed sensing. IEEE Trans. Inf. Theory **52**(4), 1289–1306 (2006)
16. Donoho, D.L., Maleki, A., Montanari, A.: Message-passing algorithms for compressed sensing. Proc. Natl. Acad. Sci. **106**(45), 18914–18919 (2009)
17. Donoho, D.L., Maleki, A., Montanari, A.: Message passing algorithms for compressed sensing: I. Motivation and construction. In: IEEE Information Theory Workshop (ITW). Cairo (2010)
18. Dymarski, P., Romaniuk, R.: Sparse signal modeling in a scalable CELP coder. In: European Signal Processing Conf. (EUSIPCO). Marrakech (2013)
19. Eldar, Y.C., Kutyniok, G. (eds.): Compressed Sensing – Theory and Applications. Cambridge University Press, New York (2012)
20. Ens, A., Yousaf, A., Ostertag, T., Reindl, L.M.: Optimized sinus wave generation with compressed sensing for radar applications. In: International Workshop on Compressed Sensing applied to Radar (CoSeRa). Bonn (2013)
21. Fay, R.: Introducing the counter mode of operation to compressed sensing based encryption. Inf. Process. Lett. **116**(4), 279–283 (2016)
22. Fischer, R.F.H., Siegl, C.: Inflated lattice precoding, bias compensation, and the uplink/downlink duality: the connection. IEEE Commun. Lett. **11**(2), 185–187 (2007)
23. Fischer, R.F.H., Wäckerle, F.: Peak-to-average power ratio reduction in OFDM via sparse signals: transmitter-side tone reservation vs. receiver-side compressed sensing. In: International OFDM Workshop. Essen (2012)
24. Fischer, R.F.H., Sippel, C., Goertz, N.: VAMP with vector-valued diagonalization. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 9110–9114. Barcelona (2020)
25. Fletcher, A., Sahraee-Ardakan, M., Rangan, S., Schniter, P.: Expectation consistent approximate inference: generalizations and convergence. In: IEEE International Symposium on Information Theory (ISIT), pp. 190–194. Barcelona (2016)
26. Flinth, A., Kutyniok, G.: PROMP: A sparse recovery approach to lattice-valued signals. Appl. Comput. Harmon. Anal. **45**(3), 668–708 (2018)
27. Forney, G.D.: On the role of MMSE estimation in approaching the information-theoretic limits of linear gaussian channels: shannon meets wiener. In: Allerton Conference on Communication, Control, and Computing. Monticello (2003)
28. Guo, Q., Huang, D.D.: A concise representation for the soft-in soft-out LMMSE detector. IEEE Commun. Lett. **15**(5), 566–568 (2011)

29. Guo, D., Wang, C.C.: Multiuser detection of sparsely spread CDMA. IEEE J. Sel. Areas Commun. **26**(3), 421–431 (2008)
30. Hernández-Lobato, J.M., Hernández-Lobato, D.: Convergent expectation propagation in linear models with spike-and-slab priors (2011). Preprint. http://arxiv.org/abs/1112.2289
31. Jacques, L., Degraux, K., Vleeschouwer, C.D.: Quantized iterative hard thresholding: bridging 1-bit and high-resolution quantized compressed sensing. In: International Conference on Sampling Theory and Applications (2013)
32. Jose, S., Simeone, O.: Free energy minimization: a unified framework for modeling, inference, learning, and optimization. IEEE Signal Process. Mag. **38**(2), 120–125 (2021)
33. Kay, S.M.: Fundamentals of Statistical Signal Processing: I. Estimation Theory. Prentice Hall Inc., Upper Saddle River (1993)
34. Keiper, S., Kutyniok, G., Lee, D.G., Pfander, G.E.: Compressed sensing for finite-valued signals. Linear Algebra Appl. **532**, 570–613 (2017)
35. Kschischang, F.R., Frey, B.J., Loeliger, H.A.: Factor graphs and the sum-product algorithm. IEEE Trans. Inf. Theory **47**(2), 498–519 (2001)
36. Kuske, J., Swoboda, P., Petra, S.: A novel convex relaxation for non-binary discrete tomography. In: Lauze, F., Dong, Y., Dahl, A. (eds.) Scale Space and Variational Methods in Computer Vision, pp. 235–246. Springer International Publishing, Berlin (2017)
37. Loeliger, H.A., Dauwels, J., Hu, J., Korl, S., Ping, L., Kschischang, F.R.: The factor graph approach to model-based signal processing. Proc. IEEE **95**(6), 1295–1322 (2007)
38. Ma, J., Ping, L.: Orthogonal AMP. IEEE Access **5**, 2020–2033 (2017)
39. Maleki, A.: Approximate message passing algorithms for compressed sensing. Ph.D. Thesis, Stanford University (2011)
40. Maleki, A., Montanari, A.: Analysis of approximate message passing algorithm. In: Annual Conference on Information Sciences and Systems (CISS). Princeton (2010)
41. Minka, T.: A family of algorithms for approximate Bayesian inference. Ph.D. Thesis, Massachusetts Institute of Technology (2001)
42. Moulin, P., Veeravalli, V.V.: Statistical Inference For Engineers and Data Scientists. Cambridge University Press, New York (2019)
43. Needell, D., Tropp, J.A.: Cosamp: Iterative signal recovery from incomplete and inaccurate samples. Appl. Comput. Harmon. Anal. **26**(3), 301–321 (2009)
44. Needell, D., Tropp, J.A.: Cosamp: Iterative signal recovery from incomplete and inaccurate samples. Commun. ACM **53**(12), 93–100 (2010)
45. Nehmhauser, G., Wolsey, L.: Integer and Combinatorial Optimization. John Wiley & Sons, New York (1988)
46. Opper, M., Winther, O.: Expectation consistent approximate inference. J. Mach. Learn. Res. **6**, 2177–2204 (2005)
47. Padberg, M., Rinaldi, G.: A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems. SIAM Rev. **33**(1), 60–100 (1991)
48. Papoulis, A., Pillai, S.U.: Probability, Random Variables, and Stochastic Processes, 4th edn. McGraw-Hill, New York (2001)
49. Pati, Y.C., Rezaiifar, R., Krishnaprasad, P.S.: Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition. In: Asilomar Conference on Signals, Systems, and Computer, pp. 40–44. Pacific Grove (1993)
50. Proakis, J.: Digital Communications, 5th edn. McGraw-Hill, New York (2008)
51. Rangan, S.: Estimation with random linear mixing, belief propagation and compressed sensing. In: Annual Conference on Information Sciences and Systems (CISS). Princeton (2010)
52. Rangan, S.: Generalized approximate message passing for estimation with random linear mixing. In: IEEE International Symposium on Information Theory (ISIT), pp. 2168–2172. St. Petersburg, Russia (2011)
53. Rangan, S., Schniter, P., Fletcher, A.K.: Vector approximate message passing. In: IEEE International Symposium on Information Theory (ISIT), pp. 1588–1592. Aachen (2017)
54. Rangan, S., Schniter, P., Fletcher, A.K.: Vector approximate message passing. IEEE Trans. Inf. Theory **65**(10), 6664–6684 (2019)

55. Schniter, P.: A simple derivation of AMP and its state evolution via first-order cancellation. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). Barcelona (2020)
56. Shore, J.E., Johnson, R.W.: Axiomatic derivation of the principle of maximum entropy and the principle of minimum cross-entropy. IEEE Trans. Inf. Theory **26**(1), 26–37 (1980)
57. Sippel, C., Fischer, R.F.H.: On the invariance of recovery algorithms for compressed sensing based on expectation-consistent approximate inference. In: 24th International ITG Workshop on Smart Antennas. Hamburg (2020)
58. Sippel, C., Fischer, R.F.H.: VAMP with individual variances and sequential processing for compressed sensing. In: European Signal Processing Conference (EUSIPCO). Dublin (2021)
59. Sparrer, S.: Algorithms for (discrete) compressed sensing – a communications engineering perspective. Ph.D. Thesis, Ulm University (2019)
60. Sparrer, S., Fischer, R.F.H.: Soft-feedback OMP for the recovery of discrete-valued sparse signals. In: European Signal Processing Conference (EUSIPCO). Nice (2015)
61. Sparrer, S., Fischer, R.F.H.: Enhanced iterative hard thresholding for the estimation of discrete-valued sparse signals. In: European Signal Processing Conference (EUSIPCO). Budapest (2016)
62. Sparrer, S., Fischer, R.F.H.: MMSE-based version of OMP for recovery of discrete-valued sparse signals. Electron. Lett. **52**(1), 75–77 (2016)
63. Sparrer, S., Fischer, R.F.H.: Algorithms for the iterative estimation of discrete-valued sparse vectors. In: International ITG Conference on Systems, Communication, and Coding (SCC). Hamburg (2017)
64. Sparrer, S., Fischer, R.F.H.: Bias compensation in iterative soft-feedback algorithms with application to (discrete) compressed sensing. In: IEEE Statistical Signal Processing Workshop. Freiburg (2018)
65. Sparrer, S., Schenk, A., Fischer, R.F.H.: Communication over impulsive noise channels: Channel coding vs. compressed sensing. In: International ITG Conference on Systems, Communication, and Coding (SCC). Munich (2013)
66. Tibshirani, R.: Regression shrinkage and selection via the lasso. J. R. Stat. Soc. Ser. B Methodol. **58**, 267–288 (1996)
67. Yedidia, J.S., Freeman, W.T., Weiss, Y.: Constructing free-energy approximations and generalized belief propagation algorithms. IEEE Trans. Inf. Theory **51**(7), 2282–2312 (2005)
68. Zhu, H., Giannakis, G.B.: Exploiting sparse user activity in multiuser detection. IEEE Trans. Commun. **59**(2), 454–465 (2011)
69. Zörlein, H., Lazich, D., Bossert, M.: Performance of error correction based on compressed sensing. In: International Symposium on Wireless Communication Systems, pp. 301–305. Aachen (2011)
70. Zymnis, A., Boyd, S., Candes, E.: Compressed sensing with quantized measurements. IEEE Signal Process Lett. **17**(2), 149–152 (2010)

# Chapter 7
# Recovery Under Side Constraints


**Khaled Ardah, Martin Haardt, Tianyi Liu, Frederic Matter, Marius Pesavento, and Marc E. Pfetsch**

## 7.1 Introduction

Compressed sensing (CS) is a signal processing technique for efficient acquisition and reconstruction of signals based on an underlying model sparsity, which allows to recover the signal of interest from far fewer samples than required by traditional acquisition systems operating at Nyquist rate. Theoretical recovery guarantees on the number of observations required can be further enhanced if side information on the measurement system and the signal representation is incorporated in the form of additional side constraints that are enforced in the recovery process. The measurement system may be subject to various types of side constraints that can be exploited and may originate from: *i)* the structure of the sensing matrix (shift-invariance, block structure, sparse co-array structures [60], etc.), *ii)* the structure of the sparse representation vector (integrality, variable bounds, unit-modulus, etc.), *iii)* the sparsity structure in the multiple snapshot case (block- and group-sparsity, rank-sparsity, etc.), as well as *iv)* the structure of the measurements (quantization effects, K-bit measures, magnitude-only measurements, etc.). A fundamental question that arises in this context is, in which sense structural information can be incorporated into the CS problem and how it affects the existing algorithms and theoretical results.

K. Ardah · M. Haardt
Communications Research Laboratory, Ilmenau University of Technology, Ilmenau, Germany
e-mail: khaled.ardah@tu-ilmenau.de; martin.haardt@tu-ilmenau.de

T. Liu · M. Pesavento (✉)
Institut für Nachrichtentechnik, TU Darmstadt, Darmstadt, Germany
e-mail: tliu@nt.tu-darmstadt.de; pesavento@nt.tu-darmstadt.de

F. Matter · M. E. Pfetsch
Department of Mathematics, TU Darmstadt, Darmstadt, Germany
e-mail: matter@mathematik.tu-darmstadt.de; pfetsch@mathematik.tu-darmstadt.de



G. Kutyniok et al. (eds.), *Compressed Sensing in Information Processing*, Applied and Numerical Harmonic Analysis, https://doi.org/10.1007/978-3-031-09745-4_7

Moreover, recovery from nonlinear measurements with sparse models has recently been investigated, e.g., in the classical phase retrieval problem, where different forms of redundancy have been incorporated through the use of known or unknown linear mixing networks. Redundancy can further enhance recovery in this case.

A large variety of applications involve data recorded from large-scale sensor arrays or massive *multiple-input multiple-output* (MIMO) arrays, which consist of an assembly of wideband sensors to meet the corresponding high-throughput and high-resolution requirements. In this context, sparsity naturally arises in the angular domain, e.g., in the form of discrete propagation models and a small number of impinging signals from different directions. Similarly, in sensor array and MIMO applications, the structure of the array, the properties of the constellation signal and the transmitted signal provide important prior information. In order to keep hardware costs in these large-scale systems at a reasonable scale while retaining high performance, mixed analog–digital sensing system designs are employed to reduce the number and the sampling rates of the analog-to-digital converters as well as the quality requirements (e.g., w.r.t. linearity, dynamic range) of the hardware components.

This chapter reviews recent developments on sparse recovery guarantees and efficient recovery algorithms in CS networks under the aforementioned side constraints in the context of multi-antenna systems. First, CS with linear and nonlinear measurement models and the corresponding recovery problems are introduced in Sect. 7.2. Theoretical results on the recoverability of linear CS measurements under side constraints are presented in Sect. 7.3. Recovery algorithms for sparse measurements under side constraints are addressed in Sect. 7.4, and a new linear mixing matrix design is proposed in Sect. 7.5. Finally, phase retrieval for known and unknown dictionaries is discussed in Sect. 7.6, before conclusions are drawn in Sect. 7.7.

## 7.2 Sparse Recovery in Sensor Arrays

Consider, as one prominent example application, the following sparse one-dimensional narrow-band array processing model that is frequently encountered in the context of direction-of-arrival (DoA) estimation [2, 6, 18, 30, 59, 70] and multiple-input multiple-output (MIMO) communication [15] and that will be used as a generic example in subsequent sections. We assume that $K$ far-field narrow-band source signals impinge on a sensor array composed of $M$ omni-directional sensors as depicted in the right-hand side of Fig. 7.1. The $t$-th time sample of the array output vector $\mathbf{y}(t) = [y_1(t), \ldots, y_M(t)]^{\mathrm{T}} \in \mathbb{C}^M$ is given by

$$\mathbf{y}(t) = \mathbf{A}(\boldsymbol{\theta}^{(0)})\, \mathbf{x}^{(0)}(t) + \mathbf{n}(t), \qquad t = 1, \ldots, D, \tag{7.1}$$

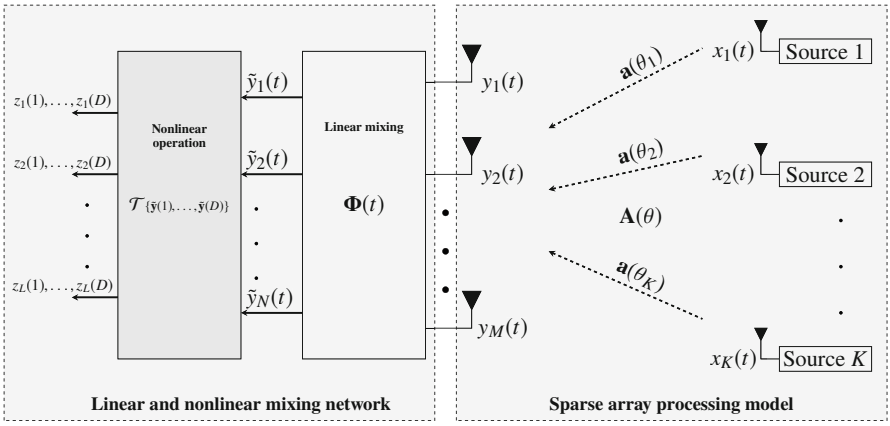



**Fig. 7.1** Sparse array processing model with linear and nonlinear mixing network

where $\mathbf{x}^{(0)}(t) = [x_1^{(0)}(t), \ldots, x_K^{(0)}(t)]^{\mathrm{T}} \in \mathbb{C}^K$ is the vector of signals emitted by the $K$ sources, $\mathbf{n}(t) \in \mathbb{C}^M$ contains the spatially and temporally white circular Gaussian sensor noise, and $D$ is the number of available time samples. The matrix $\mathbf{A}(\boldsymbol{\theta}^{(0)}) = [\mathbf{a}(\theta_1), \ldots, \mathbf{a}(\theta_K)] \in \mathbb{C}^{M \times K}$ denotes the true array steering matrix, whose $i$-th column is the array response vector $\mathbf{a}(\theta_i)$ corresponding to the $i$-th source with DoA $\theta_i \in \Theta$, where $\Theta$ defines the field of view. The steering vector $\mathbf{a}(\theta)$ describes a manifold denoted as $\mathbb{M}^M$. For example, for a uniform linear array (ULA) with half-wavelength inter-element spacing, $\mathbf{a}(\theta)$ is given by $\mathbf{a}(\theta) = [1, e^{-j\pi \sin(\theta)}, \ldots, e^{-j(M-1)\pi \sin(\theta)}]^{\mathrm{T}}$. We denote $\boldsymbol{\theta}^{(0)} = [\theta_1^{(0)}, \ldots, \theta_K^{(0)}]^{\mathrm{T}}$ as the true DoA parameter vector.

### *7.2.1 Compressive Data Model for Sensor Arrays*

The model in (7.1) presumes a dedicated radio frequency (RF) receiver chain for each individual antenna element including an LNA, filters, down-conversion, analog-to-digital converter (ADC), etc. In many applications, however, such separate RF chains for each antenna element come at a high cost in terms of the overall receiver complexity and power consumption. To reduce the number of RF channels (and time samples) without loss in the array aperture, compressed sensing can be applied, where the antenna outputs are linearly combined in the analog domain and then passed through a reduced number of RF chains to obtain the digital baseband signals as illustrated in the left-hand side of Fig. 7.1. This can be realized in hardware, e.g., by using configurable hardware components such as tunable phase shifters, a bank of fixed analog beamformers combined with a

fast switching network that enables analog beamformer selection, and/or a band of (tunable) bandpass filters. This way, $N \leq M$ RF receiver channels are used for signal processing in the digital domain.

Let $\mathbf{\Phi}^{(0)}(t) \in \mathbb{C}^{N\times M}$ denote the complex analog mixing matrix of a compressive array at time $t$, which compresses the output of $M$ antenna elements to $N$ active RF channels. Then, the complex (baseband) array output (7.1) after combining can be expressed as

$$\tilde{\mathbf{y}}(t) = \mathbf{\Phi}^{(0)}(t)\big(\mathbf{A}(\boldsymbol{\theta}^{(0)})\,\mathbf{x}^{(0)}(t) + \mathbf{n}(t)\big) + \mathbf{w}(t), \qquad t = 1, \ldots, D, \tag{7.2}$$

where $[\mathbf{\Phi}^{(0)}(t)]_{n,m} = \alpha_{n,m}(t) \cdot e^{j\varphi_{n,m}(t)}$, $n = 1, \ldots, N$, $m = 1, \ldots, M$ with $\alpha_{n,m}(t) \in [0, 1]$, $\varphi_{n,m}(t) \in [0, 2\pi]$, and $\mathbf{w}(t) \in \mathbb{C}^N$ contains the spatially and temporally white circular Gaussian measurement noise. Signals may be subject to additive noise that acts before (i.e., in the form of $\mathbf{n}(t)$) or after the mixing network (i.e., in the form of $\mathbf{w}(t)$). Defining the effective array steering matrix $\tilde{\mathbf{A}}(\boldsymbol{\theta}^{(0)}, t) = \mathbf{\Phi}^{(0)}(t)\mathbf{A}(\boldsymbol{\theta}^{(0)})$, Model (7.2) becomes

$$\tilde{\mathbf{y}}(t) = \tilde{\mathbf{A}}(\boldsymbol{\theta}^{(0)}, t)\,\mathbf{x}^{(0)}(t) + \tilde{\mathbf{n}}(t), \tag{7.3}$$

where $\tilde{\mathbf{n}}(t) = \mathbf{\Phi}^{(0)}(t)\mathbf{n}(t) + \mathbf{w}(t)$ is the effective noise vector.

Cost-efficient analog hardware devices and data acquisition systems generally involve nonlinear transformations that can perform further compression. Such nonlinear transformations are indicated by the operator $\mathcal{T}$, which performs a nonlinear mapping from $\mathbb{C}^{N\times D}$ to $\mathbb{C}^{L\times D}$ as depicted in Fig. 7.1. The types of nonlinearity consist, for instance, of nonlinear transformations introduced from low-cost power amplifiers, magnitude-only, and sub-band power measurements that are often used in cellular communications, $C$-bit quantization, the more aggressive 1-bit quantization (sign-only measurements), hard-thresholding, and soft-thresholding, or modulo operations. Considering the $D$ time samples simultaneously, the resulting measurement matrix $\mathbf{Z} = [\mathbf{z}(1), \ldots, \mathbf{z}(D)] \in \mathbb{C}^{L\times D}$ recorded at the output of the nonlinear mixing network is given by

$$\mathbf{Z} = \mathcal{T}\big\{\mathbf{\Phi}^{(0)}(1)\mathbf{A}(\boldsymbol{\theta}^{(0)})\,\mathbf{x}^{(0)}(1), \ldots, \mathbf{\Phi}^{(0)}(D)\mathbf{A}(\boldsymbol{\theta}^{(0)})\,\mathbf{x}^{(0)}(D)\big\} + \mathbf{N}, \tag{7.4}$$

where $\mathbf{N} \in \mathbb{C}^{L\times D}$ combines the various noise contributions. If the mixing matrix $\mathbf{\Phi}^{(0)}(t)$ is time-invariant, i.e., $\mathbf{\Phi}^{(0)}(t) = \mathbf{\Phi}^{(0)}$, the model (7.4) reduces to

$$\mathbf{Z} = \mathcal{T}\big\{\mathbf{\Phi}^{(0)}\mathbf{A}(\boldsymbol{\theta}^{(0)})\,\mathbf{X}^{(0)}\big\} + \mathbf{N}, \tag{7.5}$$

where $\mathbf{X}^{(0)} = [\mathbf{x}^{(0)}(1), \ldots, \mathbf{x}^{(0)}(D)] \in \mathbb{C}^{K\times D}$ comprises the $D$ time snapshots.

### 7.2.2 Sparse Recovery Formulations for Sensor Arrays

Based on (7.5), we aim to solve the sparse recovery problem that allows for a robust and efficient estimation of the frequencies of the $K$ sources $x_k^{(0)}(t)$ from the set of measurements $\mathbf{z}(t)$ while exploiting potential structure in $\boldsymbol{\Phi}(t)$, $\tilde{\mathbf{A}}$, $\mathbf{A}$, and $\mathbf{x}^{(0)}(t)$, or specific properties of $\mathcal{T}$. Specifically, we will address variations of the general multiple-measurement $\ell_{p,q}$ mixed norm minimization problem

$$\min_{\mathbf{X},\boldsymbol{\Phi}} \tfrac{1}{2} \left\| \mathbf{Z} - \mathcal{T}\left\{ \boldsymbol{\Phi}\mathbf{A}(\boldsymbol{\theta})\mathbf{X} \right\} \right\|_{\mathrm{F}}^2 + \lambda \|\mathbf{X}\|_{p,q} \ : \ \textit{side constraints}, \tag{P0}$$

where at this point $\boldsymbol{\Phi}$ is assumed to be time-invariant for simplicity of description (i.e., considering (7.5)), $\mathbf{A}(\boldsymbol{\theta}) \in \mathbb{C}^{M \times P}$ with $P \gg M$ is a "fat" sensing matrix corresponding to the $P$-dimensional DoA grid vector $\boldsymbol{\theta}$ that appropriately samples the field of view $\Theta$, and $\mathbf{X} \in \mathbb{C}^{P \times D}$ is the row-sparse (joint-sparse) signal matrix of interest, i.e., its columns share the same support. The support of the non-zero rows of $\mathbf{X}$ corresponds to the DoAs on the spatial grid. Moreover, the regularization parameter $\lambda > 0$ controls the trade-off between the data-fitting term and the sparsity level in $\mathbf{X}$. The joint-sparsity in $\mathbf{x}$ is induced by the $\ell_{p,q}$ mixed norm defined as

$$\|\mathbf{X}\|_{p,q} = \left( \sum_{k=1}^{P} \|\mathbf{x}_k\|_p^q \right)^{1/q}, \tag{7.6}$$

for $p, q \geq 1$, which applies an inner $\ell_p$-norm to the rows $\mathbf{x}_k$, $k = 1, \ldots, P$ in $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_P]^{\mathrm{T}}$ and an outer $\ell_q$-norm to the $\ell_p$-row-norms. Ideally, we aim to solve (P0) using the $\ell_{p,0}$-pseudo-norm $\|\mathbf{X}\|_{p,0}$, which is the cardinality of the non-zero $\ell_p$-norms of the rows of $\mathbf{X}$. If $D = 1$, the model reduces to the single-measurement case and the $\ell_{p,1}$-mixed-norm reduces to the $\ell_1$-norm.

In the absence of the various noise contributions, i.e., $\mathbf{N} = \mathbf{0}$, the general minimization problem (P0) can be equivalently written as

$$\min_{\mathbf{X},\boldsymbol{\Phi}} \left\{ \|\mathbf{X}\|_{p,q} \ : \ \mathcal{T}\{\boldsymbol{\Phi}\mathbf{A}(\boldsymbol{\theta})\mathbf{X}\} = \mathbf{Z}, \textit{additional side constraints} \right\}. \tag{7.7}$$

## 7.3 Recovery Guarantees Under Side Constraints

In this section, we consider the uniform recovery of sparse solutions with additional side constraints on the solutions/signals. We use the signal model (7.1) without noise in the single-measurement case, i.e., $\mathbf{n} = \mathbf{0}$ and $D = 1$. More precisely, consider the equation system $\mathbf{A}\mathbf{x} = \mathbf{y}$ for $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\mathbf{y} \in \mathbb{R}^m$. The side constraints for $\mathbf{x}$ can be expressed by requiring that $\mathbf{x} \in C \subseteq \mathbb{R}^n$. This leads to optimization models

$$\min \{\|\mathbf{x}\|_0 \ : \ \mathbf{A}\mathbf{x} = \mathbf{y},\ \mathbf{x} \in C\}, \tag{7.8}$$

i.e., variants of (7.7) in the single-measurement case without nonlinearities, which promise to be able to uniquely recover sparse solutions for a larger set of right-hand side vectors $\mathbf{y}$. This is illustrated by the following very simple toy example.

*Example 7.1* Consider the following recovery problem for $n = 2$. Let $\mathbf{A} = [1, -1]$ and $y = 1$. The system $\mathbf{A}\mathbf{x} = y$ has two sparse solutions, namely $\mathbf{x}_1 = (1, 0)^{\mathrm{T}}$ and $\mathbf{x}_2 = (0, -1)^{\mathrm{T}}$. Since $\|\mathbf{x}_1\|_1 = \|\mathbf{x}_2\|_1 = 1$, it is not possible to *uniquely* recover either point by $\ell_1$-minimization or that by $\ell_0$-minimization. But by exploiting nonnegativity, $\mathbf{x}_1$ can indeed be uniquely recovered.

Another example of a whole family of sensing matrices showing that exploiting side constraints leads to weaker recovery conditions can be found in [22, Theorem 4.5]. This shows that side constraints are not only of theoretical interest but should be exploited in the recovery process. The price to pay may of course be that the recovery problems become harder to solve.

### 7.3.1 Integrality Constraints

One particular example of an interesting side constraint is the integrality of $\mathbf{x}$. Applications include discrete tomography [31] or massive MIMO with constellation signals [20, 21]. A notable special case of this setting includes the recovery of binary vectors, which has applications in digital or wireless communication systems.

The corresponding general recovery problem can be formulated as

$$\min \{\|\mathbf{x}\|_0 \ : \ \mathbf{A}\mathbf{x} = \mathbf{A}\mathbf{x}^{(0)},\ \mathbf{x} \in [\boldsymbol{\ell}, \mathbf{u}]_{\mathbb{Z}}\}, \tag{7.9}$$

where $\mathbf{x}^{(0)} \in [\boldsymbol{\ell}, \mathbf{u}]_{\mathbb{Z}} \coloneqq \{\mathbf{x} \in \mathbb{Z}^n \ : \ \ell_i \le x_i \le u_i,\ i \in [n]\}$ is an $s$-sparse vector and $\mathbf{A} \in \mathbb{R}^{m \times n}$. Note that we can assume $\boldsymbol{\ell} \in \mathbb{Z}^n \cup \{-\infty\}$ and $\mathbf{u} \in \mathbb{Z}^n \cup \{\infty\}$. As in the case of classical sparse recovery, we consider the $\ell_1$-relaxation of (7.9), namely

$$\min \{\|\mathbf{x}\|_1 \ : \ \mathbf{A}\mathbf{x} = \mathbf{A}\mathbf{x}^{(0)},\ \mathbf{x} \in [\boldsymbol{\ell}, \mathbf{u}]_{\mathbb{Z}}\}. \tag{7.10}$$

In the literature, recovery of binary and integral sparse vectors using (7.10) has been considered for example in [26, 62], where the nonconvex integrality condition was relaxed to $\mathbf{x} \in [\boldsymbol{\ell}, \mathbf{u}] \coloneqq \{\mathbf{x} \in \mathbb{R}^n \ : \ \boldsymbol{\ell} \le \mathbf{x} \le \mathbf{u}\}$. In this case, the prior knowledge of $\mathbf{x}$ being integral does not help for recovery: uniform recovery of all sparse bounded integral $\mathbf{x}$ is equivalent to uniform recovery of all sparse bounded $\mathbf{x}$, see [26]. This already shows that in order to exploit integrality, one has to take this into account in the recovery program. Note that (7.10) is nonconvex but can be formulated as a mixed-integer (linear) program (MIP). Furthermore, note that both (7.9) and (7.10) are $\mathcal{NP}$-hard [34].

It turns out that in case of rational measurement matrices $\mathbf{A}$ and no bounds on the variables, there is again no difference between integral and general $\mathbf{x}$ [34]. However, in the presence of additional bounds, this is no longer true. In this case, it is possible to formulate null space properties depending on the bounds $\boldsymbol{\ell}$, $\mathbf{u}$ that characterize uniform recovery of integral (bounded) sparse vectors $\mathbf{x}$ using (7.10), see [34]. To this end, define the following two *null space properties* (NSP) depending on a set $V \subseteq \mathbb{R}^n$. Let $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $S \subseteq [n] := \{1, \ldots, n\}$ and define

$$\begin{aligned} \mathrm{NSP}(V): &\qquad \|\mathbf{v}_S\|_1 < \|\mathbf{v}_{\overline{S}}\|_1 \quad \forall\, \mathbf{v} \in (V \cap \mathcal{N}(\mathbf{A})) \setminus \{\mathbf{0}\}, \\ \mathrm{NSP}_+(V): &\quad \mathbf{v}_{\overline{S}} \leq \mathbf{0} \implies \sum_{i=1}^{n} v_i < 0 \quad \forall\, \mathbf{v} \in (V \cap \mathcal{N}(\mathbf{A})) \setminus \{\mathbf{0}\}, \end{aligned}$$

where $\overline{S}$ denotes the complement of a set $S$, $\mathbf{v}_S$ denotes the vector of elements indexed by $S$, and $\mathcal{N}(\mathbf{A})$ denotes the null space of the matrix $\mathbf{A}$.

Then, $\mathrm{NSP}(\mathbb{R}^n)$ is the classical null space property [14, 16] that characterizes uniform recovery of sparse vectors $\mathbf{x}$ by $\ell_1$-minimization, and $\mathrm{NSP}_+(\mathbb{R}^n)$ is the well-known nonnegative null space property [27, 87] characterizing uniform recovery via nonnegative $\ell_1$-minimization.

For integral vectors without bounds, i.e., $\ell_i = -\infty$ and $u_i = \infty$ for all $i \in [n]$, and integral nonnegative vectors, the results for uniform recovery are completely analogous to the classical case with the only exception that for satisfying the NSP, only integral vectors in the null space of $\mathbf{A}$ are of interest, see [34] for the exact statements. This observation also shows that for $\mathbf{A} \in \mathbb{Q}^{m \times n}$, the classical (nonnegative) NSP and the corresponding integral (nonnegative) NSP coincide, since for $\mathbf{A} \in \mathbb{Q}^{m \times n}$, all vectors in the null space of $\mathbf{A}$ can be scaled to integrality. Thus, for rational data, exploiting integrality does not lead to improved recovery conditions.

If the bounds $\boldsymbol{\ell}$, $\mathbf{u}$ are nontrivial, the situation changes fundamentally. The first difference is that for classical recovery, bounds on $\mathbf{x}$ do not influence recovery properties since vectors in the null space of $\mathbf{A}$ can be scaled accordingly. For integral vectors in the presence of bounds $-\infty \leq \ell_i \leq 0 \leq u_i \leq \infty$ for all $i \in [n]$, however, a new NSP arises. It turns out that the condition $\mathrm{NSP}([\boldsymbol{\ell} - \mathbf{u}, \mathbf{u} - \boldsymbol{\ell}]_{\mathbb{Z}})$ is only sufficient but not necessary for uniform recovery using (7.10). Nevertheless, we can use a variable split into positive and negative parts to obtain an NSP that characterizes uniform recovery in the following statement.

**Theorem 7.1 ([34])** *Let $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $s \geq 0$. Then every $s$-sparse vector $\mathbf{x}^{(0)} \in [\boldsymbol{\ell}, \mathbf{u}]_{\mathbb{Z}}$ is the unique solution of (7.10) if and only if*

$$-(\mathbf{v}_{\overline{S}}, \mathbf{w}_{\overline{S}})^T \in K \quad \implies \quad \sum_{i=1}^{n} v_i + w_i < 0,$$

*holds for all* $(\mathbf{v}, \mathbf{w})^T \in \mathcal{N}((\mathbf{A}, -\mathbf{A})) \cap (K + (-K))$ *with* $(\mathbf{v}, \mathbf{w})^T \neq (\mathbf{0}, \mathbf{0})^T$ *and all* $S \subseteq [n]$, $|S| \leq s$, *where*

$$K := \left\{ \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} \in \left[ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \mathbf{u} \\ -\boldsymbol{\ell} \end{pmatrix} \right]_{\mathbb{Z}} : x_i \cdot y_i = 0,\ i \in [n] \right\}.$$

The complementarity constraints $x_i \cdot y_i = 0$ in $K$ are due to the split into positive and negative parts. This already shows that the introduction of bounds leads to different recovery conditions, in contrast to the situation of classical sparse recovery over $\mathbb{R}^n$. For testing the NSP in Theorem 7.1, one needs to take care of the complementarity constraints $x_i \cdot y_i = 0$. This can be done by, e.g., using methods from [11, 12]. For nonnegative integral vectors with upper bounds, the variable split is not needed, and it can be shown that $\mathrm{NSP}_+([-\mathbf{u}, \mathbf{u}]_{\mathbb{Z}})$ characterizes uniform recovery [34].

Besides using (7.10) for recovery of sparse integral vectors, one can also use the exact recovery problem (7.9), which can be formulated as a MIP if there are finite bounds by expressing the nonconvex $\ell_0$-objective using binary variables (recall that (7.9) and (7.10) are $\mathcal{NP}$-hard [34]). In this case, it is also possible to characterize when solving (7.9) recovers any $s$-sparse integral vector with or without bounds. The condition for classical sparse recovery using $\ell_0$-minimization is $\mathrm{spark}(\mathbf{A}) > 2s$, where $\mathrm{spark}(\mathbf{A})$ denotes the smallest number of linear dependent columns in $\mathbf{A}$. The corresponding statements for integral sparse recovery using (7.9) appear in [34].

### 7.3.2 General Framework for Arbitrary Side Constraints

In the previous section, we have explicitly considered integrality constraints as one specific side constraint that can be added to the recovery problem to obtain stronger recovery guarantees. The corresponding recovery conditions resemble the well-known null space properties that exist for various other settings such as sparse (nonnegative) recovery [14, 16, 27, 87], block-sparse recovery [63], or low-rank (positive semidefinite) matrix recovery [28, 45]. Thus it seems reasonable to search for a general setting and null space property that unifies the cases already considered in the literature. Such a general framework is presented in [25] that comprises all the previously mentioned settings but does not handle additional side constraints such as nonnegativity, integrality, and positive semidefiniteness. Sparsity in this general framework is expressed using projections. Recently, this general framework was extended in [22] to also cover additional side constraints. Under mild assumptions on the side constraints and the measurement process, it is possible to state an NSP for the corresponding general recovery problem. It turns out that this general NSP specializes to the already known NSPs in the various special cases mentioned above. In the following, we will shortly describe this general recovery framework and provide an application in order to evaluate the influence of side constraints.

For the general framework, we need two finite-dimensional Euclidean spaces $\mathcal{X}$ and $\mathcal{E}$. A *linear sensing map* $A\colon \mathcal{X} \to \mathbb{R}^m$ is used for acquiring signals $x \in \mathcal{X}$, and a *linear representation map* $B\colon \mathcal{X} \to \mathcal{E}$ is used for mapping a signal to an appropriate representation. We will denote the image of $x$ under a linear operator $F$ as $Fx$. Additional side constraints are modeled using a set $C \subseteq \mathcal{X}$ with $0 \in C$. The image of $C$ under the map $B$ is denoted with $\mathcal{D}$. Finally, let $\|\cdot\|$ be a norm on $\mathcal{E}$.

Sparsity in this general framework is expressed using projections onto appropriate subspaces. Therefore, let $\mathcal{P}$ be a set of matrices representing linear maps on $\mathcal{E}$. Each $P \in \mathcal{P}$ is assigned a nonnegative real weight by $\nu\colon \mathcal{P} \to \mathbb{R}_+$ and another linear map $\overline{P}\colon \mathcal{E} \to \mathcal{E}$. Then, for $s \in \mathbb{R}_+$, an element $\mathbf{x} \in \mathcal{X}$ is called *s-sparse*, if there exists a linear map $P \in \mathcal{P}$ with $\nu(P) \leq s$ and $PB\mathbf{x} = B\mathbf{x}$. Furthermore, let $\mathcal{P}_s = \{P \in \mathcal{P} : \nu(P) \leq s\}$ be the set of linear maps that induce $s$-sparse elements.

The corresponding generalized recovery problem for a given right-hand side $\mathbf{y} \in \mathbb{R}^m$ can be formulated as

$$\min \{\|B\mathbf{x}\| : A\mathbf{x} = \mathbf{y},\ \mathbf{x} \in C\}. \tag{7.11}$$

Note that this is convex if $C$ is convex. Using this general framework, it is possible to state two NSPs that can be used to characterize uniform recovery using the general recovery problem (7.11).

**Definition 7.1** The linear sensing map $A$ satisfies the *general null space property of type I* and *type II* of order $s$ for the set $C$ if and only if for all $\mathbf{v} \in (\mathcal{N}(A) \cap (C + (-C)))$ with $B\mathbf{v} \neq \mathbf{0}$ and all $P \in \mathcal{P}_s$, it holds that

$$-\overline{P}B\mathbf{v} \in \mathcal{D} \implies \exists\, \mathbf{v}^{(1)}, \mathbf{v}^{(2)} \in C,\ \mathbf{v} = \mathbf{v}^{(1)} - \mathbf{v}^{(2)},\ \|PB\mathbf{v}^{(1)}\| - \|PB\mathbf{v}^{(2)}\| < \|\overline{P}B\mathbf{v}\|, \tag{NSP-I$^C$}$$

$$-\overline{P}B\mathbf{v} \in \mathcal{D} \implies \forall\, \mathbf{v}^{(1)}, \mathbf{v}^{(2)} \in C,\ \mathbf{v} = \mathbf{v}^{(1)} - \mathbf{v}^{(2)} : \|PB\mathbf{v}^{(1)}\| - \|PB\mathbf{v}^{(2)}\| < \|\overline{P}B\mathbf{v}\|, \tag{NSP-II$^C$}$$

respectively, where $\mathcal{N}(A) \coloneqq \{\mathbf{v} \in \mathcal{X} : A\mathbf{v} = \mathbf{0}\}$ is the null space of $A$.

*Example 7.2 (Recovery of Sparse Nonnegative Vectors by $\ell_1$-Minimization)* For the recovery of nonnegative vectors, let $\mathcal{X} = \mathcal{E} = \mathbb{R}^n$, $B$ be the identity, and $\|\cdot\| = \|\cdot\|_1$. The set of side constraints is $C = \mathbb{R}^n_+$, implying $\mathcal{D} = \mathbb{R}^n_+$. Let $\mathcal{P}$ be the set of orthogonal projectors onto all coordinate subspaces of $\mathbb{R}^n$, and define $\overline{P} \coloneqq I_n - P$, where $I_n$ denotes the identity mapping on $\mathbb{R}^n$. Define the nonnegative weight $\nu(P) \coloneqq \mathrm{rk}(P)$, so that $\nu(P)$ is the number of non-zero components of the subspace $P$ projects onto. The notion of general sparsity reduces to the classical sparsity of non-zero entries in a vector $\mathbf{x} \in \mathbb{R}^n_+$, and the recovery problem (7.11) becomes nonnegative $\ell_1$-minimization with $PB\mathbf{x} = \mathbf{x}_S$ and $\overline{P}B\mathbf{x} = \mathbf{x}_{\overline{S}}$. In this case, it can be shown that the general null space property (NSP-I$^C$) of order $s$ for the set $C$ is equivalent to the known nonnegative null space property [27, 87]

$$\mathbf{v}_{\overline{S}} \leq 0 \implies \sum_{i \in S} v_i < \|\mathbf{v}_{\overline{S}}\|_1, \ \forall \mathbf{v} \in \mathcal{N}(A) \setminus \{\mathbf{0}\}, \ \forall S \subseteq [n], \ |S| \leq s, \tag{NSP$_{\geq 0}$}$$

where $S$ denotes the index set of components on which $P$ projects.

Under mild assumptions, the null space properties (NSP-I$^C$) and (NSP-II$^C$) can be proven to characterize uniform recovery using (7.11). Which NSP is needed depends on which assumptions are satisfied; see [22] for the formal statement. More examples of how the various settings already considered in the literature turn out to be special cases of this general recovery statement can also be found in [22]. At this point, it is important to notice that already in the special case of sparse vectors, checking whether $\mathbf{A}$ satisfies the classical NSP is $\mathcal{NP}$-hard [65].

The two NSPs characterizing uniform recovery in a very general framework already indicate that a stronger, i.e., more restrictive side, constraint leads to weaker conditions that need to be satisfied to guarantee uniform recovery.

In [22], an NSP for the recovery of positive semidefinite block-diagonal matrices is derived, which has not been considered before. Let $\mathbf{X} \in \mathcal{S}^n_+$ be a (symmetric) positive semidefinite matrix and $\mathcal{A}\colon \mathcal{S}^n \to \mathbb{R}^m$, $\mathcal{A}(\mathbf{X}) = (\mathbf{A}_1 \bullet \mathbf{X}, \ldots, \mathbf{A}_m \bullet \mathbf{X})^{\mathrm{T}}$ be a linear operator, where $\mathbf{A}_1, \ldots, \mathbf{A}_m \in \mathcal{S}^n$ are symmetric matrices and "$\bullet$" denotes the component-wise inner product. In order to define a block-diagonal form, let $k \geq 1$ and $B_1, \ldots, B_k \neq \emptyset$ be a partition of $[n]$. The matrix $\mathbf{X}$ and the linear measurement operator $\mathcal{A}(\mathbf{X})$ are in *block-diagonal form* with blocks $B_1, \ldots, B_k$, if $X_{s,t} = (A_i)_{s,t} = 0$ for all $(s,t) \notin (B_1 \times B_1) \cup \cdots \cup (B_k \times B_k)$ and all $i \in [m]$. Let $\mathbf{X}_B$ be the submatrix containing rows and columns of $\mathbf{X}$ indexed by $B$. The corresponding norm is given by the $\ell_{*,q}$-norm defined as

$$\|\mathbf{X}\|_{*,q} \coloneqq \|(\|\mathbf{X}_{B_1}\|_*, \ldots, \|\mathbf{X}_{B_k}\|_*)^{\mathrm{T}}\|_q,$$

and the block support $\mathrm{BS}(X)$ is given by the indices of those blocks $\mathbf{X}_{B_i} \neq \mathbf{0}$. By using an appropriate linear representation map to encode the block-diagonal structure, (NSP-I$^C$) simplifies to

$$\mathbf{V}_{B_i} \preceq 0 \ \forall i \in \overline{S} \quad \implies \quad \sum_{i \in S} \mathbf{1}^{\mathrm{T}} \lambda(\mathbf{V}_{B_i}) < \sum_{i \in \overline{S}} \|\mathbf{V}_{B_i}\|_*, \tag{NSP$^*_{*,1,\geq 0}$}$$

for all $\mathbf{V} \in (\mathcal{N}(A) \cap \mathcal{S}^n) \setminus \{\mathbf{0}\}$ and all $S \subseteq [k]$, $|S| \leq s$, where $\lambda(\mathbf{V}_{B_i})$ is the vector of eigenvalues of $\mathbf{V}_{B_i}$, and $\mathbf{1}$ is a vector of ones. The general uniform recovery statement [22, Theorem 2.7] yields the following theorem.

**Theorem 7.2 ([22])** *Let $A(\mathbf{X})$ be a linear operator in block-diagonal form and $s \geq 1$. Then, every positive semidefinite $\mathbf{X}^{(0)} \in \mathcal{S}^n_+$ with $\|\mathbf{X}^{(0)}\|_{*,0} \leq s$ is the unique solution of* $\min\{\|\mathbf{X}\|_{*,1} : A(\mathbf{X}) = \mathbf{b}, \ \mathbf{X} \succeq 0\}$ *with $\mathbf{b} = A(\mathbf{X}^{(0)})$ if and only if $A(\mathbf{X})$ satisfies* (NSP$^*_{*,1,\geq 0}$) *of order $s$.*

As a conclusion, the general framework presented above can answer many interesting questions concerning uniform recovery in the presence of side constraints using the optimization problem (7.11). The two general null space properties (NSP-I$^C$) and (NSP-II$^C$) can be used to analyze and quantify the exact impact of various side constraints in the recovery process. Given a specific setting, the NSPs can decide whether additional side information is needed or which side constraints need to be exploited in the recovery process to guarantee uniform recovery. For instance, this framework explains why there are two seemingly different NSP formulations for classical sparse recovery and nonnegative sparse recovery and their connection.

## 7.4 Recovery Algorithms Under Different Side Constraints for the Linear Measurement Model

### *7.4.1 Constant-Modulus Constraints*

In this section, we consider a variation of Problem (7.8) for the case of noisy measurements $\mathbf{s}$ and for side constraints on the sparse representation vector of the form $\{\mathbf{x} \in \mathbb{C}^N : |x_n| \in \{0, c\} \forall n \in [N]\}$. This problem emerges, e.g., in multi-user massive MIMO hybrid precoding systems with antenna selection and strict per antenna magnitude requirements [13]. In this application, let $\mathbf{A}$ denote the MIMO $N \times K$ channel matrix, $\mathbf{y}$ denote the symbol vector of the $K$ users, and $\mathbf{x}$ denote the transmitted signal vector. To limit nonlinearity effects in the power amplifiers, the magnitudes of non-zero signals $x_n$ transmitted from the selected antennas are restricted to a constant $c$. The optimization problem can be formulated as [13]

$$\min_{\mathbf{x} \in \mathbb{C}^N} \|\mathbf{x}\|_0 \tag{7.12a}$$

$$\text{s.t. } \|\mathbf{y} - \mathbf{A}^{\mathrm{T}}\mathbf{x}\|_2 \leq \sqrt{\delta}, \tag{7.12b}$$

$$|x_n| \in \{0, c\}, \quad \forall n \in [N], \tag{7.12c}$$

where $\|\mathbf{x}\|_0 = |\{n \in [N] : x_n \neq 0\}|$ denotes the number of non-zero entries of $\mathbf{x}$, i.e., the number of active antennas. We assume without loss of generality that $c = 1$. In order to reformulate the constant-modulus constraint (7.12c), we split vector $\mathbf{x}$ into real and imaginary parts $\mathrm{Re}[\mathbf{x}]$ and $\mathrm{Im}[\mathbf{x}]$, respectively. Let $\mathbf{b} = [b_1, b_2, \ldots, b_N]^{\mathrm{T}} \in \{0, 1\}^N$ denote a vector of binary variables. Problem (7.12) can then be written as

$$\min_{\mathbf{x}\in\mathbb{C}^N,\mathbf{b}\in\{0,1\}^N} \sum_{n=1}^{N} b_n \tag{7.13a}$$

$$\text{s.t.} \sum_{k=1}^{K} \Big( \mathrm{Re}[y_k] - \big( \mathrm{Re}[\mathbf{a}_k]^{\mathrm{T}}\mathbf{w} - \mathrm{Im}[\mathbf{a}_k]^{\mathrm{T}}\mathbf{z} \big) \Big)^2 + \Big( \mathrm{Im}[y_k] - \big( \mathrm{Re}[\mathbf{a}_k]^{\mathrm{T}}\mathbf{z} + \mathrm{Im}[\mathbf{a}_k]^{\mathrm{T}}\mathbf{w} \big) \Big)^2 \leq \delta, \tag{7.13b}$$

$$\mathrm{Re}[x_n]^2 + \mathrm{Im}[x_n]^2 \leq b_n, \quad \forall n \in [N], \tag{7.13c}$$

$$\mathrm{Re}[x_n]^2 + \mathrm{Im}[x_n]^2 \geq b_n, \quad \forall n \in [N], \tag{7.13d}$$

$$b_n \in \{0, 1\}, \qquad \forall n \in [N]. \tag{7.13e}$$

In (7.13), we have replaced the modulus constraints $|x_n|^2 = \mathrm{Re}[x_n]^2 + \mathrm{Im}[x_n]^2 = b_n$, $n \in [N]$, by the two inequality constraints (7.13c) and (7.13d), which will be treated differently in the following. The mixed-integer nonlinear program (7.13) will be solved by employing a spatial branching method [71] in which branching is performed on both integral and continuous variables. In this branch-and-bound procedure, the binary constraints $b_n \in \{0, 1\}$ at each node of the tree are relaxed to linear inequality constraints $0 \leq b_n \leq 1$.

In the case that the solution $(\hat{\mathbf{x}}, \hat{\mathbf{b}})$ of the LP relaxation of Problem (7.13) does not satisfy the condition $\mathrm{Re}[\hat{x}_n]^2 + \mathrm{Im}[\hat{x}_n]^2 \geq \hat{b}_n$ for some $n \in [N]$, this violation will be resolved by one of the following steps:

1. If the binary variable $\hat{b}_n$ is already fixed to zero, inequality (7.13c) also implies that $\hat{x}_n$ is set to zero.
2. If the bounds of the continuous variables $\mathrm{Re}[x_n]$ and $\mathrm{Im}[x_n]$ are not yet restricted to one of the orthants w.r.t. $\mathrm{Re}[x_n] \times \mathrm{Im}[x_n]$, four branching nodes can be created, the first with the additional constraints $\mathrm{Re}[x_n] \geq 0$, $\mathrm{Im}[x_n] \geq 0$, the second with $\mathrm{Re}[x_n] \geq 0$, $\mathrm{Im}[x_n] \leq 0$, the third with $\mathrm{Re}[x_n] \leq 0$, $\mathrm{Im}[x_n] \leq 0$, and the fourth with $\mathrm{Re}[x_n] \leq 0$, $\mathrm{Im}[x_n] \geq 0$. This partitions the feasible solution set into these four orthants.
3. If the bounds of the continuous variables $\mathrm{Re}[x_n]$ and $\mathrm{Im}[x_n]$ are already restricted to one of these four orthants, the following steps are performed. Assume w.l.o.g. that $(\hat{x}_n, \hat{b}_n)$ is feasible for the first orthant, i.e., $\mathrm{Re}[\hat{x}_n] \geq 0$ and $\mathrm{Im}[\hat{x}_n] \geq 0$.
   **Propagation:** Let $\ell_\mathrm{r} \leq \mathrm{Re}[x_n] \leq u_\mathrm{r}$, $\ell_\mathrm{i} \leq \mathrm{Im}[x_n] \leq u_\mathrm{i}$ denote the current lower and upper bounds of the variables $\mathrm{Re}[x_n]$ and $\mathrm{Im}[x_n]$, respectively. Compute the four points $(\ell_\mathrm{r}, f(\ell_\mathrm{r}))$, $(u_\mathrm{r}, f(u_\mathrm{r}))$, $(f(\ell_\mathrm{i}), \ell_\mathrm{i})$, and $(f(u_\mathrm{i}), u_\mathrm{i})$ on the unit circle that correspond to the respective lower and upper bounds of $\mathrm{Re}[x_n]$ and $\mathrm{Im}[x_n]$, where $f(x) = \sqrt{1 - x^2}$. These four points can now be used to strengthen the lower and upper bounds of $\mathrm{Re}[x_n]$ and $\mathrm{Im}[x_n]$, as depicted in the left-hand side of Fig. 7.2. If the binary variable $b_n$ is not yet fixed to one, the lower bounds are not propagated, as $b_n$ could be set to zero in an optimal solution, implying $\mathrm{Re}[x_n] = \mathrm{Im}[x_n] = 0$ as well. As an example, consider the situation depicted in

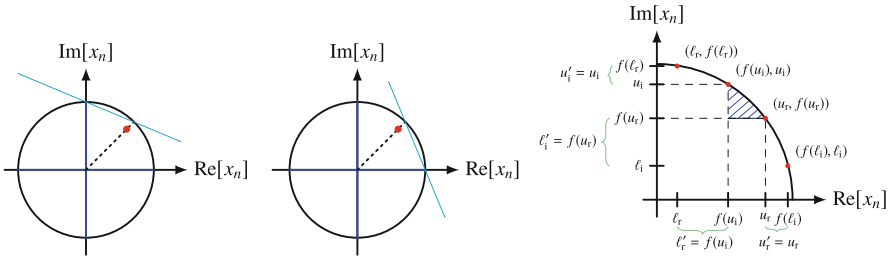



**Fig. 7.2** Left: Inequalities that are added to the sub-nodes. Right: Bound propagation for the continuous variables

Fig. 7.2. In order for an optimal solution $(\mathbf{x}^\star, \mathbf{b}^\star)$ to fulfill the modulus constraint $\mathrm{Re}[x_n]^2 + \mathrm{Im}[x_n]^2 \geq b_n$, the point $(\mathrm{Re}[x_n^\star], \mathrm{Im}[x_n^\star])$ needs to lie on or above the arc between the two points $(f(u_\mathrm{i}), u_\mathrm{i})$ and $(u_\mathrm{r}, f(u_\mathrm{r}))$ if $b_n^\star = 1$, so that the lower bounds of $\mathrm{Re}[x_n]$ and $\mathrm{Im}[x_n]$ can be tightened.

**Separation:** If $\mathrm{Re}[\hat{x}_n] + \mathrm{Im}[\hat{x}_n] < \hat{b}_n$, add the cut $\mathrm{Re}[x_n] + \mathrm{Im}[x_n] \geq b_n$ to the LP relaxation. Note that each solution in this orthant on the unit circle satisfies this inequality.

**Branching:** If $\mathrm{Re}[\hat{x}_n] + \mathrm{Im}[\hat{x}_n] \geq \hat{b}_n$, create two branching nodes defined by linear inequalities of the form $f_n\ \mathrm{Re}[x_n] + g_n\ \mathrm{Im}[x_n] \geq b_n$, as visualized in the left-hand side of Fig. 7.2.

Computationally efficient suboptimal heuristic solutions for problem (7.12) and simulation results from numerical experiments can further be found in [13].

### *7.4.2 Row- and Rank-Sparsity*

In this section, we consider row- and rank-sparse recovery from noisy measurements. The idea to exploit a common sparsity structure among multiple measurements as prior information was proposed in [23, 29, 41, 67, 69, 85], where the mixed-norm (7.6) is used to enforce row-sparsity. The corresponding row-sparse data model is illustrated in Fig. 7.3. The classical row-sparse recovery problem corresponds to a least-squares data-fitting problem with $\ell_{2,1}$-mixed-norm minimization:

$$\min_{\mathbf{X}} \frac{1}{2} \|\mathbf{A}\mathbf{X} - \mathbf{Y}\|_\mathrm{F}^2 + \lambda \sqrt{D}\, \|\mathbf{X}\|_{2,1}, \tag{7.14}$$

where $\mathbf{X} = [\mathbf{x}(1), \ldots, \mathbf{x}(D)]$. This problem emerges, e.g., in the context of direction-of-arrival (DoA) estimation, where the columns of the dictionary $\mathbf{A}$ represent the array responses for difference directions and the support of the matrix $\mathbf{X}$, i.e., the indices of the non-zero rows indicate the source DoAs. The

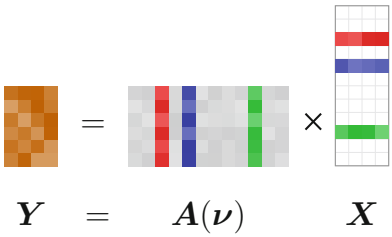



**Fig. 7.3** Multiple measurement problem with row sparsity

dimension of problem (7.14) grows with the number of measurements $D$ and the size of the dictionary and can become computationally intractable. To reduce the computational cost, it was suggested in [41] to reduce the dimension of the $M \times D$ measurement matrix $\mathbf{Y}$ by matching only the signal subspace in the form of an $M \times K$ matrix $\mathbf{Y}_{\mathrm{SV}}$, leading to the prominent $\ell_1$-SVD method. A drawback of the $\ell_1$-SVD method is that it requires knowledge of the number of source signals and that the estimation performance may deteriorate in the case of correlated source signals. To overcome this limitation, a convenient equivalent problem reformulation was derived in [54] as stated in the following theorem.

**Theorem 7.3 (Problem Equivalence 1)** *The row-sparsity inducing $\ell_{2,1}$ mixed norm minimization problem* (7.14) *is equivalent to the convex problem*

$$\min_{\mathbf{S}\in\mathbb{D}_+} \mathrm{tr}\big((\mathbf{A}\mathbf{S}\mathbf{A}^{\mathrm{H}} + \lambda\,\mathbf{I}_M)^{-1}\hat{\mathbf{R}}\big) + \mathrm{tr}(\mathbf{S}), \tag{7.15}$$

*with $\hat{\mathbf{R}} = \mathbf{Y}\mathbf{Y}^{\mathrm{H}}/D$ denoting the sample covariance matrix and $\mathbb{D}_+$ describing the set of nonnegative diagonal matrices, in the sense that minimizers $\mathbf{X}^\star$ and $\mathbf{S}^\star$ for problems* (7.14) *and* (7.15)*, respectively, are related by*

$$\mathbf{X}^\star = \mathbf{S}^\star\mathbf{A}^{\mathrm{H}}(\mathbf{A}\mathbf{S}^\star\mathbf{A}^{\mathrm{H}} + \lambda\mathbf{I}_M)^{-1}\mathbf{Y}. \tag{7.16}$$

*Conversely, $\mathbf{S}^\star = \mathrm{diag}(s_1^\star, \ldots, s_K^\star)$ contains the row-norms of the sparse signal matrix $\mathbf{X}^\star = [\mathbf{x}_1^\star, \ldots, \mathbf{x}_K^\star]^{\mathrm{T}}$ on its diagonal according to*

$$s_k^\star = \frac{1}{\sqrt{D}}\|\mathbf{x}_l^\star\|_2, \tag{7.17}$$

*for $k = 1, \ldots, K$, such that the union support of $\mathbf{X}^\star$ is equivalently represented by the support of the sparse vector of row norms $[s_1^\star, \ldots, s_K^\star]^{\mathrm{T}}$.*

Problem (7.15) is known as the SPARse ROW-norm reconstruction (SPARROW) reformulation. It reveals several interesting properties of the underlying multiple measurement problem, and it can be reformulated as a semidefinite program. Unlike Problem (7.14), the dimension of (7.15) does not grow with the number of measurements [54]. Gridless variants of the method for uniform linear arrays (ULAs), shift-invariant arrays, and augmentable arrays are reported in [3, 53, 54, 64, 72].

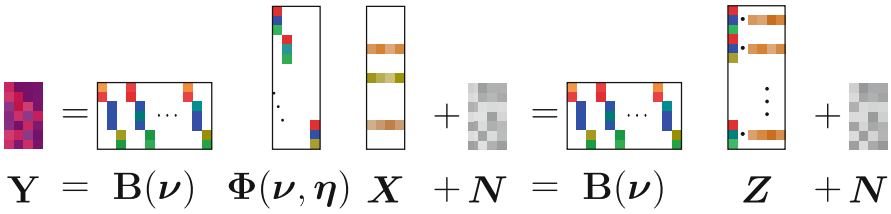



**Fig. 7.4** Multiple measurement problem with block sparsity

In the case of DoA estimation in partly calibrated subarray systems with unknown DoAs $\boldsymbol{\nu}$ and subarray position parameters $\boldsymbol{\eta}$, the recovery problem can be formulated as a rank- and block-sparse regularization problem [50]. The corresponding data model is illustrated in Fig. 7.4, where $\mathbf{B}(\boldsymbol{\nu})$ contains the subarray steering vectors, $\boldsymbol{\Phi}(\boldsymbol{\nu}, \boldsymbol{\eta}) = [\boldsymbol{\varphi}(\nu_1, \boldsymbol{\eta}), \ldots, \boldsymbol{\varphi}(\nu_K, \boldsymbol{\eta})]$ contains the inter-subarray array responses, and $\mathbf{X}$ contains the row-sparse signal waveforms. We observe that the matrix $\mathbf{Z} = [\mathbf{Z}_1^{\mathrm{T}}, \ldots, \mathbf{Z}_K^{\mathrm{T}}]^{\mathrm{T}}$ enjoys a special block- and rank-sparse structure as it is composed of $K$-stacked rank-one matrices $\mathbf{Z}_k = \boldsymbol{\varphi}(\nu_k, \boldsymbol{\eta})\, \mathbf{x}_k^{\mathrm{T}}$, for $k = 1, \ldots, K$. The block- and rank-sparse recovery problem is given by

$$\min_{\mathbf{Z}} \frac{1}{2} \|\mathbf{B}\mathbf{Z} - \mathbf{Y}\|_{\mathrm{F}}^2 + \sum_{k=1}^{K} \|\mathbf{Z}_k\|_*, \tag{7.18}$$

where the nuclear norm regularization $\|\mathbf{Z}_k\|_* = \mathrm{tr}\big((\mathbf{Z}_k^{\mathrm{H}}\mathbf{Z}_k)^{1/2}\big)$ encourages block-rank-sparsity, i.e., the solution blocks $\mathbf{Z}_k$ shall either be zero or low-rank [32, 33, 52, 77]. Similar to Problem (7.14) also Problem (7.18) admits a convenient reformulation with a significantly reduced number of optimization variables, as provided by the following theorem [50, 51].

**Theorem 7.4 (Problem Equivalence 2)** *The rank- and block-sparsity inducing $\ell_{*,1}$-mixed-norm minimization Problem (7.18) is equivalent to the convex problem*

$$\min_{\mathbf{S} \in \mathcal{S}_K^+} \mathrm{tr}\big((\mathbf{B}\mathbf{S}\mathbf{B}^{\mathrm{H}} + \lambda \mathbf{I})^{-1} \hat{\mathbf{R}}\big) + \mathrm{tr}(\mathbf{S}), \tag{7.19}$$

*with $\hat{\mathbf{R}} = \mathbf{Y}\mathbf{Y}^{\mathrm{H}}/D$ and $\mathcal{S}_K^+$ denoting the sample covariance matrix and the set of positive semidefinite block-diagonal matrices composed of $K$ blocks of size $P \times P$, respectively. The equivalence holds in the sense that a minimizer $\mathbf{Z}^\star$ for Problem (7.18) can be factorized as*

$$\mathbf{Z}^\star = \mathbf{S}^\star \mathbf{B}^{\mathrm{H}} (\mathbf{B}\mathbf{S}^\star\mathbf{B}^{\mathrm{H}} + \lambda \mathbf{I})^{-1} \mathbf{Y}, \tag{7.20}$$

*where $\mathbf{S}^\star$ is a minimizer for Problem (7.19).*

### 7.4.3 Block-Sparse Tensors

In [7], block-sparse core tensors were considered as the natural multidimensional extension of block-sparse vectors or matrices (as illustrated in Fig. 7.4) in the context of multidimensional data acquisition. Such block-sparse tensors arise naturally in a wide range of applications as, for instance, in magnetic resonance imaging (MRI), hyper-spectral imaging, multidimensional inpainting, missing data problems for electroencephalogram (EEG), super-resolution imaging, or MIMO wireless communications. The $(M_1, \ldots, M_Q)$ block sparsity for a tensor assumes that $Q$ support sets, characterized by $M_q$ indices corresponding to the non-zero entries, fully describe the sparsity pattern of the considered tensor. In the context of compressed sensing with Gaussian measurement matrices, the Cramér-Rao bound (CRB) on the estimation accuracy of a Bernoulli-distributed block-sparse core tensor was also derived in [7]. This prior assumes that each entry of the core tensor has a given probability to be non-zero, leading to random supports of truncated Binomial-distributed cardinalities. Based on the limit form of the Poisson distribution, an approximated CRB expression is provided for large dictionaries and a highly block-sparse core tensor. Using the property that the $n$-mode unfoldings of a block-sparse tensor follow the multiple-measurement vectors (MMV) model with a joint sparsity pattern, a fast and accurate estimation scheme, called Beamformed mOde-based Sparse Estimator (BOSE), is proposed in the second part of [7]. The main contribution of BOSE is to exploit the structure by mapping the MMV model onto the single-measurement vector (SMV) model, via beamforming techniques. Finally, the proposed performance bounds and BOSE are applied in the context of compressed sensing to non-bandlimited multidimensional signals with separable sampling kernels and for multipath channels in a MIMO wireless communication scheme.

### 7.4.4 Non-circularity

Recently, three different sparse recovery strategies have been proposed [55, 57, 58] for exploiting the strict non-circularity property of the impinging signals $\mathbf{x}^{(0)}(t)$ in (7.1) [56, 61], i.e., the received complex symbols $\mathbf{x}^{(0)}(t)$ result from real-valued constellations rotated by an arbitrary phase $\phi$. These strictly non-circular signals may represent real-valued modulation schemes such as BPSK (binary phase shift keying), PAM (pulse amplitude modulation), ASK (amplitude shift keying), or Offset-QPSK (offset-quadrature phase shift keying, after an appropriate derotation). As the rotation phase $\phi$ (that may be due to the propagation environment) is usually unknown, the estimation problem becomes a two-dimensional (2-D) sparse recovery problem, which requires estimating the support in the spatial domain as well as in the rotation phase domain.

In [55], a combined 2-D finite dictionary was introduced for both dimensions, and the resulting 2-D sparse recovery problem was solved by a $\ell_{2,1}$-mixed-norm relaxation using multiple-measurement vectors (MMV). Thereby, the known benefits associated with strictly non-circular (NC) sources [56, 61], e.g., an improved estimation accuracy and a doubling of the number of resolvable signals, can also be achieved via sparse recovery. In order to handle the resulting 2-D off-grid problem, an off-grid estimation procedure was introduced by means of local interpolation.

Article [58] addresses the prohibitive computational complexity required for solving the 2-D mixed-norm problem as a result of sampling both dimensions, significantly increasing the size. Thus, in [58], a sparse optimization framework was proposed based on nuclear norm (rank) minimization after lifting the original optimization problem to a semidefinite programming (SDP) problem in a higher-dimensional space. To this end, the 2-D estimation problem is reduced to a 1-D estimation problem only in the sampled spatial domain, which automatically provides gridless estimates of the rotation phases. As a result, the proposed method requires a significantly lower computational complexity while providing the same performance benefits. Additionally, an off-grid estimator for the spatial domain has been proposed.

In [57], a gridless sparse recovery algorithm for NC signals has been proposed based on atomic norm minimization (ANM). After the NC preprocessing step, the ANM-equivalent SDP problem provides a solution matrix with a two-level Hermitian Toeplitz structure. It was shown that by using the multidimensional generalization of the Vandermonde decomposition, the desired direction estimates can be uniquely extracted from the two-level Hermitian Toeplitz matrix via NC Standard ESPRIT or NC Unitary ESPRIT [17] in closed form. Due to the exploitation of the NC signal structure, the proposed NC ANM procedure provides a superior estimation accuracy as compared to the original methods for arbitrary signals. In this case, the number of estimated sources can exceed the number of sensors in the array.

## 7.5 Mixing Matrix Design

In this section, we consider a noiseless time-invariant version of (7.2) given as

$$\mathbf{y} = \boldsymbol{\Phi}\mathbf{A}\mathbf{x} = \boldsymbol{\Psi}\mathbf{x} \in \mathbb{C}^N, \tag{7.21}$$

where $\boldsymbol{\Psi} = \boldsymbol{\Phi}\mathbf{A} \in \mathbb{C}^{N\times P}$ is the total sensing matrix, $\boldsymbol{\Phi} \in \mathbb{C}^{N\times M}$ is the mixing matrix (a.k.a., the projection/compression matrix), $\mathbf{A} \in \mathbb{C}^{M\times P}$ is the dictionary matrix with $P \geq M$, and $\mathbf{x} \in \mathbb{C}^P$ is the signal vector of interest with $\|\mathbf{x}\|_0 \leq s$, i.e., $\mathbf{x}$ is $s$-sparse. To enhance recoverability of $\mathbf{x}$, the sensing matrix $\boldsymbol{\Psi}$ should be designed carefully so that it satisfies a certain property (e.g., the NSP or the restricted isometry property (RIP) [8, 9]). Among them, the mutual coherence property of the sensing

matrix $\boldsymbol{\Psi}$, denoted hereafter as $\mu_{\max}(\boldsymbol{\Psi})$, provides an easy measure with respect to recoverability, which is defined as [48]

$$\mu_{\max}(\boldsymbol{\Psi}) = \max_{i \neq j} \frac{|\boldsymbol{\psi}_i^{\mathrm{H}} \boldsymbol{\psi}_j|}{\|\boldsymbol{\psi}_i\|_2 \, \|\boldsymbol{\psi}_j\|_2}, \tag{7.22}$$

with columns $\boldsymbol{\psi}_k = [\psi_{k,1}, \ldots, \psi_{k,N}]^{\mathrm{T}} \in \mathbb{C}^N$, $k \in \{1, \ldots, P\}$. Clearly, a large coherence $\mu_{\max}(\boldsymbol{\Psi})$ means that there exist, at least, two highly correlated columns in $\boldsymbol{\Psi}$, which may confuse any pursuit technique, such as basis pursuit (BP) and orthogonal matching pursuit (OMP). However, it has been shown that if $s < \frac{1}{2}\big(1 + 1/\mu_{\max}(\boldsymbol{\Psi})\big)$, the above techniques are guaranteed to recover $\mathbf{x}$ with high probability [8, 48]. Due to its simplicity, several sensing matrix design methods via mutual coherence minimization have been proposed recently, e.g., in [1, 84, 86]. In general, the results provided by [1, 84, 86] confirm that a well-designed sensing matrix always leads to a better recoverability. However, we note that the achievable mutual coherence by the aforementioned methods is, in general, far from the known theoretical Welch lower bound, as we will also show in Sect. 7.5.3. Moreover, in the scenarios where the mixing matrix is realized using a network of phase shifters, none of the existing methods, to the best of our knowledge, have considered the constant-modulus constraints imposed by the mixing matrix hardware that involves cost-efficient analog phase shifters.

Formally, by assuming that the dictionary matrix $\mathbf{A} \in \mathbb{C}^{M \times P}$ is given and fixed, sensing matrix design reduces to finding the mixing matrix $\boldsymbol{\Phi}$ with constant-modulus entries so that the coherence $\mu_{\max}(\boldsymbol{\Psi})$ is minimized, which can be expressed as

$$\min_{\boldsymbol{\Phi} \in \mathbb{C}^{N \times M}} \mu_{\max}(\boldsymbol{\Psi}) \quad \text{s.t.} \quad \|\boldsymbol{\psi}_k\|_2 = 1, \ \forall k, \ \text{and} \ |\phi_{n,m}| = 1, \ \forall n, m, \tag{7.23}$$

where $\phi_{n,m}$ is the $(n, m)$-th entry of $\boldsymbol{\Phi}$, $n \in \{1, \ldots, N\}$, and $m \in \{1, \ldots, M\}$. Problem (7.23) is a nonconvex and $\mathcal{NP}$-hard optimization problem [40]. In the following, we propose two solution methods. Section 7.5.1 presents the sequential mutual coherence minimization (SMCM) we proposed in [4] for the case of $P = M$. In Sect. 7.5.2, we propose a new method termed enhanced gradient descent (EGD) for the more general case of $P \geq M$.

### 7.5.1 Sensing Matrix Design: $P = M$ Case

In this subsection, we present our first solution to problem (7.23) for unconstrained mixing matrix design, i.e., by neglecting the constant-modulus constraints. Specifically, for a given dictionary matrix $\mathbf{A} \in \mathbb{C}^{M \times P}$, we assume that $P = M$ and the columns of $\mathbf{A}$ are linearly independent so that the condition of $\mathbf{A}\mathbf{A}^{-1} = \mathbf{I}_M$ is guaranteed. In this case, for a given sensing matrix $\boldsymbol{\Psi} \in \mathbb{C}^{N \times P}$ with a coherence

$\mu_{\max} = \mu_{\max}(\boldsymbol{\Psi})$, the optimal unconstrained mixing matrix that preserves $\mu_{\max}$ can be obtained as $\boldsymbol{\Phi}_{\text{uncon}} = \boldsymbol{\Psi}\mathbf{A}^{-1} \in \mathbb{C}^{N\times M}$, i.e., $\mu_{\max}(\boldsymbol{\Phi}_{\text{uncon}}\mathbf{A}) = \mu_{\max}$. Therefore, the main task here is to find a low-coherence sensing matrix $\boldsymbol{\Psi} \in \mathbb{C}^{N\times P}$.

Let us assume that the columns of $\boldsymbol{\Psi}$ are normalized so that $\|\boldsymbol{\psi}_k\|_2 = 1, \forall k$, and let $\mathbf{G} = \boldsymbol{\Psi}^{\mathrm{H}}\boldsymbol{\Psi} \in \mathbb{C}^{P\times P}$ be the so-called Gram matrix of $\boldsymbol{\Psi}$. Moreover, let $\mathbf{G}_{\text{sqr-abs}} \in \mathbb{R}^{P\times P}$ be a matrix so that its $(k, j)$-th entry is given as $\mathbf{G}^{[k,j]}_{\text{sqr-abs}} = |\mathbf{G}^{[k,j]}|^2$. By expanding $\mathbf{G}_{\text{sqr-abs}}$, it can be expressed as

$$\mathbf{G}_{\text{sqr-abs}} = \begin{bmatrix} |\boldsymbol{\psi}_1^{\mathrm{H}}\boldsymbol{\psi}_1|^2 & \dots & |\boldsymbol{\psi}_1^{\mathrm{H}}\boldsymbol{\psi}_P|^2 \\ \vdots & \ddots & \vdots \\ |\boldsymbol{\psi}_P^{\mathrm{H}}\boldsymbol{\psi}_1|^2 & \dots & |\boldsymbol{\psi}_P^{\mathrm{H}}\boldsymbol{\psi}_P|^2 \end{bmatrix} = \begin{bmatrix} 1 & \dots & |\boldsymbol{\psi}_1^{\mathrm{H}}\boldsymbol{\psi}_P|^2 \\ \vdots & \ddots & \vdots \\ |\boldsymbol{\psi}_P^{\mathrm{H}}\boldsymbol{\psi}_1|^2 & \dots & 1 \end{bmatrix}, \tag{7.24}$$

which is a symmetric matrix with all ones on its main diagonal. Since all vectors in $\boldsymbol{\Psi}$ have unit norm, we have $\mathbf{G}^{[k,j]}_{\text{sqr-abs}} = |\boldsymbol{\psi}_k^{\mathrm{H}}\boldsymbol{\psi}_j|^2 \leq 1, \forall k \neq j$, and the maximum among them represents the squared coherence of the matrix $\boldsymbol{\Psi}$. According to [36, 68], $\mu_{\max}(\boldsymbol{\Psi})$ has a theoretical lower bound given as $\mu_{\max}(\boldsymbol{\Psi}) \geq \sqrt{\beta}$, where $\beta = \frac{P-N}{N(P-1)}$. This means that, at the best, we have $\mu_{\max}(\boldsymbol{\Psi}) = \sqrt{\beta}$. Noting that the $k$-th column vector $\boldsymbol{\psi}_k$ appears only in the $k$-th column/row of $\mathbf{G}_{\text{sqr-abs}}$ (due to its symmetry), we propose to solve problem (7.23) in an alternating fashion by iterating over the following $P$ subproblems, where the $k$-th subproblem for updating $\boldsymbol{\psi}_k$ is given as

$$\text{find} \quad \boldsymbol{\psi}_k \in \mathbb{C}^N \quad \text{s.t.} \quad |\boldsymbol{\psi}_j^{\mathrm{H}}\boldsymbol{\psi}_k|^2 \leq \beta \ \forall j \neq k, \ \text{and} \ \|\boldsymbol{\psi}_k\|_2 = 1. \tag{7.25}$$

Problems (7.23) and (7.25) are related in the sense that both aim to minimize the maximum off-diagonal entry in (7.24). However, the strict unit-norm constraint $\|\boldsymbol{\psi}_k\|_2 = 1$ in Problem (7.25) may result in infeasibility for poorly initialized vectors $\boldsymbol{\psi}_j, \forall j \neq k$, especially with a tight lower bound $\beta$. To avoid such a scenario, we propose to relax (7.25) by dropping the unit-norm constraint and only impose it after a solution is obtained, i.e., we first seek a solution to the following relaxed problem

$$\text{find} \quad \boldsymbol{\psi}_k \in \mathbb{C}^N \quad \text{s.t.} \quad |\boldsymbol{\psi}_j^{\mathrm{H}}\boldsymbol{\psi}_k|^2 \leq \beta \ \forall j \neq k, \tag{7.26}$$

which, unlike (7.25), is guaranteed to be feasible. To obtain a solution of problem (7.26), a suitable objective function is needed. One possible approach is as follows

$$\boldsymbol{\psi}_k \in \underset{\mathbf{v}_k\in\mathbb{C}^N}{\arg\max} \ |\boldsymbol{\psi}_k^{\mathrm{H}}\mathbf{v}_k|^2 \quad \text{s.t.} \quad |\boldsymbol{\psi}_j^{\mathrm{H}}\mathbf{v}_k|^2 \leq \beta \ \forall j \neq k. \tag{7.27}$$

In problem (7.27), we borrow the notion from the beamforming design in wireless communication systems, see Fig. 7.5, where we interpret $\mathbf{v}_k \in \mathbb{C}^N$ as the beamforming vector of the $k$-th mobile station (MS) that we wish to design

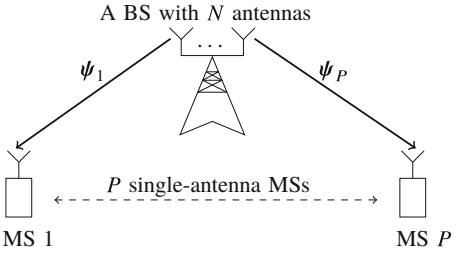



**Fig. 7.5** A $P$-user interference-channel (IC) system model in wireless communication systems, where a base station (BS) with $N$ antennas serves $P$ single-antenna mobile stations (MSs) simultaneously so that the data transmission to the $k$-th MS causes interference to the remaining $P-1$ MSs

so that the desired transmit signal to the $k$-th MS, i.e., $|\boldsymbol{\psi}_k^{\mathrm{H}}\mathbf{v}_k|^2$, is maximized and the interference signals to the remaining $P-1$ MSs, i.e., $|\boldsymbol{\psi}_j^{\mathrm{H}}\mathbf{v}_k|^2 \leq \beta, \forall j \neq k$, are minimized for given channel vectors $\{\boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_P\}$. Due to its convexity, Problem (7.27) can be efficiently solved using the existing techniques, e.g., using the proposed method in [46], as we have shown in [4, 5]. Alternatively, we can resort to the relaxed semidefinite programming (SDP) approach, by dropping the rank-one constraint, and write Problem (7.27) as

$$\max_{\mathbf{V}_k \in \mathbb{C}^{N\times N}} \ \mathrm{tr}\{\boldsymbol{\Psi}_k^{\mathrm{cov}}\mathbf{V}_k\} \quad \text{s.t.} \quad \mathrm{tr}\{\boldsymbol{\Psi}_j^{\mathrm{cov}}\mathbf{V}_k\} \leq \beta \ \forall j \neq k, \ \text{and} \ \mathbf{V}_k \succeq 0, \tag{7.28}$$

where $\boldsymbol{\Psi}_k^{\mathrm{cov}} = \boldsymbol{\psi}_k\boldsymbol{\psi}_k^{\mathrm{H}} \in \mathbb{C}^{N\times N}$ and $\mathbf{V}_k = \mathbf{v}_k\mathbf{v}_k^{\mathrm{H}} \in \mathbb{C}^{N\times N}$. Problem (7.28) is convex and can be efficiently solved using off-the-shelf solvers, e.g., the CVX toolbox. Let $\mathbf{V}_k$ denote the obtained solution of (7.28). Then, $\boldsymbol{\psi}_k$ is given by the eigenvector corresponding to the dominant eigenvalue of $\mathbf{V}_k$, i.e., $\boldsymbol{\psi}_k = \lambda_{\max}\{\mathbf{V}_k\}$. In summary, the proposed mixing matrix design method is given by Algorithm 1. Note that a naïve approach to obtain a constrained mixing matrix, i.e., one with constant-modulus entries, is given as $\boldsymbol{\Phi}_{\mathrm{con}} = \boldsymbol{\Pi}(\boldsymbol{\Phi}_{\mathrm{uncon}})$, where $\boldsymbol{\Pi}(\cdot)$ is a projection function that imposes the constant-modulus constraints on $\boldsymbol{\Phi}_{\mathrm{uncon}}$ element-wise, i.e., $\boldsymbol{\Pi}(z) = z/|z|$. The performance of such an approach will also be evaluated in Sect. 7.5.3.

### *7.5.2 Sensing Matrix Design: The General Case*

In this subsection, we propose a new solution to (7.23) for the more general case of $P \geq M$. Similarly to [1], we propose to solve (7.23) indirectly by solving

$$\min_{\boldsymbol{\Phi}\in\mathbb{C}^{N\times M}} \ \eta(\boldsymbol{\Phi}) \quad \text{s.t.} \quad \|\boldsymbol{\psi}_k\|_2 = 1, \ \forall k, \ \text{and} \ |\phi_{n,m}| = 1, \ \forall n, m, \tag{7.29}$$

**Algorithm 1:** Sequential mutual coherence minimization (SMCM)

**input** : $\mathbf{\Psi}_{(0)} \in \mathbb{C}^{N\times P}$, $\epsilon_{\text{th}}$
**initialize**: $\beta = \frac{P-N}{N(P-1)}$ and $n = 1$
**for** $n = 1, 2, \ldots$ **do**
  **for** $k = 1$ *to* $P$ **do**
    Compute $\mathbf{V}_{k(n)}$ by solving problem (7.28).
    Update the $k$-th column vector of $\mathbf{\Psi}_{(n)}$ as $\boldsymbol{\psi}_{k(t)} = \lambda_{\max}\{\mathbf{V}_{k(n)}\}$.
  **end**
  **if** $\epsilon = |\mu_{\max}(\mathbf{\Psi}_{(n)}) - \mu_{\max}(\mathbf{\Psi}_{(n-1)})|^2 \leq \epsilon_{th}$ **then**
    Break
  **end**
**end**
**output** : the sensing matrix $\mathbf{\Psi}$ and the corresponding mixing matrix $\mathbf{\Phi}_{\text{uncon}} = \mathbf{\Psi}\mathbf{A}^{-1}$.

where $\eta(\mathbf{\Phi}) = \|\mathbf{A}^{\text{H}}\mathbf{\Phi}^{\text{H}}\mathbf{\Phi}\mathbf{A} - \mathbf{I}_P\|_{\text{F}}^2$. To obtain a solution for (7.29), we propose a constrained gradient-descent (GD) method, which updates the mixing matrix $\mathbf{\Phi}$ iteratively as

$$\mathbf{\Phi}_{(n)} = \mathbf{\Pi}\left(\mathbf{\Phi}_{(n-1)} - \zeta \cdot \frac{\partial \eta(\mathbf{\Phi}_{(n-1)})}{\partial \mathbf{\Phi}_{(n-1)}}\right), \tag{7.30}$$

where $n$ is the iteration index, $\zeta$ is the step size, and $\frac{\partial \eta(\mathbf{\Phi}_{(n-1)})}{\partial \mathbf{\Phi}_{(n-1)}}$ is the gradient of $\eta(\mathbf{\Phi}_{(n-1)})$ with respect to $\mathbf{\Phi}_{(n-1)}$, which is given as [1]

$$\frac{\partial \eta(\mathbf{\Phi}_{(n-1)})}{\partial \mathbf{\Phi}_{(n-1)}} = \mathbf{\Phi}_{(n-1)}\mathbf{A}(\mathbf{A}^{\text{H}}\mathbf{\Phi}_{(n-1)}^{\text{H}}\mathbf{\Phi}_{(n-1)}\mathbf{A} - \mathbf{I}_P)\mathbf{A}^{\text{H}} = \mathbf{\Psi}_{(n-1)}\mathbf{E}_{(n-1)}\mathbf{A}^{\text{H}}, \tag{7.31}$$

where $\mathbf{\Psi}_{(n-1)} = \mathbf{\Phi}_{(n-1)}\mathbf{A}$ and $\mathbf{E}_{(n-1)} = \mathbf{\Psi}_{(n-1)}^{\text{H}}\mathbf{\Psi}_{(n-1)} - \mathbf{I}_P$. The update step in (7.30) is a direct extension of the proposed unconstrained GD method in [1] to account for the constant-modulus constraints. Our results show that both the unconstrained and the constrained GD-based methods achieve a mutual coherence that is far from the known theoretical Welch lower bound, as it is shown in Table 7.1. To enhance their performance, we propose to apply a shrinking operator on the error matrix $\mathbf{E}_{(n-1)}$ entry-wise to get $\tilde{\mathbf{E}}_{(n-1)}$ such that the $(k, j)$-th entry of $\tilde{\mathbf{E}}_{(n-1)}$ is obtained as

$$\tilde{\mathbf{E}}_{(n-1)}^{[k,j]} = \begin{cases} 0, & \left|\mathbf{E}_{(n-1)}^{[k,j]}\right| < \alpha \cdot \sqrt{\beta}, \\ \text{sgn}\{\mathbf{E}_{(n-1)}^{[k,j]}\} \cdot \left(\left|\mathbf{E}_{(n-1)}^{[k,j]}\right| - \alpha \cdot \sqrt{\beta}\right), & \text{otherwise}, \end{cases} \tag{7.32}$$

where $\alpha \geq 1$ is an uncertainty measure and $\beta$ is as defined above. After a closer look at (7.32), one can see that for a very tight threshold $\bar{\beta} = \alpha \cdot \sqrt{\beta}$, the resulting error matrix $\tilde{\mathbf{E}}_{(n-1)}$ becomes a sparse matrix, where some of its entries that are smaller

**Algorithm 2:** Enhanced gradient descent (EGD)

**input** : $\mathbf{\Phi}_{(0)} \in \mathbb{C}^{N\times M}$, $\mathbf{A} \in \mathbb{C}^{M\times P}$, $\epsilon_{\text{th}}$, and $\zeta$.
**initialize**: $\beta = \frac{P-N}{N(P-1)}$ and $n = 1$.
Normalize the columns of $\mathbf{\Psi}_{(0)} = \mathbf{\Phi}_{(0)}\mathbf{A}$ so that $\|\boldsymbol{\psi}_{(0),k}\|_2 = 1, \forall k$.
**for** $n = 1, 2, \ldots$ **do**
  Calculate the error matrix $\mathbf{E}_{(n-1)} = \mathbf{\Psi}^{\text{H}}_{(n-1)}\mathbf{\Psi}_{(n-1)} - \mathbf{I}_P$.
  Apply the shrinking operator (7.32) on $\mathbf{E}_{(n-1)}$ to get $\tilde{\mathbf{E}}_{(n-1)}$.
  **if** *mixing matrix should be unconstrained (i.e., $\mathbf{\Phi}_{uncon}$)* **then**
    Compute $\mathbf{\Phi}_{(n)} = \mathbf{\Phi}_{(n-1)} - \zeta \cdot \mathbf{\Psi}_{(n-1)}\tilde{\mathbf{E}}_{(n-1)}\mathbf{A}^{\text{H}}$.
  **else if** *mixing matrix should be constrained (i.e., $\mathbf{\Phi}_{con}$)* **then**
    Compute $\mathbf{\Phi}_{(n)} = \mathbf{\Pi}\big(\mathbf{\Phi}_{(n-1)} - \zeta \cdot \mathbf{\Psi}_{(n-1)}\tilde{\mathbf{E}}_{(n-1)}\mathbf{A}^{\text{H}}\big)$.
  **end**
  Normalize the columns of $\mathbf{\Psi}_{(n)} = \mathbf{\Phi}_{(n)}\mathbf{A}$ so that $\|\boldsymbol{\psi}_{(n),k}\|_2 = 1, \forall k$.
  **if** $\epsilon = |\mu(\mathbf{\Psi}_{(n)}) - \mu(\mathbf{\Psi}_{(n-1)})|^2 \leq \epsilon_{th}$ **then**
    Break
  **end**
**end**
**output** : Mixing matrix $\mathbf{\Phi}^{\star}$

than $\bar{\beta}$ will be set to zero. The direct implication of such a shrinking operator is that the new mixing matrix $\mathbf{\Phi}_{(n)}$ will be updated so that it mainly minimizes the entries that are larger than $\bar{\beta}$. In summary, the proposed enhanced GD (EGD) method for mixing matrix design is given by Algorithm 2. In Sect. 7.5.3, we will investigate in detail the impact of $\alpha$ on the performance of EGD method.

### *7.5.3 Numerical Results*

In this subsection, we present some numerical results for the proposed sensing matrix design methods. In all the simulation results, we set $N = 16$, $M = 64$, and design the dictionary matrix as $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_P] \in \mathbb{C}^{M\times P}$ such that its $k$-th column is given as $\mathbf{a}_k = [1, e^{j\nu_k}, \ldots, e^{j\nu_k(M-1)}]^{\text{T}} \in \mathbb{C}^M$, where $\nu_k = (2\pi(k-1))/P$. For comparison, we include results for a mixing matrix $\mathbf{\Phi}$ obtained by using the proposed closed-form method in [86],[1] the proposed methods in [84] and [1], as well as averaged over 10,000 random realizations, where the entries of $\mathbf{\Phi}$ are chosen from a zero-mean circularly symmetric complex Gaussian distribution, termed EVD, Itr-SVD, GD, and Random, respectively. We show the simulation results in terms of the maximum mutual coherence $\mu_{\max}(\mathbf{\Psi})$ defined in (7.22) and the average mutual coherence $\mu_{\text{avg}}(\mathbf{\Psi})$ defined as

[1] Let $\mathbf{U}\mathbf{\Lambda}\mathbf{U}^{\text{H}}$ be the eigenvalue decomposition of $\mathbf{A}^{\text{H}}\mathbf{A}$. Then, the unconstrained mixing matrix is obtained as $\mathbf{\Phi}_{\text{uncon}} = \mathbf{\Lambda}_N^{-1/2}\mathbf{U}_N^{\text{H}}$, where $\mathbf{\Lambda}_N$ and $\mathbf{U}_N$ contain the leading $N$ eigenvalues and eigenvectors, respectively. For constrained mixing matrix scenarios, simply $\mathbf{\Phi}_{\text{con}} = \mathbf{\Pi}(\mathbf{\Phi}_{\text{uncon}})$.

**Table 7.1** Coherence $\mu_{\max}(\mathbf{\Psi})$ ($\mu_{\text{avg}}(\mathbf{\Psi})$) versus $P$ ($N = 16$ and $M = 64$)

| | $P$ | Random | EVD | Itr-SVD | GD | EGD | SMCM |
|---|---|---|---|---|---|---|---|
| $\mathbf{\Phi}_{\text{uncon}}$ | 64 | 0.64 (0.32) | 0.56 (0.30) | **0.24** (0.23) | 0.56 (0.31) | 0.26 (0.25) [$\alpha = 1.2$] | **0.24** (0.23) |
| | 96 | 0.74 (0.33) | 0.74 (0.32) | 0.34 (0.25) | 0.67 (0.33) | **0.32** (0.30) [$\alpha = 1.4$] | 0.53 (0.28) |
| | 128 | 0.85 (0.34) | 0.81 (0.33) | 0.50 (0.27) | 0.84 (0.34) | **0.44** (0.32) [$\alpha = 1.7$] | 0.73 (0.32) |
| $\mathbf{\Phi}_{\text{con}}$ | 64 | 0.64 (0.32) | 0.74 (0.32) | 0.51 (0.29) | 0.64 (0.31) | **0.31** (0.27) [$\alpha = 1.3$] | 0.57 (0.30) |
| | 96 | 0.74 (0.33) | 0.75 (0.33) | 0.67 (0.31) | 0.68 (0.33) | **0.47** (0.30) [$\alpha = 1.5$] | 0.68 (0.33) |
| | 128 | 0.85 (0.34) | 0.82 (0.34) | 0.79 (0.33) | 0.84 (0.34) | **0.72** (0.33) [$\alpha = 1.9$] | 0.80 (0.33) |

$$\mu_{\text{avg}}(\mathbf{\Psi}) = \frac{1}{N_\beta}\Big( \sum_{(k,j)\in\mathcal{S}_\beta} \big|\mathbf{G}^{[k,j]}\big| \Big), \tag{7.33}$$

where $\mathcal{S}_\beta = \{(k, j) : |\mathbf{G}^{[k,j]}| > \sqrt{\beta}\}$, $N_\beta$ is the number of elements in the set $\mathcal{S}_\beta$, and $\mathbf{G} = \mathbf{\Psi}^{\text{H}}\mathbf{\Psi}$ is the normalized-diagonal Gram matrix. Table 7.1 shows the obtained results for different values of $P$. Moreover, Fig. 7.6 shows the convergence behavior of the iterative methods for the scenarios with $P = 64$ and $P = 128$. For the GD method [1], we use the step size $\zeta = 5 \times 10^{-4}/n$, while for the EGD method, we use $\zeta = 5 \times 10^{-2}/n$, where $n$ is the iteration index.

From Table 7.1, when $P = M = 64$, we can see that the SMCM and the Itr-SVD methods achieve similar performance, where the only difference is that SMCM has a faster convergence rate compared to Itr-SVD, as can be seen from Fig. 7.6. However, as expected, when the ratio $P/M$ increases above 1, the SMCM performance decreases, since the naïve approach of calculating the mixing matrix $\mathbf{\Phi}$ from the designed sensing matrix $\mathbf{\Psi}$ incurs a performance loss. On the other hand, it can be seen that the proposed EGD method has the best performance in almost all of the considered scenarios. Here, we note that the introduced uncertainty measure $\alpha$ has a big impact on the EGD performance and the convergence rate, as can be seen from Fig. 7.7. In general, for a sufficiently large $\alpha$, the EGD converges faster, but its performance degrades and approaches that of the GD. On the other hand, from Fig. 7.7, we can also note that $\alpha$ should not be too small since in this case most of the entries within the resulting error matrix $\tilde{\mathbf{E}}$ will be set to zero. From our simulation results in Table 7.1, we observe that $\alpha$ should be selected so that it is approximately equal to $P/M$.

In this section, we have proposed the two mixing matrix design methods SMCM and EGD via mutual coherence minimization. For the unconstrained mixing matrix and $P = M$, we have shown that the original nonconvex problem can be relaxed and divided into $P$ convex subproblems, which are updated iteratively using an alternating optimization technique. However, SMCM incurs some performance loss for the constrained case and for $P > M$. To overcome this issue, we have proposed the EGD method, which enhances the classical GD-based method of [1] by introducing a shrinking operator on the error matrix. Using computer

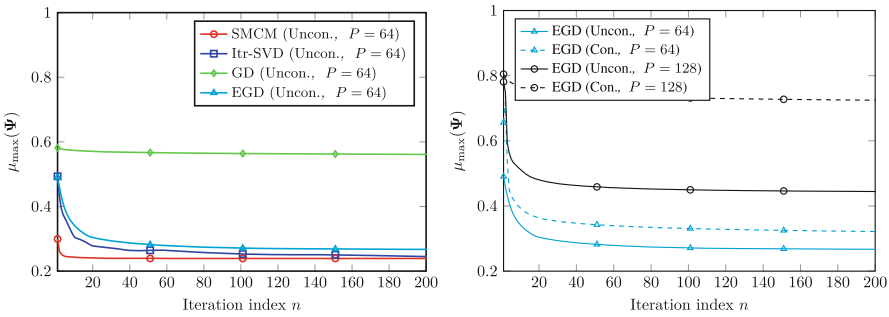



**Fig. 7.6** Coherence $\mu_{\max}(\mathbf{\Psi})$ versus the iteration index

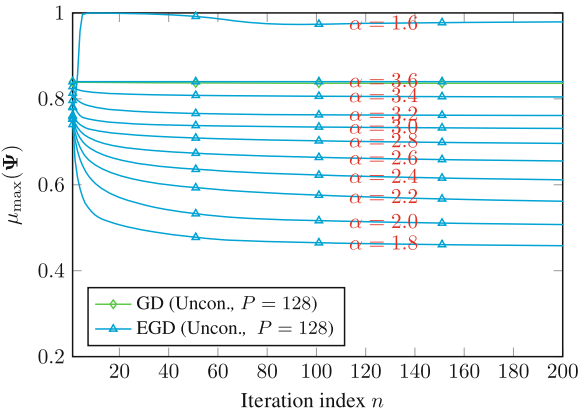



**Fig. 7.7** Coherence $\mu_{\max}(\mathbf{\Psi})$ versus the iteration index

simulations, we have shown that the proposed SMCM and EGD methods have a faster convergence rate and a lower mutual coherence compared to the benchmark methods.

## 7.6 Recovery Algorithms for the Nonlinear Measurement Model

This section is devoted to recovery techniques that explicitly consider the specific structure of the measurements $\mathbf{z}$ themselves. More specifically, we consider the special case of magnitude-only measurements. Hence, we use the information that measurements are nonnegative, and we intend to uniquely recover the phase of the measurement signal along with the sparse representation vector.

### 7.6.1 Sparse Phase Retrieval

In this subsection, we consider the sparse phase retrieval problem [19, 24, 35, 42, 44, 47, 49, 73, 82], which aims to reconstruct an unknown complex-valued sparse signal $\mathbf{x} \in \mathbb{C}^K$ from $M$ noise-corrupted magnitude-only measurements:

$$\mathbf{z} = |\mathbf{A}\mathbf{x}| + \mathbf{n}, \tag{7.34}$$

where $\mathbf{A}$ is a designed sensing matrix, $\mathbf{n} \in \mathbb{C}^M$ is an additive noise vector, and $|\cdot|$ is applied element-wise. The measurement model (7.34) can be viewed as a special case of the system depicted in Fig. 7.1, where $\mathbf{\Phi}$ is the identity and $\mathcal{T}\{\cdot\} = |\cdot|$. The recovery problem can be formulated as the following $\ell_1$-regularized nonlinear least-squares [47, 82]:

$$\min_{\mathbf{x}\in\mathbb{C}^K} h(\mathbf{x}) = \underbrace{\tfrac{1}{2}\left\|\mathbf{z} - |\mathbf{A}\mathbf{x}|\right\|_2^2}_{f(\mathbf{x})} + \underbrace{\lambda \left\|\mathbf{x}\right\|_1}_{g(\mathbf{x})}. \tag{7.35}$$

It is a very challenging optimization problem due to the fact that $g$ is nonsmooth and, more notably, $f$ is nonsmooth and nonconvex. Besides, the original signal $\mathbf{x}$ can only be recovered up to a global phase ambiguity as $\mathbf{x} \cdot e^{j\phi}$ preserves both the magnitude measurements and the sparsity pattern.

We solve problem (7.35) using the STELA algorithm in [82], which is built on the majorization-minimization (MM) techniques in [47] and the block successive convex approximation (BSCA) framework in [75, 83]. The algorithm finds a stationary point of (7.35) according to a generalized concept of stationarity via a sequence of approximate problems that can be solved in parallel [88]. As $f$ in the objective function of (7.35) is nonconvex and nonsmooth, in each iteration we first construct a smooth upper bound function for $f$. Then, a descent direction of the upper bound function is obtained by solving a separable convex approximate problem, and a step size along the descent direction is computed efficiently by exact line search. A decrease of the original objective function $h$ is ensured as its upper bound is decreased. Let $\mathbf{x}^{(l)}$ be the current point in the $l$-th iteration. Specifically, the algorithm performs the following three steps in each iteration:

1. **Smooth majorization.** The quadratic function $f$ in (7.35) can be expanded as

$$f(\mathbf{x}) = \tfrac{1}{2}\left(\|\mathbf{z}\|_2^2 + \|\mathbf{A}\mathbf{x}\|_2^2\right) - \mathbf{z}^{\mathrm{T}}|\mathbf{A}\mathbf{x}|. \tag{7.36}$$

Further, we note that for any $x \in \mathbb{C}$ and $\phi \in [0, 2\pi)$

$$|x| = |x \cdot e^{j\phi}| \geq \mathrm{Re}\{x \cdot e^{j\phi}\}, \tag{7.37}$$

and equality holds for $\phi = -\arg(x)$. Thus, defining $\mathbf{z}^{(l)} = \mathbf{z} \odot e^{j \arg(\mathbf{A}\mathbf{x}^{(l)})}$, where $e^{(\cdot)}$ and $\arg(\cdot)$ are applied element-wise and $\odot$ denotes the Hadamard multiplication, we obtain the following smooth and convex upper bound for $f$ in the $l$-th iteration [47]:

$$\bar{f}^{(l)}(\mathbf{x}) = \tfrac{1}{2}\big(\|\mathbf{z}\|_2^2 + \|\mathbf{A}\mathbf{x}\|_2^2\big) - \mathbf{z}^{\mathrm{T}} \operatorname{Re}\big\{\mathbf{A}\mathbf{x} \odot e^{-j \arg(\mathbf{A}\mathbf{x}^{(l)})}\big\} = \tfrac{1}{2}\|\mathbf{z}^{(l)} - \mathbf{A}\mathbf{x}\|_2^2, \tag{7.38}$$

which is tight at $\mathbf{x}^{(l)}$, i.e., $\bar{f}^{(l)}(\mathbf{x}^{(l)}) = f(\mathbf{x}^{(l)})$. Consequently, function $\bar{h}^{(l)}(\mathbf{x}) = \bar{f}^{(l)}(\mathbf{x}) + g(\mathbf{x})$ is also an upper bound of the objective function $h$ and tight at $\mathbf{x}^{(l)}$.

2. **Descent direction computation.** Departing from the conventional MM algorithm, we minimize a separable convex approximation of $\bar{h}^{(l)}$ because $\bar{h}^{(l)}$ is computationally too expensive to minimize exactly for our present purpose. Based on the Jacobi algorithm [75], the convex approximate problem in the $l$-th iteration around point $\mathbf{x}^{(l)}$ is constructed as

$$\widetilde{\mathbf{x}}^{(l)} = \operatorname*{argmin}_{\mathbf{x} \in \mathbb{C}^K} \sum_{k=1}^{K} \bar{f}^{(l)}\big(x_k, \mathbf{x}_{-k}^{(l)}\big) + g(\mathbf{x}), \tag{7.39}$$

where $\mathbf{x}_{-k}$ is a $(K-1)$-dimensional vector obtained by removing the $k$-th element $x_k$ from $\mathbf{x}$. Problem (7.39) is decomposed into $K$ independent subproblems, which can be solved in parallel with suitable hardware [74]. Each subproblem is a Lagrangian form of single-variate LASSO, which admits a closed-form solution. According to [75, Prop. 1], the vector $\widetilde{\mathbf{x}}^{(l)} - \mathbf{x}^{(l)}$ represents a descent direction of $\bar{h}^{(l)}$. This motivates us to update $\mathbf{x}^{(l)}$ as follows:

$$\mathbf{x}^{(l+1)} = \mathbf{x}^{(l)} + \gamma^{(l)}(\widetilde{\mathbf{x}}^{(l)} - \mathbf{x}^{(l)}), \tag{7.40}$$

where $\gamma^{(l)} \in [0, 1]$ is the step size. When $\widetilde{\mathbf{x}}^{(l)} = \mathbf{x}^{(l)}$, the algorithm has converged to a stationary point of $\bar{h}^{(l)}$, which is also stationary for the original problem (7.35) [83, Thm. 1].

3. **Step size computation.** To efficiently find a proper step size $\gamma^{(l)}$ for the update in (7.40), we perform an exact line search on a differentiable upper bound of $\bar{h}^{(l)}$ [75]. Thus, the computation of step size $\gamma^{(l)}$ is formulated as

$$\gamma^{(l)} = \operatorname*{argmin}_{0 \le \gamma \le 1} \bar{f}^{(l)}\big(\mathbf{x}^{(l)} + \gamma(\widetilde{\mathbf{x}}^{(l)} - \mathbf{x}^{(l)})\big) + g(\mathbf{x}^{(l)}) + \gamma\big(g(\widetilde{\mathbf{x}}^{(l)}) - g(\mathbf{x}^{(l)})\big). \tag{7.41}$$

The line search (7.41) corresponds to minimizing a convex quadratic function in the interval [0, 1], which can be solved in closed form. Using the step size $\gamma^{(l)}$ obtained by the line search (7.41) in the update (7.40), a monotonic decrease of the original objective function $h$ in problem (7.35) is ensured, cf. [82].

The mathematical expressions for the solutions of approximate problem (7.39) and line search (7.41) can be further found in [82]. Simulation results with Gaussian random sensing matrix $\mathbf{A}$ are also provided in [82]. The convergence analysis of the BSCA framework is presented in [83]. Besides, several other applications of the BSCA framework can be found in [37, 38, 76, 78, 79, 81]. Furthermore, nonconvex regularization functions can be employed to resolve the defect that the $\ell_1$-regularization tends to produce biased estimates when the sparse signal has large coefficients [80]. In addition, a comprehensive review of recent advances in phase retrieval from a numerical perspective is presented in [10]. The conditions for unique and stable reconstruction in sparse phase retrieval are discussed in [24, 43].

### 7.6.2 *Phase Retrieval with Dictionary Learning*

In the previous subsection, we considered the phase retrieval problem for signals that are sparse in the standard basis. However, in some cases, the signals that need to be recovered may only be sparse with respect to an unknown dictionary. Therefore, in this subsection, we consider the phase retrieval with dictionary learning problem, which jointly learns a dictionary and sparse representations for reconstructing unknown signals. This recovery problem is involved in several applications such as diffraction imaging [47, 66] and blind channel estimation in multi-antenna random access network [39, 88].

We consider a special case of the system depicted in Fig. 7.1 with a known mixing matrix $\mathbf{\Phi}$ and $\mathcal{T}\{\cdot\} = |\cdot|$:

$$\mathbf{z}(t) = |\mathbf{\Phi}\mathbf{A}\mathbf{x}(t)| + \mathbf{n}(t),\ t = 1, \ldots, D. \tag{7.42}$$

Given $D$ time samples $\mathbf{Z} = [\mathbf{z}(1), \ldots, \mathbf{z}(D)]$, the objective is to jointly recover the unknown sensing matrix $\mathbf{A}$ and sparse transmitted signals $\mathbf{X} = [\mathbf{x}(1), \ldots, \mathbf{x}(D)]$. The recovery problem is then formulated as the following phase retrieval with dictionary learning problem [39, 88]:

$$\min_{\mathbf{A}\in\mathcal{A},\mathbf{X}\in\mathbb{C}^{K\times D}} h(\mathbf{A},\mathbf{X}) = \underbrace{\tfrac{1}{2}\left\|\mathbf{Z} - |\mathbf{\Phi}\mathbf{A}\mathbf{X}|\right\|_2^2}_{f(\mathbf{A},\mathbf{X})} + \underbrace{\lambda\left\|\mathbf{X}\right\|_{1,1}}_{g(\mathbf{X})}. \tag{7.43}$$

To avoid scaling ambiguities, we restrict $\mathbf{A}$ to be in the convex set $\mathcal{A} = \{\mathbf{A} \in \mathbb{C}^{M\times K} : \|\mathbf{a}_k\|_2 \le 1, \forall k = 1, \ldots, K\}$. Also, $D > K$ is required to avoid trivial solutions.

Analogously, a stationary point of problem (7.43) according to a generalized concept of stationarity can be found by using the majorization technique in (7.37) and the BSCA framework [88]. In addition to the procedure described in Sect. 7.6.1, we also partition the variables into two blocks, i.e., $\mathbf{A}$ and $\mathbf{X}$, and select a given

number $k_B \in \{1, 2\}$ of block variables to update in each iteration. The block variables can be selected by cyclic or random update rules [83].

Let $\left(\mathbf{A}^{(l)}, \mathbf{X}^{(l)}\right)$ be the current point in the $l$-th iteration. We first consider the case where both block variables $\mathbf{A}$ and $\mathbf{X}$ are selected to update. Then, the three main steps that are performed in each iteration by the BSCA-based algorithm for problem (7.43) are outlined as follows:

1. **Smooth majorization.** Exploiting the same majorization technique given in (7.37), we construct a smooth upper bound for $f$ in (7.43). Defining $\mathbf{Z}^{(l)} = \mathbf{Z} \odot e^{j \arg(\mathbf{\Phi A X}^{(l)})}$, we can obtain the following smooth upper bound for $f$ in the $l$-th iteration:

$$\bar{f}^{(l)}(\mathbf{A}, \mathbf{X}) = \tfrac{1}{2}\|\mathbf{Z}^{(l)} - \mathbf{\Phi A X}\|_{\mathrm{F}}^2, \tag{7.44}$$

   which is tight at $(\mathbf{A}^{(l)}, \mathbf{X}^{(l)})$. Similarly, we construct function $\bar{h}^{(l)}(\mathbf{A}, \mathbf{X}) = \bar{f}^{(l)}(\mathbf{A}, \mathbf{X}) + g(\mathbf{X})$ as an upper bound of the objective function $h$ that is tight at $\left(\mathbf{A}^{(l)}, \mathbf{X}^{(l)}\right)$. However, we remark that, unlike in Sect. 7.6.1, the upper-bound function $\bar{f}^{(l)}$ in (7.44) is nonconvex due to the bilinear terms $\mathbf{AX}$. Therefore, the convex approximation in the next step becomes necessary for efficiently finding a descent direction.
2. **Descent direction computation.** Based on the Jacobi algorithm [75], the separable convex approximation for the minimization of $\bar{h}^{(l)}$ is constructed as

$$\left(\widetilde{\mathbf{A}}^{(l)}, \widetilde{\mathbf{X}}^{(l)}\right) \in \underset{\mathbf{A}\in\mathcal{A},\mathbf{X}}{\operatorname{argmin}} \left\{ \begin{array}{l} \sum_{m=1}^{M}\sum_{k=1}^{K} \bar{f}^{(l)}\left(x_{mk}, \mathbf{A}^{(l)}, \mathbf{X}_{-mk}^{(l)}\right) \\ +\sum_{m=1}^{M} \bar{f}^{(l)}\left(\mathbf{a}_k, \mathbf{A}_{-k}^{(l)}, \mathbf{X}^{(l)}\right) + g(\mathbf{X}) \end{array} \right\}, \tag{7.45}$$

   where $\mathbf{A}_{-k}$ is an $M \times (K-1)$ matrix obtained by removing the $k$-th column $\mathbf{a}_k$ from $\mathbf{A}$ and $\mathbf{X}_{-mk}$ denotes the collection of all entries of $\mathbf{X}$ except the $(m, k)$-th entry $x_{mk}$. Problem (7.35) can be decomposed into $K + (K \times D)$ independent subproblems. Each subproblem can be solved either in closed form or by an efficient algorithm. Then, the difference $\left(\widetilde{\mathbf{A}}^{(l)} - \mathbf{A}^{(l)}, \widetilde{\mathbf{X}}^{(l)} - \mathbf{X}^{(l)}\right)$ represents a descent direction of $\bar{h}^{(l)}$ in the domain of problem (7.43). Defining $\Delta\mathbf{A} = \widetilde{\mathbf{A}}^{(l)} - \mathbf{A}^{(l)}$ and $\Delta\mathbf{X} = \widetilde{\mathbf{X}}^{(l)} - \mathbf{X}^{(l)}$, the following simultaneous update rule can be applied:

$$\mathbf{A}^{(l+1)} = \mathbf{A}^{(l)} + \gamma^{(l)}\Delta\mathbf{A} \quad \text{and} \quad \mathbf{X}^{(l+1)} = \mathbf{X}^{(l)} + \gamma^{(l)}\Delta\mathbf{X}, \tag{7.46}$$

   with a proper step size $\gamma^{(l)} \in [0, 1]$. When $\left(\widetilde{\mathbf{A}}^{(l)}, \widetilde{\mathbf{X}}^{(l)}\right) = \left(\mathbf{A}^{(t)}, \mathbf{X}^{(l)}\right)$, the algorithm has converged to a stationary point of $\bar{h}^{(l)}$, which is also stationary for the original problem (7.43) [83, Thm. 1].

3. **Step size computation.** We perform an exact line search on a differentiable upper bound of $\bar{h}^{(l)}$ to efficiently find a step size $\gamma^{(l)}$ that ensures a monotonic decrease of the original objective function $h$ in (7.43). The computation of step size $\gamma^{(l)}$ is then formulated as

$$\gamma^{(l)} = \underset{0\le\gamma\le 1}{\operatorname{argmin}} \left\{ \begin{array}{l} \bar{f}^{(l)}\left(\mathbf{A}^{(l)} + \gamma\Delta\mathbf{A}, \mathbf{X}^{(l)} + \gamma\Delta\mathbf{X}\right) \\ +g\left(\mathbf{X}^{(l)}\right) + \gamma\left(g\left(\widetilde{\mathbf{X}}^{(l)}\right) - g\left(\mathbf{X}^{(l)}\right)\right) \end{array} \right\}. \tag{7.47}$$

Problem (7.47) can be solved by rooting its derivative, a third-order polynomial, which admits a closed-form expression.

In contrast to the above joint update case, if only one block variable is selected to update in the $l$-th iteration, then we solve the approximate problem (7.45) only with respect to the selected block variable, which requires solving only the corresponding subproblems. Moreover, the update (7.46) is also performed only on the selected block variable, which is equivalent to setting the difference of the non-selected block variable to be all zero. Further, when either of the matrices $\Delta\mathbf{A}$ and $\Delta\mathbf{X}$ is all zero, the line search problem (7.47) reduces to a simple convex quadratic program.

Details of the BSCA-based algorithm for phase retrieval with dictionary learning and results from numerical experiments can be further found in [39].

## 7.7 Conclusions

Compressed sensing (CS) is a powerful technique for estimating sparse signals, which can be recovered, under mild conditions, from far fewer samples than otherwise indicated by the Nyquist-Shannon sampling theorem. Moreover, it was observed that incorporating side constraints not only improves the recovery guarantees but also reduces the required number of samples. This chapter builds on this important observation by addressing sparse signal reconstruction under various types of structural side constraints, including integrality, constant-modulus, row- and rank-sparsity, and strict non-circularity constraints. Moreover, this chapter addresses the measurement system design for linear and nonlinear measurements of sparse signals. For the linear measurement systems, two mixing matrix design methods based on mutual coherence minimization are proposed, where constant-modulus constraints are imposed element-wise to satisfy the mixing matrix hardware that involves cost-efficient analog phase shifters. For nonlinear measurement systems, parallel optimization design algorithms are proposed to efficiently compute the stationary points in the sparse phase retrieval problem with and without dictionary learning.

## References


1. Abolghasemi, V., Ferdowsi, S., Makkiabadi, B., Sanei, S.: On optimization of the measurement matrix for compressive sensing. In: Proceedings of the 18th European Signal Processing Conference, pp. 427–431 (2010)
2. Ardah, K., d. Almeida, A.L.F., Haardt, M.: Low-complexity millimeter wave CSI estimation in MIMO-OFDM hybrid beamforming systems. In: WSA 2019; 23rd International ITG Workshop on Smart Antennas, pp. 1–5 (2019)
3. Ardah, K., de Almeida, A.L.F., Haardt, M.: A gridless CS approach for channel estimation in hybrid massive MIMO systems. In: 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 4160–4164 (2019)
4. Ardah, K., Pesavento, M., Haardt, M.: A novel sensing matrix design for compressed sensing via mutual coherence minimization. In: 2019 IEEE 8th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), pp. 66–70 (2019)
5. Ardah, K., Sokal, B., de Almeida, A.L.F., Haardt, M.: Compressed sensing based channel estimation and open-loop training design for hybrid analog-digital massive MIMO systems. In: 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 4597–4601 (2020)
6. Ardah, K., Gherekhloo, S., de Almeida, A.L.F., Haardt, M.: TRICE: A channel estimation framework for RIS-aided millimeter-wave MIMO systems. IEEE Signal Process. Lett. **28**, 513–517 (2021)
7. Boyer, R., Haardt, M.: Noisy compressive sampling based on block-sparse tensors: Performance limits and beamforming techniques. IEEE Trans. Signal Process. (23), 6075–6088 (2016)
8. Choi, J.W., Shim, B., Ding, Y., Rao, B., Kim, D.I.: Compressed sensing for wireless communications: Useful tips and tricks. IEEE Commun. Surv. Tutorials **19**(3), 1527–1550 (2017)
9. Donoho, D.L.: Compressed sensing. IEEE Trans. Inf. Theory **52**(4), 1289–1306 (2006)
10. Fannjiang, A., Strohmer, T.: The numerics of phase retrieval. Acta Numerica **29**, 125–228 (2020)
11. Fischer, T., Pfetsch, M.E.: Monoidal cut strengthening and generalized mixed-integer rounding for disjunctive programs. Oper. Res. Lett. **45**(6), 556–560 (2017)
12. Fischer, T., Pfetsch, M.E.: Branch-and-cut for linear programs with overlapping SOS1 constraints. Math. Prog. Comp. **10**(1), 33–68 (2018)
13. Fischer, T., Hegde, G., Matter, F., Pesavento, M., Pfetsch, M.E., Tillmann, A.M.: Joint antenna selection and phase-only beamforming using mixed-integer nonlinear programming. In: WSA 2018; 22nd International ITG Workshop on Smart Antennas, pp. 1–7 (2018)
14. Foucart, S., Rauhut, H.: A Mathematical Introduction to Compressive Sensing. Applied and Numerical Harmonic Analysis. Birkhäuser/Springer, New York (2013)
15. Gao, F., Tian, Z., Larsson, E.G., Pesavento, M., Jin, S.: Introduction to the special issue on array signal processing for angular models in massive MIMO communications. IEEE J. Sel. Topics Signal Process. **13**(5), 882–885 (2019)
16. Gribonval, R., Nielsen, M.: Sparse representations in unions of bases. IEEE Trans. Inf. Theory **49**(12), 3320–3325 (2003)
17. Haardt, M., Roemer, F.: Enhancements of Unitary ESPRIT for non-circular sources. In: 2004 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Montreal, vol. II, pp. 101–104 (2004)
18. Haardt, M., Pesavento, M., Roemer, F., El Korso, M.N.: Subspace methods and exploitation of special array structures. In: Zoubir, A.M., Viberg, M., Chellappa, R., Theodoridis, S. (eds.) Academic Press Library in Signal Processing: Volume 3 – Array and Statistical Signal Processing, pp. 651–717. Elsevier, Amsterdam (2014). Chapter 15
19. Hand, P., Voroninski, V.: Compressed sensing from phaseless Gaussian measurements via linear programming in the natural parameter space. Preprint, arXiv:1611.05985 (2016)

20. Hegde, G., Yang, Y., Steffens, C., Pesavento, M.: Parallel low-complexity M-PSK detector for large-scale MIMO systems. In: 2016 IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM), pp. 1–5. IEEE, Piscataway (2016)
21. Hegde, G., Pesavento, M., Pfetsch, M.E.: Joint active device identification and symbol detection using sparse constraints in massive MIMO systems. In: 2017 25th European Signal Processing Conference (EUSIPCO), pp. 703–707. IEEE, Piscataway (2017)
22. Heuer, J., Matter, F., Pfetsch, M.E., Theobald, T.: Block-sparse recovery of semidefinite systems and generalized null space conditions. Linear Algebra Appl. **603**, 470–495 (2020)
23. Hyder, M.M., Mahata, K.: Direction-of-arrival estimation using a mixed $\ell_{2,0}$ norm approximation. IEEE Trans. Signal Process. **58**(9), 4646–4655 (2010)
24. Jaganathan, K., Oymak, S., Hassibi, B.: Sparse phase retrieval: uniqueness guarantees and recovery algorithms. IEEE Trans. Signal Process. **65**(9), 2402–2410 (2017)
25. Juditsky, A., Karzan, F.K., Nemirovski, A.: On a unified view of nullspace-type conditions for recoveries associated with general sparsity structures. Linear Algebra Appl. **441**, 124–151 (2014)
26. Keiper, S., Kutyniok, G., Lee, D.G., Pfander, G.E.: Compressed sensing for finite-valued signals. Linear Algebra Appl. **532**, 570–613 (2017)
27. Khajehnejad, M.A., Dimakis, A.G., Xu, W., Hassibi, B.: Sparse recovery of nonnegative signals with minimal expansion. IEEE Trans. Signal Process. **59**(1), 196–208 (2011)
28. Kong, L., Sun, J., Xiu, N.: S-semigoodness for low-rank semidefinite matrix recovery. Pac. J. Optim. **10**(1), 73–83 (2014)
29. Kowalski, M.: Sparse regression using mixed norms. Appl. Comput. Harmon. Anal. **27**(3), 303–324 (2009)
30. Krim, H., Viberg, M.: Two decades of array signal processing research: the parametric approach. IEEE Signal Process. Mag. **13**(4), 67–94 (1996)
31. Kuske, J., Swoboda, P., Petra, S.: A novel convex relaxation for non-binary discrete tomography. In: International Conference on Scale Space and Variational Methods in Computer Vision, pp. 235–246. Springer, Berlin (2017)
32. Kushe, G., Yang, Y., Steffens, C., Pesavento, M.: A parallel sparse regularization method for structured multilinear low-rank tensor decomposition. In: 2019 27th European Signal Processing Conference (EUSIPCO), pp. 1–5 (2019)
33. Kushe, G., Yang, Y., Pesavento, M.: A block successive convex approximation framework for multidimensional harmonic retrieval and imperfect measurements. In: WSA 2020; 24th International ITG Workshop on Smart Antennas, pp. 1–5 (2020)
34. Lange, J.H., Pfetsch, M.E., Seib, B.M., Tillmann, A.M.: Sparse recovery with integrality constraints. Discrete Applied Math. **283**, 346–366 (2020)
35. Li, X., Voroninski, V.: Sparse signal recovery from quadratic measurements via convex programming. SIAM J. Math. Anal. **45**(5), 3019–3033 (2013)
36. Li, X., Ye, J., Li, G., Bai, H., Jiang, Q.: A new approach to sensing matrix optimization using steepest descent algorithm. In: 2015 34th Chinese Control Conference (CCC), pp. 4939–4944 (2015)
37. Liu, T., Hoang, M.T., Yang, Y., Pesavento, M.: A block coordinate descent algorithm for sparse Gaussian graphical model inference with Laplacian constraints. In: 2019 IEEE 8th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), pp. 236–240 (2019)
38. Liu, T., Hoang, M.T., Yang, Y., Pesavento, M.: A parallel optimization approach on the infinity norm minimization problem. In: 2019 27th European Signal Processing Conference (EUSIPCO), pp. 1–5. IEEE, Piscataway (2019)
39. Liu, T., Tillmann, A.M., Yang, Y., Eldar, Y.C., Pesavento, M.: A parallel algorithm for phase retrieval with dictionary learning. In: 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (2021)
40. Lu, C., Li, H., Lin, Z.: Optimized projections for compressed sensing via direct mutual coherence minimization. Signal Process. **151**, 45–55 (2018)

41. Malioutov, D., Çetin, M., Willsky, A.: A sparse signal reconstruction perspective for source localization with sensor arrays. IEEE Trans. Signal Process. **53**(8), 3010–3022 (2005)
42. Netrapalli, P., Jain, P., Sanghavi, S.: Phase retrieval using alternating minimization. IEEE Trans. Signal Process. **63**(18), 4814–4826 (2015)
43. Ohlsson, H., Eldar, Y.C.: On conditions for uniqueness in sparse phase retrieval. In: 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1841–1845 (2014)
44. Ohlsson, H., Yang, A., Dong, R., Sastry, S.: CPRL – an extension of compressive sensing to the phase retrieval problem. In: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems, pp. 1367–1375. Curran Associates Inc., Red Hook (2012)
45. Oymak, S., Hassibi, B.: New null space results and recovery thresholds for matrix rank minimization. In: Proceedings of the ISIT 2011. Preprint. arXiv:1011.6326 (2010)
46. Park, J., Lee, G., Sung, Y., Yukawa, M.: Coordinated beamforming with relaxed zero forcing: the sequential orthogonal projection combining method and rate control. IEEE Trans. Signal Process. **61**(12), 3100–3112 (2013)
47. Qiu, T., Palomar, D.P.: Undersampled sparse phase retrieval via majorization-minimization. IEEE Trans. Signal Process. **65**(22), 5957–5969 (2017)
48. Rani, M., Dhok, S.B., Deshmukh, R.B.: A systematic review of compressive sensing: concepts, implementations and applications. IEEE Access **6**, 4875–4894 (2018)
49. Shechtman, Y., Beck, A., Eldar, Y.C.: Gespar: Efficient phase retrieval of sparse signals. IEEE Trans. Signal Process. **62**(4), 928–938 (2014)
50. Steffens, C., Pesavento, M.: Block- and rank-sparse recovery for direction finding in partly calibrated arrays. IEEE Trans. Signal Process. **66**(2), 384–399 (2018)
51. Steffens, C., Pesavento, M.: Collaborative Sensing Techniques, chap. 7, pp. 121–145. John Wiley & Sons Ltd., Hoboken (2020)
52. Steffens, C., Yang, Y., Pesavento, M.: Multidimensional sparse recovery for MIMO channel parameter estimation. In: 2016 24th European Signal Processing Conference (EUSIPCO), pp. 66–70 (2016)
53. Steffens, C., Suleiman, W., Sorg, A., Pesavento, M.: Gridless compressed sensing under shift-invariant sampling. In: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 4735–4739 (2017)
54. Steffens, C., Pesavento, M., Pfetsch, M.E.: A compact formulation for the $\ell_{2,1}$ mixed-norm minimization problem. IEEE Trans. Signal Process. **66**(6), 1483–1497 (2018)
55. Steinwandt, J., Roemer, F., Haardt, M.: Sparsity-based direction-of-arrival estimation for strictly non-circular sources. In: 2016 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Shanghai (2016)
56. Steinwandt, J., Roemer, F., Haardt, M., Del Galdo, G.: Deterministic Cramér-Rao bound for strictly non-circular sources and analytical analysis of the achievable gains. IEEE Trans. Signal Process. **64**(17), 4417–4431 (2016)
57. Steinwandt, J., Roemer, F., Steffens, C., Haardt, M., Pesavento, M.: Gridless superresolution direction finding for strictly non-circular sources based on atomic norm minimization. In: 2016 50th Asilomar Conference on Signals, Systems, and Computers, Pacific Grove (2016)
58. Steinwandt, J., Steffens, C., Pesavento, M., Haardt, M.: Sparsity-aware direction finding for strictly non-circular sources based on rank minimization. In: 2016 IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM), Rio de Janeiro (2016)
59. Steinwandt, J., Roemer, F., Haardt, M.: Generalized least squares for ESPRIT-type direction of arrival estimation. IEEE Signal Process. Lett. **24**(11), 1681–1685 (2017)
60. Steinwandt, J., Roemer, F., Haardt, M.: Performance analysis of ESPRIT-type algorithms for co-array structures. In: 2017 IEEE 7th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), pp. 1–5 (2017)
61. Steinwandt, J., Roemer, F., Haardt, M., Del Galdo, G.: Performance analysis of multi-dimensional ESPRIT-type algorithms for arbitrary and strictly non-circular sources with spatial smoothing. IEEE Trans. Signal Process. **65**(9), 2262–2276 (2017)

62. Stojnic, M.: Recovery thresholds for $\ell_1$ optimization in binary compressed sensing. In: 2010 IEEE International Symposium on Information Theory, pp. 1593–1597. IEEE, Piscataway (2010)
63. Stojnic, M., Parvaresh, F., Hassibi, B.: On the reconstruction of block-sparse signals with an optimal number of measurements. IEEE Trans. Signal Process. **57**(8), 3075–3085 (2009)
64. Suleiman, W., Steffens, C., Sorg, A., Pesavento, M.: Gridless compressed sensing for fully augmentable arrays. In: 2017 25th European Signal Processing Conference (EUSIPCO), pp. 1986–1990 (2017)
65. Tillmann, A.M., Pfetsch, M.E.: The computational complexity of the restricted isometry property, the nullspace property, and related concepts in compressed sensing. IEEE Trans. Inf. Theory **60**(2), 1248–1259 (2014)
66. Tillmann, A.M., Eldar, Y.C., Mairal, J.: DOLPHIn – dictionary learning for phase retrieval. IEEE Trans. Signal Process. **64**(24), 6485–6500 (2016)
67. Tropp, J.A.: Algorithms for simultaneous sparse approximation. Part II: convex relaxation. Signal Process. **86**(3), 589–602 (2006)
68. Tropp, J.A., Dhillon, I.S., Heath, R.W., Strohmer, T.: Designing structured tight frames via an alternating projection method. IEEE Trans. Inf. Theory **51**(1), 188–209 (2005)
69. Turlach, B.A., Venables, W.N., Wright, S.J.: Simultaneous variable selection. Technometrics **47**(3), 349–363 (2005)
70. Van Trees, H.L.: Optimum Array Processing. Wiley, New York (2002)
71. Vigerske, S.: Decomposition in multistage stochastic programming and a constraint integer programming approach to mixed-integer nonlinear programming. Ph.D. Thesis, Humboldt-Universität zu Berlin (2013)
72. Walewski, A.C., Steffens, C., Pesavento, M.: Off-grid parameter estimation based on joint sparse regularization. In: SCC 2017; 11th International ITG Conference on Systems, Communications and Coding, pp. 1–6 (2017)
73. Wang, G., Zhang, L., Giannakis, G.B., Akçakaya, M., Chen, J.: Sparse phase retrieval via truncated amplitude flow. IEEE Trans. Signal Process. **66**(2), 479–491 (2018)
74. Wang, X., Liu, T., Trinh-Hoang, M., Pesavento, M.: GPU-accelerated parallel optimization for sparse regularization. In: 2020 IEEE 11th Sensor Array and Multichannel Signal Processing Workshop (SAM), pp. 1–5 (2020)
75. Yang, Y., Pesavento, M.: A unified successive pseudoconvex approximation framework. IEEE Trans. Signal Process. **65**(13), 3313–3328 (2017)
76. Yang, Y., Pesavento, M.: Energy efficiency in MIMO interference channels: social optimality and max-min fairness. In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 3689–3693 (2018)
77. Yang, Y., Pesavento, M.: A parallel best-response algorithm with exact line search for nonconvex sparsity-regularized rank minimization. In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 6323–6327 (2018)
78. Yang, Y., Pesavento, M., Zhang, M., Palomar, D.P.: An online parallel algorithm for recursive estimation of sparse signals. IEEE Trans. Signal Inf. Process. Netw. **2**(3), 290–305 (2016)
79. Yang, Y., Pesavento, M., Chatzinotas, S., Ottersten, B.: Parallel and hybrid soft-thresholding algorithms with line search for sparse nonlinear regression. In: European Signal Processing Conference, vol. 2018, pp. 1587–1591 (2018)
80. Yang, Y., Pesavento, M., Chatzinotas, S., Ottersten, B.: Successive convex approximation algorithms for sparse signal estimation with nonconvex regularizations. IEEE J. Sel. Topics Signal Process. **12**(6), 1286–1302 (2018)
81. Yang, Y., Pesavento, M., Chatzinotas, S., Ottersten, B.: Energy efficiency optimization in MIMO interference channels: a successive pseudoconvex approximation approach. IEEE Trans. Signal Process. **67**(15), 4107–4121 (2019)
82. Yang, Y., Pesavento, M., Eldar, Y.C., Ottersten, B.: Parallel coordinate descent algorithms for sparse phase retrieval. In: 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 7670–7674 (2019)

83. Yang, Y., Pesavento, M., Luo, Z.Q., Ottersten, B.: Inexact block coordinate descent algorithms for nonsmooth nonconvex optimization. IEEE Trans. Signal Process. **68**, 947–961 (2020)
84. Yu, L., Li, G., Chang, L.: Optimizing projection matrix for compressed sensing systems. In: 2011 8th International Conference on Information, Communications Signal Processing (ICICS), pp. 1–5 (2011)
85. Yuan, M., Lin, Y.: Model selection and estimation in regression with grouped variables. J. R. Stat. Soc. Series B (Statistical Methodology) **68**(1), 49–67 (2006)
86. Zelnik-Manor, L., Rosenblum, K., Eldar, Y.C.: Sensing matrix optimization for block-sparse decoding. IEEE Trans. Signal Process. **59**(9), 4300–4312 (2011)
87. Zhang, Y.: A simple proof for recoverability of $\ell_1$-minimization (II): the nonnegativity case. Technical report TR05-10, Dept. of Computational and Applied Mathematics, Rice University (2005)
88. Liu, T., Tillmann, A.M., Yang, Y., Eldar, Y.C., Pesavento, M.: Extended Successive Convex Approximation for Phase Retrieval with Dictionary Learning. Preprint, arXiv:2109.05646 (2022)

# Chapter 8
# Compressive Sensing and Neural Networks from a Statistical Learning Perspective


**Arash Behboodi, Holger Rauhut, and Ekkehard Schnoor**

## 8.1 Introduction

Learning representations of, or extracting features from, data is an important aspect of deep neural networks. In the past decade, this approach has led to impressive results and achieved state-of-the-art performances, e.g., for various classification tasks. However, due to the black-box nature of the end-to-end learning of neural networks, such features are usually abstract and difficult to interpret. On the other hand, algorithms such as the iterative soft-thresholding algorithm (ISTA) can be regarded as neural networks. Thus, with the help of modern deep learning software libraries, they can easily be implemented and optimized, such that the trained parameters can adapt to datasets of interest. When such algorithms are well-understood, it can be possible to transfer results shown for the classical variant to their neural network variant and in this way increase our understanding of deep neural networks. A class of neural networks that we discuss in the present work aims at joint reconstruction and dictionary learning problem based on unfolding iterative soft-thresholding algorithm. Here, unfolding means that each step of an iterative algorithm constitutes a neural network layer whose parameters can be learned from data.

Here, the learned representation (a dictionary) is a very well-understood model in image and signal processing, which can be easily interpreted and visualized. As a practical application, one may think of reconstructing images from measurements taken by a medical imaging device. Instead of only trying to reconstruct the image,

A. Behboodi
Institute for Theoretical Information Technology, RWTH Aachen University, Aachen, Germany
e-mail: arash.behboodi@ti.rwth-aachen.de

H. Rauhut · E. Schnoor (✉)
Chair for Mathematics of Information Processing, RWTH Aachen University, Aachen, Germany
e-mail: rauhut@mathc.rwth-aachen.de; schnoor@mathc.rwth-aachen.de



G. Kutyniok et al. (eds.), *Compressed Sensing in Information Processing*, Applied and Numerical Harmonic Analysis, https://doi.org/10.1007/978-3-031-09745-4_8

we would also like to implicitly learn a meaningful representation system that is adapted to the image class of interest and leads to good generalization (e.g., when taking measurements of new patients). More generally, this is the approach of solving inverse problems in a data-driven way, e.g., by training neural networks [3, 16].

The natural question arises how well these learned reconstruction methods work. We take the viewpoint of statistical learning theory and assume the data (signals, images, etc.) to be generated independently by some unknown distribution. Generalization bounds give probabilistic estimates on the difference between the true error (with respect to the unknown distribution) and the empirical error for a hypothesis function. Thereby, such bounds predict how well a learned neural network performs on yet unseen data. By now, classical results bound the generalization error in terms of the VC dimension or based on the Rademacher complexity [4, 41]. More recent methods include a compression approach [2] and a PAC Bayesian approach [36]. So far, generalization properties of neural networks have been studied mostly in the context of classification using feed-forward neural networks, see, e.g., [5, 15, 36]. Especially, in the overparameterized scenario with more network parameters than samples that is common in deep learning, it is still a mystery why learned networks generalize very well, and the present bounds cannot yet explain their success [21, 33, 54], although some works attribute this to the so-called implicit bias of learning algorithms [8, 34, 35, 37] such as the commonly used (stochastic) gradient descent. We will, however, not pursue this direction further in this chapter.

The case studied here, a recurrent neural network used for a regression problem, has received less attention so far from the perspective of generalization.

Due to the weight sharing, this is a non-overparameterized network. However, it is straightforward to decouple the layers and thus obtain a network that is more similar to standard feed-forward neural networks. Furthermore, we impose an orthogonality constraint on the dictionary, which in fact constitutes the learned parameters of the network. We derive generalization bounds for such thresholding networks with orthogonal dictionaries. In order to upper bound the Rademacher complexity of the hypothesis class consisting of such deep networks, we apply a generalization of Talagrand's contraction principle [28] for vector-valued functions, which is typically not needed when considering real-valued hypothesis classes, e.g., with the ramp loss (applied to the margin) in a multiclass classification problem [5]. A similar idea for multiclass classification tasks has been tried in [37]. We further estimate the resulting expectation of the supremum of a certain Rademacher process via Dudley's integral (which in particular involves covering numbers) to upper bound the Rademacher complexity of hypothesis classes consisting of such deep networks.

Sample complexity of dictionary learning has been studied before in the literature [14, 18, 19, 40, 46]. The authors in [46] also use a Rademacher complexity analysis for dictionary learning, but they aim at sparse representation of signals rather than reconstruction from compressed measurements, and moreover, they do not use neural network structures. Fundamental limits of dictionary learning from an information-theoretic perspective have been studied in [22, 23]. Uniqueness about

our perspective and different to the cited papers is our approach for determining the sample complexity based on learning a dictionary by training a neural network.

This chapter is structured as follows. In Sect. 8.2, we introduce learned soft iterative thresholding architecture, define the generalization error, and review some of the related works. We discuss works on generalization bounds for deep neural networks in Sect. 8.3 and introduce Rademacher complexity analysis. The main result of this chapter with detailed proofs is given in Sect. 8.4. Finally, we present the numerical results in Sect. 8.5.

### *8.1.1 Notation*

Vectors $\mathbf{x} \in \mathbb{R}^N$ and matrices $\mathbf{A} \in \mathbb{R}^{n\times N}$ are denoted with bold letters, unlike scalars $\lambda \in \mathbb{R}$. We will denote the spectral norm by $\|\mathbf{A}\|_{2\to 2}$ and the Frobenius norm by $\|\mathbf{A}\|_F$. The $N \times m$ matrix $\mathbf{X}$ contains the data points, $\mathbf{x}_1, \ldots, \mathbf{x}_m \in \mathbb{R}^N$ as its columns, analogously $\mathbf{Y} \in \mathbb{R}^{n\times m}$ to collect the measurements $\mathbf{y}_1, \ldots, \mathbf{y}_m \in \mathbb{R}^n$. As a short notation for indices, we use $[m] := \{1, \ldots, m\}$. To make the notation more compact, with a slight abuse of notation, for functions $f : \mathbb{R}^n \to \mathbb{R}^N$, we denote by $f(\mathbf{Y})$ the matrix whose $i$-th column is $f(\mathbf{y}_i)$. The unit ball of the $n$-dimensional normed space $\mathbb{R}^n$ is denoted by $B^n_{\|\cdot\|} := \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\| \le 1\}$. The covering number $\mathcal{N}(\mathcal{M}, d, \epsilon)$ of a metric space $(\mathcal{M}, d)$ at level $\epsilon$ is defined as the smallest number of balls of radius $\epsilon$ with respect to $d$ required to cover $\mathcal{M}$. When the metric is induced by some norm, we write $\mathcal{N}(\mathcal{M}, \|\cdot\|, \epsilon)$. We denote the $N$-dimensional orthogonal group by $O(N)$.

## 8.2 Deep Learning and Inverse Problems

During recent years, many works studied the application of neural networks in solving inverse problems (see, for example, [9, 13]). In this chapter, we focus on joint dictionary learning and sparse recovery using neural networks. Compressive sensing using dictionaries has been studied before, but, in contrast to the scenario discussed here, typically using a fixed (and possibly even redundant) dictionary and a random measurement matrix [39]. The idea of interpreting thresholded gradient steps of iterative algorithms such as ISTA [11] for sparse recovery as layers of neural networks is well-known since [17] and has since then been an active research topic, see, e.g., [7, 24, 30, 32, 49, 50]. Thresholding networks fall into the larger class of proximal neural networks studied in [20]. The key aspect is to learn weight matrices for an unfolded version of ISTA. Different works focus on different parameterizations of the network for faster convergence and better reconstructions. Learning the dictionary can also be implicit in these works. In this chapter, we consider algorithms that try to find a dictionary suitable for reconstruction. Some of the examples of these algorithms are the recently suggested Learning ISTA (LISTA)

[17], Ada-LISTA [1], and convolutional sparse coding [44] that learn efficient sparse and low-rank models [43]. Like many other related papers, such as ISTA-Net [53], these methods are mainly motivated by applications such as inpainting [1].

Instead of novel algorithmic aspects, our contribution is to conduct a generalization analysis for these algorithms, which to the best of our knowledge has not been addressed in the literature before in this particular setting. In this way, we connect this line of research with recent developments [5, 15] in the study of generalization of deep neural networks.

### *8.2.1 Learned Iterative Soft Thresholding*

Let us begin by recalling the well-known iterative soft-thresholding algorithm (ISTA) and how it can be interpreted as a neural network. Given a high-dimensional $s$-sparse signal $\mathbf{x} \in \mathbb{R}^N$ and a measurement matrix $\mathbf{A} \in \mathbb{R}^{n \times N}$ (i.e., taking $n$ linear measurements, with typically $s \ll N$), we would like to recover $\mathbf{x}$ from given $\mathbf{y} = \mathbf{A}\mathbf{x}$. Although this is an under-determined linear system of equations, under certain conditions on the signal (typically, as already mentioned above: sparsity) and (random) measurement matrix (null space property, restricted isometry property), the true signal $\mathbf{x}$ can be recovered [12]. A well-known reconstruction method is $\ell_1$-minimization, which consists in computing a minimizer of the convex optimization problem

$$\min_{\mathbf{x} \in \mathbb{R}^N} \frac{1}{2} \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 + \lambda \|\mathbf{x}\|_1, \tag{8.1}$$

where $\|\mathbf{x}\|_1 = \sum_{\ell=1}^N |x_\ell|$ is the $\ell_1$-norm and $\lambda > 0$ is a regularization parameter. An actual algorithm for computing such minimizer is ISTA [11], where we initialize $\mathbf{x}^0 = \mathbf{0}$ and then recursively compute

$$\begin{aligned} \mathbf{x}^{k+1} &= S_{\tau\lambda}\left[\mathbf{x}^k + \tau \mathbf{A}^\top (\mathbf{y} - \mathbf{A}\mathbf{x}^k)\right] \\ &= S_{\tau\lambda}\left[\left(\mathbf{I} - \tau \mathbf{A}^\top \mathbf{A}\right)\mathbf{x}^k + \tau \mathbf{A}^\top \mathbf{y}\right], \end{aligned} \tag{8.2}$$

where $\lambda$ and $\tau$ are parameters of the algorithm, and $S_\lambda$ (applied entrywise) is the shrinkage operator defined as

$$S_\lambda : \mathbb{R} \to \mathbb{R}, \qquad x \mapsto \begin{cases} 0 & \text{if } |x| \le \lambda, \\ x - \lambda \operatorname{sign}(x) & \text{if } |x| > \lambda, \end{cases} \tag{8.3}$$

which can also be expressed in closed form as $S_\lambda(x) = \operatorname{sign}(x) \cdot \max(0, |x| - \lambda)$ for any $x \in \mathbb{R}$. It is well-known, see, e.g., [11], that $\mathbf{x}^k$ converges to a minimizer of (8.1) under the condition

$$\tau \|\mathbf{A}\|_{2\to 2}^2 \leq 1. \tag{8.4}$$

Note that (8.2) can be interpreted as a layer of a neural network with weight matrix $\mathbf{I} - \tau \mathbf{A}^\top \mathbf{A}$, bias $\tau \mathbf{A}^\top \mathbf{y}$ and activation function $S_{\tau\lambda}$. As a side remark, let us observe that $S_\lambda$ can be written as the sum of two *rectified linear units* via $S_\lambda(x) = \mathrm{ReLU}(x - \lambda) - \mathrm{ReLU}(-x - \lambda)$. Here, $\mathrm{ReLU}(x) = \max(0, x)$ is one of the most popular activation functions used by deep learning practitioners, so that it is also often the default choice for theoretical investigations. While this may be regarded as a natural connection between ISTA and neural networks, we will make no more use of it, as it turned out to be convenient enough to work with $S_\lambda$ as the activation function itself.

This interpretation of ISTA as an *unfolded neural network* has been studied for the first time in [17] leading to the introduction of LISTA. Since then, it has inspired research at the intersection of neural networks and inverse problems in recent years, and many variants of neural-network-enhanced iterative thresholding algorithms have been proposed by now.

Note that in the current form ISTA only takes the form of a neural network but has no trainable parameters. To introduce trainable parameters, one may consider the following scenario. Namely, let us be given a class of signals $\mathbf{x} \in \mathbb{R}^N$ that are not necessarily sparse themselves but sparsely representable with respect to a dictionary $\mathbf{\Phi}_o \in \mathbb{R}^{N\times N}$. In other words, for each $\mathbf{x}$, there is a sparse vector $\mathbf{z} \in \mathbb{R}^N$ such that $\mathbf{x} = \mathbf{\Phi}_o \mathbf{z}$. The dictionary $\mathbf{\Phi}_o$ is assumed to be unknown. It is possible to extend to overcomplete dictionaries, but we will stick to bases for the sake of simplicity.

We would like to learn a dictionary suitable for sparse reconstruction from a training sequence $\mathcal{S} = ((\mathbf{x}_i, \mathbf{y}_i))_{i=1,\dots,m}$ with i.i.d. samples drawn from an (unknown) distribution $\mathcal{D}$. Formally, this is a distribution over the $\mathbf{x}_i$, and then the corresponding measurements $\mathbf{y}_i$ are given by $\mathbf{y}_i = \mathbf{A}\mathbf{x}_i$, with $\mathbf{A}$ being fixed. We assume that the signals $\mathbf{x}$ in the class are bounded by a value, say $B_{\mathrm{in}}$, in the $\ell_2$-norm.

While taking the measurements $\mathbf{y} = \mathbf{A}\mathbf{x} =: \mathrm{enc}_{\mathbf{A}}(\mathbf{x})$ may be interpreted as *encoding* the signal $\mathbf{x}$ into $\mathbf{y}$, corresponding to a shallow, one-layer linear neural network (which is deterministic, when the measurement matrix $\mathbf{A}$ is considered to be fixed), the *decoder* is based on the unfolded version of the iterative soft-thresholding algorithm (ISTA) with $L$ iterations as follows. For a fixed stepsize $\tau > 0$, and a fixed $\lambda > 0$, the first layer is defined by $f_1(\mathbf{y}) = S_{\tau\lambda}(\tau(\mathbf{A}\mathbf{\Phi})^\top \mathbf{y})$. For the iteration (or layer, respectively) $l > 1$, the output is given by

$$\begin{aligned} f_l(\mathbf{z}) &= S_{\tau\lambda}\left[\mathbf{z} + \tau(\mathbf{A}\mathbf{\Phi})^\top(\mathbf{y} - (\mathbf{A}\mathbf{\Phi})\mathbf{z})\right] \\ &= S_{\tau\lambda}\left[\left(\mathbf{I} - \tau\mathbf{\Phi}^\top\mathbf{A}^\top\mathbf{A}\mathbf{\Phi}\right)\mathbf{z} + \tau(\mathbf{A}\mathbf{\Phi})^\top\mathbf{y}\right], \end{aligned} \tag{8.5}$$

which again can be interpreted as a layer of a neural network with weight matrix $\mathbf{I} - \tau\mathbf{\Phi}^\top\mathbf{A}^\top\mathbf{A}\mathbf{\Phi}$, bias $\tau(\mathbf{A}\mathbf{\Phi})^\top\mathbf{y}$ and activation function $S_{\tau\lambda}$, where the trainable parameters are the entries of $\mathbf{\Phi}$. Note that for $l > 1$, all $f_l$ coincide as functions

on $\mathbb{R}^N$. The index then refers to the iteration step or layer of the neural network, respectively. Then we denote the concatenation of $l$ such layers as $f^l_{\mathbf{\Phi}}$, i.e., for $\mathbf{\Phi}$ in every layer and given by

$$f^L_{\mathbf{\Phi}}(\mathbf{y}) = f_L \circ f_{L-1} \cdots \circ f_1(\mathbf{y}). \tag{8.6}$$

Note that, strictly speaking, the vector $\mathbf{y}$ will also be an input to the subsequent layers $f_2$, $f_3$, etc., but to simplify the notation, we do not write it explicitly after each layer. This point will not be of major importance for our derivations throughout this chapter.

For an actual reconstruction, we need to apply the dictionary $\mathbf{\Phi}$ again after the final layer. This means that a decoder (for a fixed number of layers $L$) is a neural network with shared weights

$$\mathrm{dec}^L_{\mathbf{\Phi}}(\mathbf{y}) = \mathbf{\Phi} f_L \circ f_{L-1} \cdots \circ f_1(\mathbf{y}) = \mathbf{\Phi} f^L_{\mathbf{\Phi}}(\mathbf{y}).$$

For technical reasons that will become apparent later in the proofs in Sect. 8.3, we will add an additional function $\sigma$ after the final layer. Different choices are possible here; we consider the choice

$$\sigma : \mathbb{R}^N \to \mathbb{R}^N, \qquad \mathbf{x} \mapsto \begin{cases} \mathbf{x} & \text{if } \|\mathbf{x}\|_2 \le B_{\text{out}}, \\ B_{\text{out}} \frac{\mathbf{x}}{\|\mathbf{x}\|_2} & \text{if } \|\mathbf{x}\|_2 > B_{\text{out}}, \end{cases} \tag{8.7}$$

with some fixed constant $B_{\text{out}}$. Obviously, this ensures $\|\sigma(\mathbf{x})\|_2 \le B_{\text{out}}$. Furthermore, note that $\sigma$ is norm-contractive and 1-Lipschitz, i.e.,

$$\|\sigma(\mathbf{x})\|_2 \le \|\mathbf{x}\|_2 \qquad \text{and} \qquad \|\sigma(\mathbf{x}_1) - \sigma(\mathbf{x}_2)\|_2 \le \|\mathbf{x}_1 - \mathbf{x}_2\|_2 \tag{8.8}$$

for any $\mathbf{x}$ and $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^N$. The role of $\sigma$ is to push the output of the network inside the $\ell_2$-ball of radius $B_{\text{out}}$, which in many applications is approximately known. The prior knowledge about the range of the outputs (boundedness) can improve the reconstruction performance and generalization [49]. The constant $B_{\text{out}}$ may be simply chosen to be equal to $B_{\text{in}}$.

To formulate this as a statistical learning problem, we will formally introduce a hypothesis class and a loss function. The hypothesis set consists of all functions that can be expressed as $L$-step soft thresholding, where the dictionary matrix $\mathbf{\Phi}$ parameterizes the hypothesis class, and with an additional $\sigma$ after the final layer added. That is,

$$\mathcal{H}^L_1 = \{\sigma \circ h : \mathbb{R}^n \to \mathbb{R}^N : h(\mathbf{y}) = \mathbf{\Phi} f^L_{\mathbf{\Phi}}(\mathbf{y}), \mathbf{\Phi} \in O(N)\}. \tag{8.9}$$

The assumption that $\mathbf{\Phi}$ ranges over the orthogonal group $O(N)$ and is shared across the layers leads to a recurrent neural network with a moderate number of weights. Using weight sharing enables a straightforward interpretation of learning

a dictionary for reconstruction. Much more general scenarios are discussed later, including models without weight sharing (or different degrees thereof), and models where also the threshold $\lambda$ and the stepsize $\tau$ may be trainable and even be altered from layer to layer.

Based on the training samples $\mathcal{S}$ and given the hypothesis space $\mathcal{H}_1^L$, a learning algorithm yields a function $h_{\mathcal{S}} \in \mathcal{H}_1^L$ that aims at reconstructing $\mathbf{x}$ from the measurements $\mathbf{y} = \mathbf{A}\mathbf{x}$. The empirical loss of a hypothesis $h$ is the reconstruction error on the training sequence, i.e., the difference between $\mathbf{x}_i$ and $\hat{\mathbf{x}}_i = h(\mathbf{y}_i)$, that is

$$\hat{\mathcal{L}}(h) = \frac{1}{m} \sum_{j=1}^{m} \ell(h, \mathbf{x}_j, \mathbf{y}_j).$$

Different choices for the loss function $\ell$ to measure the reconstruction error are possible. A typical choice is the *mean squared error* (MSE)

$$\ell_{\mathrm{MSE}}(h, \mathbf{x}, \mathbf{y}) = \|h(\mathbf{y}) - \mathbf{x}\|_2^2. \tag{8.10}$$

The true loss, i.e., the risk of a hypothesis $h$, is accordingly defined as follows:

$$\mathcal{L}(h) = \mathbb{E}_{\mathbf{x},\mathbf{y}\sim\mathcal{D}} \left(\ell(h, \mathbf{x}, \mathbf{y})\right).$$

The generalization error of the hypothesis $h_{\mathcal{S}}$ based on the training samples $\mathcal{S}$ is given as the difference between the empirical loss and the true loss,

$$\mathrm{GE}(h_{\mathcal{S}}) = \left| \hat{\mathcal{L}}(h_{\mathcal{S}}) - \mathcal{L}(h_{\mathcal{S}}) \right|.$$

Note that some references denote the true loss $\mathcal{L}(h_{\mathcal{S}})$ as the generalization error. However, the above definition is more convenient for our purposes.

This motivating example explains how iterative reconstruction algorithms such as ISTA can be unfolded as a neural network, which is then trained on some training data. By this transformation into a machine learning problem, this raises the question of generalization, i.e., how well the trained decoder works on unseen data (from the same distribution of interest). We will return to the problem of bounding the generalization error in the third section, after giving a more detailed overview on LISTA and its variants in the remainder of this section, and discussing different approaches to and challenges of generalization in deep learning in the next section.

### *8.2.2 Variants of LISTA*

Before moving to the generalization analysis, we review some of the variants of LISTA algorithms. The original paper [17] focuses on sparse coding applications where a sparse representation of data needs to be learned. Since $\ell_1$-based methods

are slow and do not scale to larger datasets, the authors propose a time unfolded version of ISTA with a fixed number of iterations. The first layer of the network is simply given by

$$\mathbf{x}^0 = S_\lambda \left[\mathbf{W}\mathbf{y}\right].$$

The next layers are formulated as

$$\mathbf{x}^t = S_\lambda \left[\mathbf{W}\mathbf{y} + \mathbf{S}\mathbf{x}^{t-1}\right].$$

LISTA learns $(\lambda, \mathbf{W}, \mathbf{S})$ by back-propagation. These parameters can be common or different across layers.

Following LISTA, many works explored the similar idea of unfolding iterative thresholding algorithms [30, 43, 49, 50, 52, 53]. For example, the authors of [49] address two problems of ISTA-type methods. First, the convergence of LISTA requires higher thresholds in the shrinkage operator to produce sparse vectors. This comes at the cost of shrinkage of the output with respect to the original vector. The authors in [49] introduce a gain gate to increase output values and compensate this effect. They additionally introduce an overshoot gate that tries to improve the convergence by learning to boost the estimated vector close to the ground truth.

As mentioned before, it is possible to change further the structure of LISTA and possibly improve the performance. For example, in analytic LISTA (ALISTA) [30], only thresholds and stepsize parameters are learned. The update rule (8.5) is modified into $f_l(\mathbf{z}) = S_{\lambda^{(l)}}\left[\mathbf{z} + \tau^{(l)}\mathbf{W}^\top(\mathbf{y} - \mathbf{A}\mathbf{x}^l)\right]$, where the matrix $\mathbf{W}$ is chosen without using data, namely as a minimizer of the coherence with respect to $\mathbf{A}$. Instead of learning the same stepsizes and thresholds for all the samples as in ALISTA, these parameters are updated based on the output of the previous layer in neurally augmented ALISTA [6].

## 8.3 Generalization of Deep Neural Networks

The generalization error of machine learning algorithms is the gap between their performance averaged over the samples of training data and the expected performance computed using the actual distribution. In this chapter, we define the generalization error as the absolute difference between these two losses.

A machine learning algorithm $\mathcal{A}$ returns a function $h : \mathcal{X} \to \mathcal{Y}$ from a set of choice functions called hypothesis class $\mathcal{H}$ based on the training data defined as $m$ i.i.d. samples $\mathbf{z}_i = (\mathbf{x}_i, y_i)$ from a distribution $\mathcal{D}$ on $\mathcal{X} \times \mathcal{Y}$. (Since $y_i$ may often refer to labels, we do not use a boldface notation.) If the hypothesis class is *large*, it may contain complex enough functions that match the training data perfectly with zero training error. These functions, however, do not necessarily generalize well to new, yet unseen data (or the test data in experiments). Statistical learning theory

aims at bounding the generalization error in terms of the complexity of hypothesis class and training set size. There are different notions of complexities available in the literature such as Vapnik–Chervonenkis (VC) dimension [47, 48], Rademacher complexity [4, 26], stability [38, 42], and robustness [51]. In this chapter, we focus on the Rademacher complexity framework for bounding the generalization error.

### 8.3.1 Rademacher Complexity Analysis

In order to bound the generalization error, we use the Rademacher complexity. Consider a class $\mathcal{G}$ of real-valued functions $g$. The empirical Rademacher complexity is defined as

$$\mathcal{R}_{\mathcal{S}}(\mathcal{G}) = \mathbb{E}_{\epsilon} \sup_{h \in \mathcal{G}} \frac{1}{m} \sum_{i=1}^{m} \epsilon_i g(\mathbf{x}_i), \tag{8.11}$$

where $\epsilon$ is a Rademacher vector, i.e., a vector of independent Rademacher variables $\epsilon_i$, $i = 1, \ldots, m$, taking the values $\pm 1$ with equal probability. The Rademacher complexity is then given as $\mathcal{R}_m(\mathcal{G}) = \mathbb{E}_{\mathcal{S} \sim \mathcal{D}^m} \mathcal{R}_{\mathcal{S}}(\mathcal{G})$. We will exclusively work with the empirical Rademacher complexity. The Rademacher complexity provides a complexity measure that can bound the generalization error. Suppose that the training samples are given by $\mathcal{S} = (\mathbf{z}_1, \ldots, \mathbf{z}_m)$, where $\mathbf{z}_i \in \mathcal{Z} = \mathcal{X} \times \mathcal{Y}$. The hypothesis class $\mathcal{H}$ consists of function $h : \mathcal{X} \to \mathcal{Y}$. Consider a loss function $\ell : \mathcal{H} \times Z \to \mathbb{R}$. The empirical loss of a function $h$ is defined by

$$\hat{\mathcal{L}}(h) = \frac{1}{m} \sum_{j=1}^{m} \ell(h, \mathbf{z}_j).$$

This is the performance of $h$ on the training data. We can write the true loss of $h$ as

$$\mathcal{L}(h) = \mathbb{E}_{z \sim \mathcal{D}} \left( \ell(h, \mathbf{z}) \right).$$

Given a loss function $\ell$ and a hypothesis class $\mathcal{H}$, we are interested in the Rademacher complexity of the class $\mathcal{G} = \ell \circ \mathcal{H} = \{g(\mathbf{z}) = \ell \circ h(\mathbf{z}) : h \in \mathcal{H}\}$. We rely on the following theorem that bounds the generalization error in terms of the empirical Rademacher complexity.

**Theorem 8.1 ([41, Theorem 26.5])** *Let $\mathcal{H}$ be a family of functions, $\mathcal{S}$ the training set drawn from $\mathcal{D}^n$, and $\ell$ a real-valued bounded loss function satisfying $|\ell(h, z)| \leq c$ for all $h \in \mathcal{H}$, $z \in Z$. Then, for $\delta \in (0, 1)$, with probability at least $1 - \delta$, we have, for all $h \in \mathcal{S}$,*

$$\mathcal{L}(h) \leq \hat{\mathcal{L}}(h) + 2\mathcal{R}_{\mathcal{S}}(\ell \circ \mathcal{H}) + 4c\sqrt{\frac{2\log(4/\delta)}{m}}. \tag{8.12}$$

For real-valued functions $h$, i.e., when $\mathcal{Y} = \mathbb{R}$, the Rademacher complexity of $\ell \circ \mathcal{H}$ can be bounded using the so-called contraction lemma [41, Lemma 26.9].

**Lemma 8.1 (Contraction Lemma)** *Let $\mathcal{S}$ be the training sequence and the functions $f_i$ be $K$-Lipschitz from $\mathbb{R}$ to $\mathbb{R}$ for $i \in [m]$. Then, for a class of real-valued functions $\mathcal{H}$, we have*

$$\mathbb{E}_\epsilon \sup_{h\in\mathcal{H}} \sum_{i=1}^{m} \epsilon_i f_i \circ h(\mathbf{x}_i) \leq K\mathbb{E}_\epsilon \sup_{h\in\mathcal{H}} \sum_{i=1}^{m} \epsilon_i h(\mathbf{x}_i). \tag{8.13}$$

With the contraction lemma, we can remove the loss function and work only with the hypothesis class.

### 8.3.2 Generalization Bounds for Deep Neural Networks

Many recent works aim at explaining the excellent generalization properties of deep neural networks. In order to provide a brief review of this body of literature, we consider an $L$-layer neural network

$$f_{\mathbf{W}_1,\ldots,\mathbf{W}_L}(\mathbf{y}) = \sigma(\mathbf{W}_L \cdot \sigma(\cdots \sigma(\mathbf{W}_1\mathbf{y})\cdots))$$

with weight matrices $\mathbf{W}_j \in \mathbb{R}^{n_{j-1}\times n_j}$, $j = 1, \ldots, L$ ($n_0 = n$), and an elementwise activation function $\sigma : \mathbb{R} \to \mathbb{R}$. All the works to be mentioned below consider the matrices $\mathbf{W}_j$ as free parameters of the hypothesis class; hence, they aim at an overparameterized setting. Moreover, they consider classification problems. In contrast, our work considers regression problems and networks with shared weights, leading to a non-overparameterized setting.

The Rademacher complexity was used in [5] to obtain norm-based generalization error bounds for the probability of misclassification via the argmax of a neural network in a multiclass problem with $K$ classes. The margin-type bound in [5] states that, with probability at least $1 - \delta$ over the i.i.d. samples $(\mathbf{x}_i, \mathbf{y}_i) \in \mathbb{R}^N \times [K]$,

$$\mathbb{P}(\operatorname{argmax} f_{\mathbf{W}_1,\ldots,\mathbf{W}_L}(\mathbf{x}) \neq \mathbf{y}) \leq \widehat{\mathcal{R}}_\gamma(f_{\mathbf{W}_1,\ldots,\mathbf{W}_L})$$
$$+ C\frac{\|\mathbf{X}\|_F \log(\max_j n_j)}{\gamma m}\left(\sum_{i=\ell}^{L}\left(\frac{\|\mathbf{W}_\ell^T\|_{2,1}}{\|\mathbf{W}_\ell\|_{2\to 2}}\right)^{2/3}\right)^{3/2}\prod_{\ell=1}^{L}(\rho\|\mathbf{W}_\ell\|_{2\to 2}) + \sqrt{\frac{C\log(1/\delta)}{m}},$$

where $\gamma > 0$ is a suitable parameter, $\rho > 0$ the Lipschitz constant of $\sigma$, and $\|\mathbf{W}_\ell^T\|_{2,1}$ is the sum of $\ell_2$-norm of rows of $\mathbf{W}_\ell^T$. Furthermore, the empirical margin-type risk is defined as

$$\widehat{\mathcal{R}}_\gamma(f) = \frac{1}{m}\sum_{i=1}^{m} \mathbf{1}[f(\mathbf{x}_i)_{\mathbf{y}_i} \leq \gamma + \max_{j \neq \mathbf{y}_i} f(\mathbf{x}_i)_j].$$

Noting that in general $\|\mathbf{W}_\ell^T\|_{2,1} = \|\mathbf{W}_\ell\|_{2\to 1} \leq \sqrt{n_\ell}\|\mathbf{W}_\ell\|_{2\to 2}$, and assuming that all data points $\mathbf{x}_i$ are bounded in $\ell_2$, i.e., $\|\mathbf{X}\|_F \leq \sqrt{m}B$, the first term in the second line of the bound can be estimated by

$$L^{3/2}\max_j \sqrt{n_j}\log(n_j)\left(\rho \max_\ell \|\mathbf{W}_\ell\|_{2\to 2}\right)^L \frac{B}{\gamma\sqrt{m}}.$$

A similar, but slightly worse, norm-based bound was obtained [36] using a PAC Bayesian approach, which leads to a completely different analysis.

A bound with potentially better dimension dependence was obtained in [15]. As an example, the result of [5] can be improved to a generalization error bound scaling in the following way, for any $p \geq 1$,

$$\tilde{O}\left(\prod_{\ell=1}^{L} M(\ell)\min\left\{\frac{\log\left(\frac{1}{\Gamma}\prod_{\ell=1}^{L} M_p(\ell)\right)^{\frac{1}{p+\frac{2}{3}}}}{m^{\frac{1}{2+3p}}}, \sqrt{\frac{L^3}{m}}\right\}\right), \tag{8.14}$$

where $M(\ell)$ and $M_p(\ell)$ are, respectively, upper bounds on $\|\mathbf{W}_\ell\|_{2\to 2}$ and $p$-Schatten norm of $\mathbf{W}_\ell$, and $\Gamma$ is a lower bound on $\prod_{\ell=1}^{L}\|\mathbf{W}_\ell\|_{2\to 2}$. This result can remove the dependence on $L$ if the norms are well behaved. Note that this comes at the price of a worse sample efficiency. For instance, for $p = 1$, the dimension-free bound scales as $m^{1/5}$ in contrast with $m^{1/2}$ (however, note that a minimum with $\sqrt{L^3/m}$ is taken).

Instead of continuing the discussion on the existing generalization bounds for deep networks, we invite the interested reader to refer to [21] for a detailed experimental account and [33] for some shortcomings of existing bounds, for example, they tend to grow with training data size.

In the remainder of this chapter, we will show a generalization error bound for regression (reconstruction) with the introduced thresholding networks that shows linear dimension dependence and linear dependence on the number of layers, see Theorem 8.2. Our proof uses different techniques than in the works mentioned above. In fact, it is not straightforward to apply those techniques due to the weight sharing between different layers in our case.

## 8.4 Generalization of Deep Thresholding Networks

In this section, we return to the original problem introduced already in the beginning of this chapter. We will prove the following result on the generalization error of the class of neural networks $\mathcal{H}_1$ introduced in Sect. 8.2.1 with a learned orthogonal dictionary. We state our theorem here under the simplifying but reasonable assumption that $\tau\|\mathbf{A}\|_2^2 \leq 1$, see (8.4). A more general version of the result will be presented in Sect. 8.4.6. We continue to use the notation introduced in Sect. 8.2.1.

**Theorem 8.2** *Consider the hypothesis space $\mathcal{H}_1^L$ defined in (8.9) and assume the samples $\mathbf{x}_i$, $i = 1, \ldots, m$, to be drawn i.i.d. at random according to some (unknown) distribution such that $\|\mathbf{x}_i\|_2 \leq B_{\text{in}}$ almost surely with $B_{\text{in}} = B_{\text{out}}$ in (8.7). Let $\mathbf{y}_i = \mathbf{A}\mathbf{x}_i$, and assume that $\tau\|\mathbf{A}\|_{2\to 2}^2 \leq 1$. Then with probability at least $1-\delta$, for all $h \in \mathcal{H}_1^L$, the generalization error is bounded as*

$$\mathcal{L}(h) \leq \hat{\mathcal{L}}(h) + 8B_{\text{out}}\sqrt{\frac{Nn\log(2+8L(L+3))}{m}} + 8B_{\text{out}}\frac{N\sqrt{\log(e+8eL)}}{\sqrt{m}} + B_{\text{out}}\sqrt{\frac{128\log(4/\delta)}{m}}. \tag{8.15}$$

Further details and a slightly simplified bound for $L \geq 2$ can be found in Corollary 8.3 and the discussion thereafter.

Of course, the idea is to choose an $h$ that minimizes the empirical loss $\hat{\mathcal{L}}(h)$, i.e., the first term on the right-hand side of (8.15), but in principle any $h$ (computed by some algorithm) can be inserted into this bound. Since the samples are available, both $\hat{\mathcal{L}}(h)$ and the other terms can be computed (assuming $B_{\text{in}}$ is known), so that the theorem allows to provide a concrete bound of the true risk $\mathcal{L}(h)$. Roughly speaking, i.e., ignoring constants, the generalization error can be bounded as

$$|\mathcal{L}(h) - \hat{\mathcal{L}}(h)| \lesssim \sqrt{\frac{Nn\log(L) + N^2\log(L)}{m}}. \tag{8.16}$$

In other words, once the number of training samples scales like $m \sim (Nn + N^2)\log(L)$, the generalization error is guaranteed to be small with high probability.

Remarkably, the number $L$ of layers only enters logarithmically, while some of the previously available bounds for deep neural networks (in the context of classification, however) scale even only exponentially with $L$ (at least in many interesting settings).

The remainder of this section is devoted to the proof of the above statement. We will use the approach based on the Rademacher complexity as described in Sect. 8.3.1, in particular Theorem 8.1. Hence, we need to estimate the Rademacher complexity

$$\mathcal{R}_m(\ell \circ \mathcal{H}_1^L) = \mathbb{E} \sup_{h \in \mathcal{H}_1^L} \frac{1}{m} \sum_{i=1}^{m} \epsilon_i \|\sigma(h(\mathbf{y}_i) - \mathbf{x}_i)\|_2. \tag{8.17}$$

As explained in Sect. 8.3.1, the so-called contraction principle is often applied in such situations. However, since we are dealing with a hypothesis class of vector-valued functions, it is not applicable in its standard form. The following result [31, Corollary 4] is a generalization to this situation, and it is a crucial tool for our proof.

**Lemma 8.2** *Suppose that $\mathcal{H}$ is a set of functions $h : \mathcal{X} \to \mathbb{R}^N$ and that $f : \mathbb{R}^N \to \mathbb{R}^N$ is $K$-Lipschitz. Let $\mathcal{S} = (\mathbf{x}_i)_{i \in [m]}$ be the training sequence. Then*

$$\mathbb{E} \sup_{h \in \mathcal{H}} \sum_{i=1}^{m} \epsilon_i f \circ h(\mathbf{x}_i) \leq \sqrt{2} K \mathbb{E} \sup_{h \in \mathcal{H}} \sum_{i=1}^{m} \sum_{k=1}^{N} \epsilon_{ik} h_k(\mathbf{x}_i), \tag{8.18}$$

*where $(\epsilon_i)$ and $(\epsilon_{ik})$ are both Rademacher sequences.*

As both the $\ell_2$-norm and the function $\sigma$ (the latter by assumption) are 1-Lipschitz, applying Lemma 8.2 yields

$$\mathcal{R}_m(\ell \circ \mathcal{H}^L) \leq \sqrt{2} \mathbb{E} \sup_{h \in \mathcal{H}^L} \frac{1}{m} \sum_{i=1}^{m} \sum_{k=1}^{N} \epsilon_{ik} h_k(\mathbf{x}_i), \tag{8.19}$$

where $\mathcal{H}^L$ denotes either $\mathcal{H}_1^L$ or the hypothesis class $\mathcal{H}_2^L$ to be defined in the next section. In order to derive a bound for the Rademacher complexity, we use chaining techniques. Roughly speaking, this refers to bounding the expectation of a stochastic process by geometric properties of its index set (covering numbers at different scales), equipped with an appropriate norm (or metric). We briefly provide the necessary results in the next section; for a more detailed introduction to the topic, we refer the reader to [28, 45].

### 8.4.1 Boundedness: Assumptions and Results

For technical reasons that will become apparent, we will introduce a separate dictionary for the linear transformation after the very final layer and consider the enlarged hypothesis class

$$\mathcal{H}_2^L = \{\sigma \circ h : \mathbb{R}^n \to \mathbb{R}^N : h(\mathbf{y}) = \boldsymbol{\Psi} f_{\boldsymbol{\Phi}}^L(\mathbf{y}), \boldsymbol{\Psi}, \boldsymbol{\Phi} \in O(N)\}. \tag{8.20}$$

In order to apply Theorem 8.1, the loss function needs to be bounded. Therefore, and as commonly done in the machine learning literature, we assume (as already mentioned) that the input is bounded in the $\ell_2$-norm by some constant $B_{\text{in}}$, i.e.,

$$\|\mathbf{x}\|_2 \leq B_{\text{in}}. \tag{8.21}$$

Furthermore, let us recall from (8.8) that the function $\sigma$ is bounded by $B_{\text{out}}$. In particular, this means that every $h \in \mathcal{H}_2^L$ (analogously for $\mathcal{H}_1^L$) is also bounded by

$$\|h(\mathbf{y})\|_2 = \left\|\sigma\left(\mathbf{\Psi} f_{\mathbf{\Phi}}^L(\mathbf{y})\right)\right\|_2 \leq B_{\text{out}} \tag{8.22}$$

independently of $\mathbf{\Psi}$ and $\mathbf{\Phi}$. By passing to the matrix notation (i.e., considering the matrix $\mathbf{Y}$ collecting all measurements, instead of a single measurement $\mathbf{y}$), we obtain the similar estimate

$$\|h(\mathbf{Y})\|_F \leq \sqrt{m} B_{\text{out}}, \tag{8.23}$$

where the additional term of $\sqrt{m}$ takes the number of training points into account. By combining (8.21) and (8.22), we find that the loss function is bounded by

$$\begin{aligned} \ell(h, \mathbf{y}, \mathbf{x}) &= \|h(\mathbf{y}) - \mathbf{x}\|_2 \leq \|\mathbf{x}\|_2 + \|h(\mathbf{y})\|_2 \\ &\leq B_{\text{in}} + B_{\text{out}}, \end{aligned} \tag{8.24}$$

so that $B_{\text{in}} + B_{\text{out}}$ plays the role of $c$ in Theorem 8.1. Besides these boundedness assumptions, we can also upper bound the output $f_{\mathbf{\Phi}}^l(\mathbf{Y})$ with respect to the Frobenius norm after any number of layers $l$ (in particular for $l < L$, when the layer is not directly followed by an application of the $\sigma$ function) as follows. This will be used later in the main technical result, Theorem 8.5.

**Lemma 8.3** *For any $\mathbf{\Phi} \in O(N)$, $l \in \mathbb{N}$, and arbitrary $\tau, \lambda > 0$ in $S_{\tau\lambda}$ in the definition (8.3) of $f_{\mathbf{\Phi}}^l$, we have*

$$\left\|f_{\mathbf{\Phi}}^l(\mathbf{Y})\right\|_F \leq \left\|\tau(\mathbf{A}\mathbf{\Phi})^\top \mathbf{Y}\right\|_F \sum_{k=0}^{l-1} \left\|\mathbf{I} - \tau \mathbf{\Phi}^\top \mathbf{A}^\top \mathbf{A} \mathbf{\Phi}\right\|_{2\to 2}^k \tag{8.25}$$

$$\leq \tau \|\mathbf{A}\|_{2\to 2} \|\mathbf{Y}\|_F \sum_{k=0}^{l-1} \left\|\mathbf{I} - \tau \mathbf{A}^\top \mathbf{A}\right\|_{2\to 2}^k. \tag{8.26}$$

We will encounter the expression $\|\mathbf{I} - \tau \mathbf{A}^\top \mathbf{A}\|_{2\to 2}$ more often in the sequel. The following remark is useful and shows it can be easily bounded under realistic assumptions. In particular, we can use it to simplify the above estimate to obtain for arbitrary $\mathbf{\Psi}, \mathbf{\Phi} \in O(N)$. Namely, under the condition of $\tau \|\mathbf{A}\|_{2\to 2}^2 \leq 1$ and assuming $\mathbf{y}_i = \mathbf{A}(\mathbf{x}_i)$, we have

$$\begin{aligned} \left\|\mathbf{\Psi} f_{\mathbf{\Phi}}^L(\mathbf{Y})\right\|_2 &= \left\|f_{\mathbf{\Phi}}^L(\mathbf{Y})\right\|_2 \leq L\tau \|\mathbf{A}\|_{2\to 2} \|\mathbf{Y}\|_F = L\tau \|\mathbf{A}\|_{2\to 2} \|\mathbf{A}\mathbf{X}\|_F \\ &\leq L \|\mathbf{X}\|_F \leq L\sqrt{m} B_{\text{in}}, \end{aligned} \tag{8.27}$$

i.e., a linear growth with $L$. Note that this is a worst-case bound, and might possibly be improved under additional assumptions.

*Remark 8.1* Assume $\tau \|\mathbf{A}\|_{2\to 2}^2 \leq 1$. Then $\|\mathbf{I} - \tau \mathbf{A}^\top \mathbf{A}\|_{2\to 2} \leq 1$. In the compressive sensing setup ($n < N$), the $N \times N$—matrix $\mathbf{A}^\top \mathbf{A}$ is rank-deficient so that even $\|\mathbf{I} - \tau \mathbf{A}^\top \mathbf{A}\|_{2\to 2} = 1$ holds in this case.

***Proof*** Note that the second inequality (8.26) immediately follows from (8.25) due to the orthogonality of $\mathbf{\Phi}$. We will prove (8.25) via induction. Clearly, for $l = 1$, we have $\left\| f_{\mathbf{\Phi}}^1(\mathbf{Y}) \right\|_F = \left\| \tau(\mathbf{A}\mathbf{\Phi})^\top \mathbf{Y} \right\|_F$. Assuming the statement is true for $l$, we obtain it for $l+1$ by the following chain of inequalities, using in particular the contractivity $S_{\tau\lambda}$ with respect to the Frobenius norm,

$$\begin{aligned}
\left\| f_{\mathbf{\Phi}}^{l+1}(\mathbf{Y}) \right\|_F &= \left\| S_{\tau\lambda} \left[ \left( \mathbf{I} - \tau \mathbf{\Phi}^\top \mathbf{A}^\top \mathbf{A} \mathbf{\Phi} \right) f_{\mathbf{\Phi}}^l(\mathbf{Y}) + \tau(\mathbf{A}\mathbf{\Phi})^\top \mathbf{Y} \right] \right\|_F \\
&\leq \left\| \left( \mathbf{I} - \tau \mathbf{\Phi}^\top \mathbf{A}^\top \mathbf{A} \mathbf{\Phi} \right) f_{\mathbf{\Phi}}^l(\mathbf{Y}) \|_F + \| \tau(\mathbf{A}\mathbf{\Phi})^\top \mathbf{Y} \right\|_F \\
&\leq \left\| \mathbf{I} - \tau \mathbf{\Phi}^\top \mathbf{A}^\top \mathbf{A} \mathbf{\Phi} \right\|_{2\to 2} \left\| f_{\mathbf{\Phi}}^l(\mathbf{Y}) \right\|_F + \left\| \tau(\mathbf{A}\mathbf{\Phi})^\top \mathbf{Y} \right\|_F \\
&\leq \left\| \tau(\mathbf{A}\mathbf{\Phi})^\top \mathbf{Y} \right\|_F \left( \sum_{k=0}^{l-1} \left\| \mathbf{I} - \tau \mathbf{\Phi}^\top \mathbf{A}^\top \mathbf{A} \mathbf{\Phi} \right\|_{2\to 2}^{k+1} \right) + \left\| \tau(\mathbf{A}\mathbf{\Phi})^\top \mathbf{Y} \right\|_F \\
&= \left\| \tau(\mathbf{A}\mathbf{\Phi})^\top \mathbf{Y} \right\|_F \sum_{k=0}^{l} \left\| \mathbf{I} - \tau \mathbf{\Phi}^\top \mathbf{A}^\top \mathbf{A} \mathbf{\Phi} \right\|_{2\to 2}^{k},
\end{aligned}$$

where we have used the induction hypothesis to arrive at the fourth line. □

### 8.4.2 Dudley's Inequality

We use the following version of Dudley's inequality [12, Theorem 8.23]. To state the theorem, we require additional definitions. Consider a stochastic process $(X_t)_{t\in\mathcal{T}}$ with the index set $\mathcal{T}$ in a space with pseudo-metric $d$ given by

$$d(s,t) = \left( \mathbb{E}|X_s - X_t|^2 \right)^{1/2}.$$

A zero-mean process $X_t$ for $t \in \mathcal{T}$ is called *sub-Gaussian*, if

$$\mathbb{E} \exp(\theta(X_s - X_t)) \leq \exp\left( \theta^2 d(s,t)^2/2 \right) \quad \forall s, t \in \mathcal{T},\ \theta > 0.$$

Finally, define the radius of $\mathcal{T}$ as $\Delta(\mathcal{T}) = \sup_{t\in\mathcal{T}} \sqrt{\mathbb{E}|X_t|^2}$. Dudley's inequality, which will be used to bound the Rademacher complexity term, is stated as follows.

**Theorem 8.3 (Dudley's Inequality)** *Let $(X_t)_{t\in\mathcal{T}}$ be a centered (i.e., $\mathbb{E}X_t = 0$ for every $t \in \mathcal{T}$) sub-Gaussian process with radius $\Delta(\mathcal{T})$. Then,*

$$\mathbb{E}\sup_{t\in\mathcal{T}} X_t \leq 4\sqrt{2}\int_0^{\Delta(\mathcal{T})/2} \sqrt{\log \mathcal{N}(\mathcal{T}, d, u)}\,\mathrm{d}u. \tag{8.28}$$

### 8.4.3 Bounding the Rademacher Complexity

Recalling our hypothesis spaces introduced above, obviously $\mathcal{H}_1^L$ is embedded in $\mathcal{H}_2^L$, i.e., we have the inclusion

$$\mathcal{H}_1^L \subset \mathcal{H}_2^L. \tag{8.29}$$

For a fixed number of layers $L \in \mathbb{N}$ and $i = 1, 2$, define the set $\mathcal{M}_i \subset \mathbb{R}^{N\times m}$ as follows:

$$\mathcal{M}_i = \left\{(h(\mathbf{y}_1)|\ldots|h(\mathbf{y}_m)) \in \mathbb{R}^{N\times m} : h \in \mathcal{H}_i^L\right\}. \tag{8.30}$$

For the case $i = 2$, the set $\mathcal{M}_2$ corresponding to the hypothesis space $\mathcal{H}_2^L$ reads as

$$\mathcal{M}_2 = \left\{\sigma\left(\mathbf{\Psi} f_{\mathbf{\Phi}}^L(\mathbf{Y})\right) \in \mathbb{R}^{N\times m} : \mathbf{\Psi}, \mathbf{\Phi} \in O(N)\right\}. \tag{8.31}$$

Note that $\mathcal{M}_2$ is parameterized by $\mathbf{\Psi}, \mathbf{\Phi} \in O(N)$ (as the hypothesis space $\mathcal{H}_2^L$ is), such that we can rewrite (8.19) as

$$\mathcal{R}_m(\ell \circ \mathcal{H}_2^L) \leq \mathbb{E}\sup_{\mathbf{M}\in\mathcal{M}_2} \frac{1}{m}\sum_{i=1}^{m}\sum_{k=1}^{N} \epsilon_{ik} M_{ik}. \tag{8.32}$$

We use Dudley's inequality and a covering number argument to bound the Rademacher complexity term. The Rademacher process defined in (8.32) is a sub-Gaussian process, and therefore, we can apply Dudley's inequality. For the set of matrices $\mathcal{M}_2$ defined above, the radius can be estimated as

$$\begin{aligned}\Delta(\mathcal{M}_2) &= \sup_{h\in\mathcal{H}_2^L} \sqrt{\mathbb{E}\left(\sum_{i=1}^{m}\sum_{k=1}^{N} \epsilon_{ik} h_k(\mathbf{y}_i)\right)^2} \leq \sup_{h\in\mathcal{H}_2^L} \sqrt{\mathbb{E}\sum_{i=1}^{m}\sum_{k=1}^{N} (h_k(\mathbf{y}_i))^2} \\ &\leq \sup_{h\in\mathcal{H}_2^L} \sqrt{\sum_{i=1}^{m} \|h(\mathbf{y}_i)\|^2} \leq \sqrt{m} B_{\text{out}},\end{aligned}$$

where the last inequality has already been stated in (8.23). Plugging this bound in Dudley's inequality, we obtain the following upper bound for the Rademacher complexity,

$$\mathcal{R}_m(\ell \circ \mathcal{H}_2^L) \leq \frac{4\sqrt{2}}{m} \int_0^{\sqrt{m}B_{\text{out}}/2} \sqrt{\log \mathcal{N}(\mathcal{M}_2, \|\cdot\|_F, \epsilon)}\, \mathrm{d}\epsilon. \tag{8.33}$$

We only need to find the covering numbers inside the integral. For that, we bound the covering number of the hypothesis classes by the covering number of its parameter spaces. This is done using a perturbation analysis argument.

### 8.4.4 A Perturbation Result

The following theorem relates the effect of perturbation of the parameters on the function outputs. This result will be used to bound their covering numbers.

**Theorem 8.4** *Consider the functions $f_{\mathbf{\Phi}}^l$ defined as in (8.6) with $L \geq 2$ and dictionary $\mathbf{\Phi}$ in $O(N)$. Then, for any $\mathbf{\Phi}_1, \mathbf{\Phi}_2 \in O(N)$, we have*

$$\left\| f_{\mathbf{\Phi}_1}^L(\mathbf{Y}) - f_{\mathbf{\Phi}_2}^L(\mathbf{Y}) \right\|_F \leq K_L \|\mathbf{A}\mathbf{\Phi}_1 - \mathbf{A}\mathbf{\Phi}_2\|_{2\to 2}, \tag{8.34}$$

*where $K_L$ is given by*

$$\begin{aligned} K_L = {} & \tau \|\mathbf{Y}\|_F \|\mathbf{I} - \tau \mathbf{A}^\top \mathbf{A}\|_{2\to 2}^{L-1} \\ & + \tau \|\mathbf{Y}\|_F \sum_{l=2}^{L} \|\mathbf{I} - \tau \mathbf{A}^\top \mathbf{A}\|_{2\to 2}^{L-l} \left( 1 + 2\tau \|\mathbf{A}\|_{2\to 2}^2 \sum_{k=0}^{l-2} \|\mathbf{I} - \tau \mathbf{A}^\top \mathbf{A}\|_{2\to 2}^{k} \right). \end{aligned} \tag{8.35}$$

*If $\tau \|\mathbf{A}\|_{2\to 2}^2 \leq 1$, we have the simplified upper bound*

$$K_L \leq \tau \|\mathbf{Y}\|_F L(L+3). \tag{8.36}$$

The bound (8.36) follows from the observation in Remark 8.1.

***Proof*** We formally set $f_{\mathbf{\Phi}_1}^0(\mathbf{Y}) = f_{\mathbf{\Phi}_2}^0(\mathbf{Y}) = \mathbf{Y}$ for a unified treatment of all layers $l \geq 1$. Using the fact that $S_{\tau\lambda}$ is 1-Lipschitz, we obtain

$$\begin{aligned} & \left\| f_{\mathbf{\Phi}_1}^l(\mathbf{Y}) - f_{\mathbf{\Phi}_2}^l(\mathbf{Y}) \right\|_F \\ \leq {} & \left\| \left( \mathbf{I} - \tau (\mathbf{A}\mathbf{\Phi}_1)^\top \mathbf{A}\mathbf{\Phi}_1 \right) f_{\mathbf{\Phi}_1}^{l-1}(\mathbf{Y}) + \tau (\mathbf{A}\mathbf{\Phi}_1)^\top \mathbf{Y} \right. \end{aligned}$$

$$
\begin{aligned}
&\quad - \left(\mathbf{I} - \tau(\mathbf{A}\boldsymbol{\Phi}_2)^\top \mathbf{A}\boldsymbol{\Phi}_2\right) f_{\boldsymbol{\Phi}_2}^{l-1}(\mathbf{Y}) - \tau(\mathbf{A}\boldsymbol{\Phi}_2)^\top \mathbf{Y}\Big\|_F \\
&\le \left\|\left(\mathbf{I} - \tau(\mathbf{A}\boldsymbol{\Phi}_1)^\top \mathbf{A}\boldsymbol{\Phi}_1\right) f_{\boldsymbol{\Phi}_1}^{l-1}(\mathbf{Y}) - \left(\mathbf{I} - \tau(\mathbf{A}\boldsymbol{\Phi}_2)^\top \mathbf{A}\boldsymbol{\Phi}_2\right) f_{\boldsymbol{\Phi}_2}^{l-1}(\mathbf{Y})\right\|_F \\
&\quad + \left\|\tau(\mathbf{A}\boldsymbol{\Phi}_1)^\top \mathbf{Y} - \tau(\mathbf{A}\boldsymbol{\Phi}_2)^\top \mathbf{Y}\right\|_F \\
&\le \left\|\left(\mathbf{I} - \tau(\mathbf{A}\boldsymbol{\Phi}_1)^\top \mathbf{A}\boldsymbol{\Phi}_1\right) f_{\boldsymbol{\Phi}_1}^{l-1}(\mathbf{Y}) - \left(\mathbf{I} - \tau(\mathbf{A}\boldsymbol{\Phi}_2)^\top \mathbf{A}\boldsymbol{\Phi}_2\right) f_{\boldsymbol{\Phi}_2}^{l-1}(\mathbf{Y})\right\|_F \quad (8.37) \\
&\quad + 2\tau \left\|\mathbf{Y}\right\|_F \left\|\mathbf{A}\boldsymbol{\Phi}_1 - \mathbf{A}\boldsymbol{\Phi}_2\right\|_{2\to 2}.
\end{aligned}
$$

The term (8.37) is estimated further as follows:

$$
\begin{aligned}
&\left\|\left(\mathbf{I} - \tau(\mathbf{A}\boldsymbol{\Phi}_1)^\top \mathbf{A}\boldsymbol{\Phi}_1\right) f_{\boldsymbol{\Phi}_1}^{l-1}(\mathbf{Y}) - \left(\mathbf{I} - \tau(\mathbf{A}\boldsymbol{\Phi}_2)^\top \mathbf{A}\boldsymbol{\Phi}_2\right) f_{\boldsymbol{\Phi}_2}^{l-1}(\mathbf{Y})\right\|_F \\
&\le \Big\|\left(\mathbf{I} - \tau(\mathbf{A}\boldsymbol{\Phi}_1)^\top \mathbf{A}\boldsymbol{\Phi}_1\right) f_{\boldsymbol{\Phi}_1}^{l-1}(\mathbf{Y}) - \left(\mathbf{I} - \tau(\mathbf{A}\boldsymbol{\Phi}_1)^\top \mathbf{A}\boldsymbol{\Phi}_2\right) f_{\boldsymbol{\Phi}_1}^{l-1}(\mathbf{Y}) \\
&\quad + \left(\mathbf{I} - \tau(\mathbf{A}\boldsymbol{\Phi}_1)^\top \mathbf{A}\boldsymbol{\Phi}_2\right) f_{\boldsymbol{\Phi}_1}^{l-1}(\mathbf{Y}) - \left(\mathbf{I} - \tau(\mathbf{A}\boldsymbol{\Phi}_2)^\top \mathbf{A}\boldsymbol{\Phi}_2\right) f_{\boldsymbol{\Phi}_1}^{l-1}(\mathbf{Y})\Big\|_F \\
&\quad + \left\|\left(\mathbf{I} - \tau(\mathbf{A}\boldsymbol{\Phi}_2)^\top \mathbf{A}\boldsymbol{\Phi}_2\right) f_{\boldsymbol{\Phi}_1}^{l-1}(\mathbf{Y}) - \left(\mathbf{I} - \tau(\mathbf{A}\boldsymbol{\Phi}_2)^\top \mathbf{A}\boldsymbol{\Phi}_2\right) f_{\boldsymbol{\Phi}_2}^{l-1}(\mathbf{Y})\right\|_F \\
&\le \Big\|\left(\mathbf{I} - \tau(\mathbf{A}\boldsymbol{\Phi}_1)^\top \mathbf{A}\boldsymbol{\Phi}_1\right) f_{\boldsymbol{\Phi}_1}^{l-1}(\mathbf{Y}) - \left(\mathbf{I} - \tau(\mathbf{A}\boldsymbol{\Phi}_1)^\top \mathbf{A}\boldsymbol{\Phi}_2\right) f_{\boldsymbol{\Phi}_1}^{l-1}(\mathbf{Y}) \\
&\quad + \left(\mathbf{I} - \tau(\mathbf{A}\boldsymbol{\Phi}_1)^\top \mathbf{A}\boldsymbol{\Phi}_2\right) f_{\boldsymbol{\Phi}_1}^{l-1}(\mathbf{Y}) - \left(\mathbf{I} - \tau(\mathbf{A}\boldsymbol{\Phi}_2)^\top \mathbf{A}\boldsymbol{\Phi}_2\right) f_{\boldsymbol{\Phi}_1}^{l-1}(\mathbf{Y}) \\
&\quad + \left(\mathbf{I} - \tau(\mathbf{A}\boldsymbol{\Phi}_2)^\top \mathbf{A}\boldsymbol{\Phi}_2\right)\left(f_{\boldsymbol{\Phi}_1}^{l-1}(\mathbf{Y}) - f_{\boldsymbol{\Phi}_2}^{l-1}(\mathbf{Y})\right)\Big\|_F \\
&\le \Big\|\tau(\mathbf{A}\boldsymbol{\Phi}_1)^\top \mathbf{A}\boldsymbol{\Phi}_1 f_{\boldsymbol{\Phi}_1}^{l-1}(\mathbf{Y}) - \tau(\mathbf{A}\boldsymbol{\Phi}_1)^\top \mathbf{A}\boldsymbol{\Phi}_2 f_{\boldsymbol{\Phi}_1}^{l-1}(\mathbf{Y}) \\
&\quad + \tau(\mathbf{A}\boldsymbol{\Phi}_1)^\top \mathbf{A}\boldsymbol{\Phi}_2 f_{\boldsymbol{\Phi}_1}^{l-1}(\mathbf{Y}) - \tau(\mathbf{A}\boldsymbol{\Phi}_2)^\top \mathbf{A}\boldsymbol{\Phi}_2 f_{\boldsymbol{\Phi}_1}^{l-1}(\mathbf{Y})\Big\|_F \\
&\quad + \left\|\left(\mathbf{I} - \tau(\mathbf{A}\boldsymbol{\Phi}_2)^\top \mathbf{A}\boldsymbol{\Phi}_2\right)\right\|_{2\to 2} \left\|f_{\boldsymbol{\Phi}_1}^{l-1}(\mathbf{Y}) - f_{\boldsymbol{\Phi}_2}^{l-1}(\mathbf{Y})\right\|_F \\
&\le \left\|\tau(\mathbf{A}\boldsymbol{\Phi}_1)^\top\right\|_{2\to 2} \left\|(\mathbf{A}\boldsymbol{\Phi}_1 - \mathbf{A}\boldsymbol{\Phi}_2) f_{\boldsymbol{\Phi}_1}^{l-1}(\mathbf{Y})\right\|_F \\
&\quad + \tau \left\|(\mathbf{A}\boldsymbol{\Phi}_1)^\top - (\mathbf{A}\boldsymbol{\Phi}_2)^\top\right\|_{2\to 2} \left\|\mathbf{A}\boldsymbol{\Phi}_2 f_{\boldsymbol{\Phi}_1}^{l-1}(\mathbf{Y})\right\|_F \\
&\quad + \left\|\left(\mathbf{I} - \tau(\mathbf{A}\boldsymbol{\Phi}_2)^\top \mathbf{A}\boldsymbol{\Phi}_2\right)\right\|_{2\to 2} \left\|f_{\boldsymbol{\Phi}_1}^{l-1}(\mathbf{Y}) - f_{\boldsymbol{\Phi}_2}^{l-1}(\mathbf{Y})\right\|_F \\
&\le \tau \left\|\mathbf{A}\right\|_{2\to 2} \left\|\mathbf{A}\boldsymbol{\Phi}_1 - \mathbf{A}\boldsymbol{\Phi}_2\right\|_{2\to 2} \left\|f_{\boldsymbol{\Phi}_1}^{l-1}(\mathbf{Y})\right\|_F + \tau \left\|\mathbf{A}\right\|_{2\to 2} \left\|\mathbf{A}\boldsymbol{\Phi}_1 - \mathbf{A}\boldsymbol{\Phi}_2\right\|_{2\to 2} \left\|f_{\boldsymbol{\Phi}_1}^{l-1}(\mathbf{Y})\right\|_F \\
&\quad + \left\|\left(\mathbf{I} - \tau(\mathbf{A}\boldsymbol{\Phi}_2)^\top \mathbf{A}\boldsymbol{\Phi}_2\right)\right\|_{2\to 2} \left\|f_{\boldsymbol{\Phi}_1}^{l-1}(\mathbf{Y}) - f_{\boldsymbol{\Phi}_2}^{l-1}(\mathbf{Y})\right\|_F \\
&= 2\tau \left\|\mathbf{A}\right\|_{2\to 2} \left\|\mathbf{A}\boldsymbol{\Phi}_1 - \mathbf{A}\boldsymbol{\Phi}_2\right\|_{2\to 2} \left\|f_{\boldsymbol{\Phi}_1}^{l-1}(\mathbf{Y})\right\|_F + \left\|\mathbf{I} - \tau\mathbf{A}^\top\mathbf{A}\right\|_{2\to 2} \left\|f_{\boldsymbol{\Phi}_1}^{l-1}(\mathbf{Y}) - f_{\boldsymbol{\Phi}_2}^{l-1}(\mathbf{Y})\right\|_F.
\end{aligned}
$$

Plugging this back into (8.37) gives us

$$\begin{aligned}
&\left\| f^l_{\mathbf{\Phi}_1}(\mathbf{Y}) - f^l_{\mathbf{\Phi}_2}(\mathbf{Y}) \right\|_F \qquad (8.38)\\
\leq &\left\| \mathbf{I} - \tau \mathbf{A}^\top \mathbf{A} \right\|_{2\to 2} \left\| f^{l-1}_{\mathbf{\Phi}_1}(\mathbf{Y}) - f^{l-1}_{\mathbf{\Phi}_2}(\mathbf{Y}) \right\|_F \\
&+\tau \left( 2 \|\mathbf{Y}\|_F + 2 \|\mathbf{A}\|_{2\to 2} \left\| f^{l-1}_{\mathbf{\Phi}_1}(\mathbf{Y}) \right\|_F \right) \|\mathbf{A}\mathbf{\Phi}_1 - \mathbf{A}\mathbf{\Phi}_2\|_{2\to 2} \\
\leq &\, A \left\| f^{l-1}_{\mathbf{\Phi}_1}(\mathbf{Y}) - f^{l-1}_{\mathbf{\Phi}_2}(\mathbf{Y}) \right\|_F + B_l \|\mathbf{A}\mathbf{\Phi}_1 - \mathbf{A}\mathbf{\Phi}_2\|_{2\to 2}\,, \qquad (8.39)
\end{aligned}$$

where $A$ and $B_l$ in the previous estimate (8.39) are given by

$$\begin{aligned}
A &= \left\| \mathbf{I} - \tau \mathbf{A}^\top \mathbf{A} \right\|_{2\to 2}, \\
Z_0 &= 0, \qquad Z_l = \sum_{k=0}^{l-1} \left\| \mathbf{I} - \tau \mathbf{A}^\top \mathbf{A} \right\|^k_{2\to 2}, \quad l \geq 1, \\
B_l &= \tau \|\mathbf{Y}\|_F \left( 2 + 2\tau \, \|\mathbf{A}\|^2_{2\to 2} \, Z_{l-1} \right), \qquad l \geq 1.
\end{aligned}$$

Using these abbreviations, the general formula for $K_L$ in (8.35) has the compact form

$$K_L = \sum_{l=1}^{L} A^{L-l} B_l, \qquad L \geq 1. \qquad (8.40)$$

Based on (8.39), we prove via induction that (8.34) holds for any number of layers $L \in \mathbb{N}$ with $K_L$ given by (8.40). For $L = 1$, we can directly calculate the constant $K_1$ via

$$\begin{aligned}
\left\| f^1_{\mathbf{\Phi}_1}(\mathbf{Y}) - f^1_{\mathbf{\Phi}_2}(\mathbf{Y}) \right\|_F &= \left\| S_{\tau\lambda}(\tau(\mathbf{A}\mathbf{\Phi}_1)^\top \mathbf{Y}) - S_{\tau\lambda}(\tau(\mathbf{A}\mathbf{\Phi}_2)^\top \mathbf{Y}) \right\|_F \\
&\leq \tau \|\mathbf{Y}\|_F \, \|\mathbf{A}\mathbf{\Phi}_1 - \mathbf{A}\mathbf{\Phi}_2\|_{2\to 2}\,,
\end{aligned}$$

so that $\tau \|\mathbf{Y}\|_F \leq 2\tau \|\mathbf{Y}\|_F = B_1 = K_1$, as claimed in (8.40).

Now we proceed with the induction step, assuming formula (8.40) to hold for some $L \in \mathbb{N}$. Applying the estimate after (8.38) for the output after layer $L+1$, we obtain

$$\begin{aligned}
\left\| f^{L+1}_{\mathbf{\Phi}_1}(\mathbf{Y}) - f^{L+1}_{\mathbf{\Phi}_2}(\mathbf{Y}) \right\|_F &\leq A \left\| f^{L}_{\mathbf{\Phi}_1}(\mathbf{Y}) - f^{L}_{\mathbf{\Phi}_2}(\mathbf{Y}) \right\|_F + B_{L+1} \|\mathbf{A}\mathbf{\Phi}_2 - \mathbf{A}\mathbf{\Phi}_1\|_{2\to 2} \\
&\leq A K_L \|\mathbf{A}\mathbf{\Phi}_2 - \mathbf{A}\mathbf{\Phi}_1\|_{2\to 2} + B_{L+1} \|\mathbf{A}\mathbf{\Phi}_2 - \mathbf{A}\mathbf{\Phi}_1\|_{2\to 2} \\
&\leq (A K_L + B_{L+1}) \|\mathbf{A}\mathbf{\Phi}_2 - \mathbf{A}\mathbf{\Phi}_1\|_{2\to 2},
\end{aligned}$$

and therefore,

$$K_{L+1} = AK_L + B_{L+1} = A\sum_{l=1}^{L} A^{L-l}B_l + B_{L+1} = \sum_{l=1}^{L+1} A^{(L+1)-l}B_l.$$

This is the desired expression for $K_{L+1}$ and finishes the proof of (8.34). It remains to prove the upper bound (8.36). In Remark 8.1, we have observed that $\|\mathbf{I} - \tau\mathbf{A}^\top\mathbf{A}\|_{2\to 2} = 1$ when $\tau\|\mathbf{A}\|_{2\to 2}^2 \le 1$. Therefore, we obtain

$$\begin{aligned} K_L &= \sum_{l=1}^{L} A^{L-l}B_l \le \sum_{l=1}^{L} B_l = \tau\|\mathbf{Y}\|_F \sum_{l=1}^{L}\left(2 + 2\tau\,\|\mathbf{A}\|_{2\to 2}^2\, Z_{l-1}\right) \\ &\le 2L\tau\|\mathbf{Y}\|_F + 2\tau\|\mathbf{Y}\|_F \sum_{l=1}^{L} Z_{l-1} \le 2L\tau\|\mathbf{Y}\|_F + 2\tau\|\mathbf{Y}\|_F \sum_{l=1}^{L} l \\ &= \tau\|\mathbf{Y}\|_F L(L+3), \end{aligned}$$

finishing the proof of the theorem. □

The following result is an adaptation of the previous theorem to take the special form of the final layer into account (a final linear transformation, followed by applying the function $\sigma$).

**Corollary 8.1** *Consider the thresholding networks $\boldsymbol{\Psi} f_{\boldsymbol{\Phi}}^L \in \mathcal{H}_2^L$ as defined in Sect. 8.4.3, with $L \ge 2$ and $\boldsymbol{\Psi}, \boldsymbol{\Phi} \in O(N)$. Then, for any $\boldsymbol{\Phi}_1, \boldsymbol{\Phi}_2 \in O(N)$ and $\boldsymbol{\Psi}_1, \boldsymbol{\Psi}_2 \in O(N)$, we have*

$$\begin{aligned} &\left\|\sigma(\boldsymbol{\Psi}_1 f_{\boldsymbol{\Phi}_1}^L(\mathbf{Y})) - \sigma(\boldsymbol{\Psi}_2 f_{\boldsymbol{\Phi}_2}^L(\mathbf{Y}))\right\|_F \\ &\le M_L\|\boldsymbol{\Psi}_1 - \boldsymbol{\Psi}_2\|_{2\to 2} + K_L\|\mathbf{A}\boldsymbol{\Phi}_1 - \mathbf{A}\boldsymbol{\Phi}_2\|_{2\to 2}, \end{aligned} \tag{8.41}$$

*with $K_L$ as in Theorem 8.4 and*

$$M_L = \tau\|\mathbf{A}\|_{2\to 2}\|\mathbf{Y}\|_F \sum_{k=0}^{L-1} \left\|\mathbf{I} - \tau\mathbf{A}^\top\mathbf{A}\right\|_{2\to 2}^k. \tag{8.42}$$

*Under the additional assumption that $\tau\|\mathbf{A}\|_{2\to 2}^2 \le 1$, we have*

$$\begin{aligned} &\left\|\sigma(\boldsymbol{\Psi}_1 f_{\boldsymbol{\Phi}_1}^L(\mathbf{Y})) - \sigma(\boldsymbol{\Psi}_2 f_{\boldsymbol{\Phi}_2}^L(\mathbf{Y}))\right\|_F \\ &\le \tau\|\mathbf{Y}\|_F \left(L\|\mathbf{A}\|_{2\to 2}\|\boldsymbol{\Psi}_1 - \boldsymbol{\Psi}_2\|_{2\to 2} + L(L+3)\|\mathbf{A}\boldsymbol{\Phi}_1 - \mathbf{A}\boldsymbol{\Phi}_2\|_{2\to 2}\right). \end{aligned}$$

***Proof*** Let us begin with the following estimates, which now include the application of the measurement and the respective dictionary after the final layer. By the 1-Lipschitzness of $\sigma$, adding mixed terms and applying the triangle inequality, and finally using Theorem 8.4 for the second summand in the last step, we obtain

$$
\begin{aligned}
&\left\| \sigma\left(\boldsymbol{\Psi}_1 f^L_{\boldsymbol{\Phi}_1}(\mathbf{Y})\right) - \sigma\left(\boldsymbol{\Psi}_2 f^L_{\boldsymbol{\Phi}_2}(\mathbf{Y})\right) \right\|_F \\
&\leq \left\| \boldsymbol{\Psi}_1 f^L_{\boldsymbol{\Phi}_1}(\mathbf{Y}) - \boldsymbol{\Psi}_2 f^L_{\boldsymbol{\Phi}_1}(\mathbf{Y}) + \boldsymbol{\Psi}_2 f^L_{\boldsymbol{\Phi}_1}(\mathbf{Y}) - \boldsymbol{\Psi}_2 f^L_{\boldsymbol{\Phi}_2}(\mathbf{Y}) \right\|_F \\
&\leq \left\| \boldsymbol{\Psi}_1 f^L_{\boldsymbol{\Phi}_1}(\mathbf{Y}) - \boldsymbol{\Psi}_2 f^L_{\boldsymbol{\Phi}_1}(\mathbf{Y}) \right\|_F + \left\| \boldsymbol{\Psi}_2 f^L_{\boldsymbol{\Phi}_1}(\mathbf{Y}) - \boldsymbol{\Psi}_2 f^L_{\boldsymbol{\Phi}_2}(\mathbf{Y}) \right\|_F \\
&\leq \left\| f^L_{\boldsymbol{\Phi}_1}(\mathbf{Y}) \right\|_F \|\boldsymbol{\Psi}_1 - \boldsymbol{\Psi}_2\|_{2\to 2} + \left\| f^L_{\boldsymbol{\Phi}_1}(\mathbf{Y}) - f^L_{\boldsymbol{\Phi}_2}(\mathbf{Y}) \right\|_F \\
&\leq \left\| f^L_{\boldsymbol{\Phi}_1}(\mathbf{Y}) \right\|_F \|\boldsymbol{\Psi}_1 - \boldsymbol{\Psi}_2\|_{2\to 2} + K_L \, \|\mathbf{A}\boldsymbol{\Phi}_1 - \mathbf{A}\boldsymbol{\Phi}_2\|_{2\to 2} \, .
\end{aligned}
$$

Now, (8.41) follows from Lemma 8.3. The additional simplified bounds then easily follow from the respective ones in Theorem 8.4 as well as in (8.27). □

*Remark 8.2* One may try a similar computation like in the proof above for the hypothesis space $\mathcal{H}^L_1$ instead $\mathcal{H}^L_2$. However, after the analog estimate for $\boldsymbol{\Phi}_1, \boldsymbol{\Phi}_2 \in O(N)$,

$$
\left\| \boldsymbol{\Phi}_1 f^L_{\boldsymbol{\Phi}_1}(\mathbf{Y}) - \boldsymbol{\Phi}_2 f^L_{\boldsymbol{\Phi}_2}(\mathbf{Y}) \right\|_F \leq \left\| f^L_{\boldsymbol{\Phi}_1}(\mathbf{Y}) \right\|_F \|\boldsymbol{\Phi}_1 - \boldsymbol{\Phi}_2\|_{2\to 2} + K_L \, \|\mathbf{A}\boldsymbol{\Phi}_1 - \mathbf{A}\boldsymbol{\Phi}_2\|_{2\to 2} \, ,
$$

we need to consider both $\|\mathbf{A}\boldsymbol{\Phi}_1 - \mathbf{A}\boldsymbol{\Phi}_2\|_{2\to 2}$ and $\|\boldsymbol{\Phi}_1 - \boldsymbol{\Phi}_2\|_{2\to 2}$ for later covering number arguments. Using $\mathcal{H}^L_2$ helps to obtain more concise covering numbers for the class. Therefore, we decouple the single dictionary applied after the final layer from the previous layers (which all appear together with $\mathbf{A}$).

### 8.4.5 Covering number estimates

Our proof is built on Dudley's integral in (8.33). We need to compute covering numbers $\mathcal{N}(\mathcal{M}_2, \|\cdot\|_F, \epsilon)$ at different scales $\epsilon > 0$ to evaluate the integral for the space $\mathcal{M}_2$. We start from the following lemma [12, Proposition C.3] and adapt it to our problem.

**Lemma 8.4** *Let $\epsilon > 0$, and let $\|\cdot\|$ be a norm on a $n$-dimensional vector space $V$. Then, for any subset $U \subseteq B_{\|\cdot\|} := \{x \in V : \|x\| \leq 1\}$, it holds*

$$
\mathcal{N}(U, \|\cdot\|, \epsilon) \leq \left(1 + \frac{2}{\epsilon}\right)^n .
$$

The next lemma provides a bound for product spaces, based on individual covering numbers.

**Lemma 8.5** *Consider two metric spaces $(\mathcal{S}_1, d_1)$, $(\mathcal{S}_2, d_2)$. We define the product metric $\mathcal{S}$, equipped with the metric $d$ by*

$$\mathcal{S} = (\mathcal{S}_1 \times \mathcal{S}_2, d), \qquad d(x, y) = \sum_{k=1}^{2} d_k(x_k, y_k), \tag{8.43}$$

*where* $x = (x_1, x_2)$, $y = (y_1, y_2) \in \mathcal{S}$. *Then, we have the covering number estimate*

$$\mathcal{N}(\mathcal{S}, d, \varepsilon) \leq \prod_{k=1}^{2} \mathcal{N}(\mathcal{S}_k, d_k, \varepsilon/2)\,. \tag{8.44}$$

***Proof*** Suppose that, for $k = 1, 2$, we have individual coverings of $\mathcal{S}_k$ at level $\varepsilon/2$ of cardinality $\mathcal{N}(\mathcal{S}_k, d_k, \varepsilon/2)$. We will show that the product of all these $\varepsilon/2$-nets is an $\varepsilon$-net for the product space $S$. Indeed, let $x = (x_1, x_2) \in \mathcal{S}$, i.e., $x_k \in \mathcal{S}_k$. Then, for each $x_k \in \mathcal{S}_k$, there exists some element $y_k$ in the $\varepsilon/2$-net of $\mathcal{S}_k$, i.e., $d_k(x_k, y_k) \leq \varepsilon/2$. Then, $y = (y_1, y_2)$ is an element of the product of all nets, and by the definition of the metric $d$, there is $d(x, y) \leq \varepsilon/2 + \varepsilon/2 = \varepsilon$. □

The following lemma provides a covering number estimate of $\mathbf{A}$ applied to the orthogonal group.

**Lemma 8.6** *For a fixed matrix* $\mathbf{A} \in \mathbb{R}^{n\times N}$, *consider the set* $\mathcal{W}$ *defined by*

$$\mathcal{W} = \{\mathbf{A}\boldsymbol{\Phi} : \boldsymbol{\Phi} \in O(N)\} \subset \mathbb{R}^{n\times N}, \tag{8.45}$$

*i.e.,* $\mathbf{A}$ *applied to the orthogonal group. The covering number estimate is given by*

$$\mathcal{N}(\mathcal{W}, \|\cdot\|_{2\to 2}, \epsilon) \leq \left(1 + \frac{2\|\mathbf{A}\|_{2\to 2}}{\epsilon}\right)^{nN}.$$

***Proof*** First note that $\mathcal{W}$ can be rewritten as

$$\mathcal{W} = \left\{\|\mathbf{A}\|_{2\to 2} \frac{\mathbf{A}\boldsymbol{\Phi}}{\|\mathbf{A}\|_{2\to 2}} : \boldsymbol{\Phi} \in O(N)\right\}. \tag{8.46}$$

For the covering numbers of the orthogonal group $(O(N), \|\cdot\|_{2\to 2})$ Equipped with the spectral norm, we have

$$\mathcal{N}(O(N), \|\cdot\|_{2\to 2}, \epsilon) \leq \left(1 + \frac{2}{\epsilon}\right)^{N^2}.$$

This follows from the fact that the orthogonal group $O(N)$ is contained in $B^{N\times N}_{\|\cdot\|_{2\to 2}}$, and therefore, Lemma 8.4 applies. This bound then gives

$$\mathcal{N}(\mathcal{W}, \|\cdot\|_{2\to 2}, \epsilon) = \mathcal{N}(\{\mathbf{A}\boldsymbol{\Phi}/\|\mathbf{A}\|_{2\to 2} : \boldsymbol{\Phi} \in O(N)\}, \|\cdot\|_{2\to 2}, \epsilon/\|\mathbf{A}\|_{2\to 2})$$

$$\leq \left(1 + \frac{2\|\mathbf{A}\|_{2\to 2}}{\epsilon}\right)^{nN}.$$

□

Recall that for Dudley's inequality, we need to estimate the covering numbers $\mathcal{N}(\mathcal{M}_2, \|\cdot\|_{2\to 2}, \epsilon)$ of the set $\mathcal{M}_2$ defined in (8.31). In Corollary 8.1, we showed we can estimate distances in $\mathcal{M}_2$ via distances of the underlying parameters, $\|\mathbf{\Psi}_1 - \mathbf{\Psi}_2\|_{2\to 2}$ and $\|\mathbf{A}\mathbf{\Phi}_1 - \mathbf{A}\mathbf{\Phi}_2\|_{2\to 2}$. We make use of this in the next corollary, which prepares the application of Dudley's inequality afterward.

**Corollary 8.2** *The covering numbers of the set $\mathcal{M}_2$ are bounded by*

$$\log\left(\mathcal{N}(\mathcal{M}_2, \|\cdot\|_{2\to 2}, \epsilon)\right) \leq N^2 \cdot \log\left(1 + \frac{4M_L}{\epsilon}\right) + nN \cdot \log\left(1 + \frac{4\|\mathbf{A}\|_{2\to 2} K_L}{\epsilon}\right).$$

***Proof*** Using the definition of the set (8.45), we have

$$\mathcal{N}(K_L\{\mathbf{A}\mathbf{\Phi} : \mathbf{\Phi} \in O(N)\}, \|\cdot\|_{2\to 2}, \epsilon) = \mathcal{N}(\{\mathbf{A}\mathbf{\Phi} : \mathbf{\Phi} \in O(N)\}, \|\cdot\|_{2\to 2}, \epsilon/K_L) \leq \left(1 + \frac{2\|\mathbf{A}\|_{2\to 2} K_L}{\epsilon}\right)^{nN}.$$

Furthermore, since $O(N) \subset B_{\|\cdot\|_{2\to 2}}^{N\times N}$, and by Lemma 8.4 (with $\varepsilon/2$ instead of $\varepsilon$)

$$\mathcal{N}(M_L \cdot O(N), \|\cdot\|_{2\to 2}, \epsilon/2) = \mathcal{N}(O(N), \|\cdot\|_{2\to 2}, \epsilon/(2M_L)) \leq \left(1 + \frac{4M_L}{\epsilon}\right)^{N^2}.$$

Applying Lemma 8.5 and the previous estimates, we can now bound the covering number of $\mathcal{M}_2$ by

$$\begin{aligned}\mathcal{N}(\mathcal{M}_2, \|\cdot\|_F, \epsilon) &\leq \mathcal{N}(M_L \cdot O(N) \times K_L \cdot \mathcal{W}, \|\cdot\|_{2\to 2}, \epsilon) \\ &\leq \mathcal{N}(M_L \cdot O(N), \|\cdot\|_{2\to 2}, \epsilon/2)\, \mathcal{N}(K_L \cdot \mathcal{W}, \|\cdot\|_{2\to 2}, \epsilon/2) \\ &\leq \left(1 + \frac{4M_L}{\epsilon}\right)^{N^2} \left(1 + \frac{4\|\mathbf{A}\|_{2\to 2} K_L}{\epsilon}\right)^{nN},\end{aligned}$$

which immediately gives us the desired statement. □

### 8.4.6 Main result

Finally, we are able to state and prove the main result of this section.

**Theorem 8.5** *Consider the hypothesis space $\mathcal{H}_2^L$ defined in (8.20). With probability at least $1-\delta$, for all $h \in \mathcal{H}_2^L$, the generalization error is bounded as*

$$\mathcal{L}(h) \leq \hat{\mathcal{L}}(h) + 8B_{\text{out}}\sqrt{\frac{Nn}{m}}\sqrt{\log e\left(1+\frac{8K_L\|\mathbf{A}\|_{2\to 2}}{\sqrt{m}B_{\text{out}}}\right)} + 8B_{\text{out}}\frac{N}{\sqrt{m}}\sqrt{\log e\left(1+\frac{8M_L}{\sqrt{m}B_{\text{out}}}\right)} + 4(B_{\text{in}}+B_{\text{out}})\sqrt{\frac{2\log(4/\delta)}{m}},$$

*where $K_L$ is the constant in (8.35).*

***Proof*** For the proof, it remains to bound the Rademacher complexity via Dudley's integral (8.33), for which in turn we use the covering number arguments from the previous subsection (Corollary 8.2) as follows:

$$\begin{aligned}
\mathcal{R}_m(\ell\circ\mathcal{H}_2^L) &= \mathbb{E}\sup_{\mathbf{M}\in\mathcal{M}_2}\frac{1}{m}\sum_{i=1}^{m}\sum_{k=1}^{N}\epsilon_{ik}M_{ik}\\
&\leq \frac{4\sqrt{2}}{m}\int_0^{\sqrt{m}B_{\text{out}}/2}\sqrt{\log\mathcal{N}(\mathcal{M}_2,,\|\cdot\|_F,\epsilon)}\,\mathrm{d}\epsilon\\
&\leq \frac{4\sqrt{2}}{m}\int_0^{\sqrt{m}B_{\text{out}}/2}\sqrt{N^2\cdot\log\left(1+\frac{4M_L}{\epsilon}\right)}\,\mathrm{d}\epsilon\\
&\quad+\frac{4\sqrt{2}}{m}\int_0^{\sqrt{m}B_{\text{out}}/2}\sqrt{nN\cdot\log\left(1+\frac{4\|\mathbf{A}\|_{2\to 2}K_L}{\epsilon}\right)}\,\mathrm{d}\epsilon\\
&\leq \frac{4\sqrt{2}N}{m}\int_0^{\sqrt{m}B_{\text{out}}/2}\sqrt{\log\left(1+\frac{4M_L}{\epsilon}\right)}\,\mathrm{d}\epsilon\\
&\quad+\frac{4\sqrt{2nN}}{m}\int_0^{\sqrt{m}B_{\text{out}}/2}\sqrt{\log\left(1+\frac{4\|\mathbf{A}\|_{2\to 2}K_L}{\epsilon}\right)}\,\mathrm{d}\epsilon\\
&\leq 2\sqrt{2}B_{\text{out}}\frac{N}{\sqrt{m}}\sqrt{\log\left(e\left(1+\frac{4M_L}{\sqrt{m}B_{\text{out}}/2}\right)\right)}\\
&\quad+2\sqrt{2}B_{\text{out}}\sqrt{\frac{Nn}{m}}\sqrt{\log\left(e\left(1+\frac{4K_L\|\mathbf{A}\|_{2\to 2}}{\sqrt{m}B_{\text{out}}/2}\right)\right)},
\end{aligned}$$

where we have used the following inequality for the last step [12, Lemma C.9]

$$\int_0^\alpha \sqrt{\log\left(1+\frac{\beta}{t}\right)}\,\mathrm{d}t \le \alpha\sqrt{\log\left(e(1+\beta/\alpha)\right)} \quad \text{for} \quad \alpha,\beta>0. \tag{8.47}$$

The theorem is obtained using Theorem 8.1 with the upper bound $c = B_{\text{in}} + B_{\text{out}}$ for the functions output from (8.23) and bounding the Rademacher complexity term (8.19) with the generalized contraction principle Lemma 8.2, which in turn is bounded using Dudley's integral as above. □

Let us make the reasonable assumption that $\tau\|\mathbf{A}\|_{2\to 2} \le 1$. Taking into account that $M_L \le \tau\|\mathbf{A}\|_{2\to 2}\|\mathbf{Y}\|_F L$, see also (8.27), i.e., $M_L$ scales at most linearly in $L$ (which remains inside the logarithm), and since $K_L$ depends quadratically on $L$, see (8.36), we have

$$\mathcal{L}(h) - \hat{\mathcal{L}}(h) \lesssim \frac{N}{\sqrt{m}}\sqrt{\log(L)} + \sqrt{\frac{Nn}{m}}\sqrt{\log(L)} \sim \sqrt{\frac{\log(L)N(N+n)}{m}} \sim \sqrt{\frac{\log(L)N^2}{m}}, \tag{8.48}$$

where the last relation holds under the reasonable assumption that $1 \le n \le N$. This estimate is stated more rigorously in the following corollary.

**Corollary 8.3** *Consider the hypothesis space $\mathcal{H}_2^L$ defined in (8.20) and assume that $\tau\|\mathbf{A}\|_{2\to 2}^2 \le 1$. With probability at least $1-\delta$, for all $h \in \mathcal{H}_2^L$, the generalization error is bounded as*

$$\begin{aligned}\mathcal{L}(h) \le \hat{\mathcal{L}}(h) &+ 8B_{\text{out}}\sqrt{\frac{Nn}{m}}\sqrt{1+\log\left(2+\frac{8L(L+3)\tau\|\mathbf{Y}\|_F\|\mathbf{A}\|_{2\to 2}}{\sqrt{m}B_{\text{out}}}\right)} \\ &+8B_{\text{out}}\frac{N}{\sqrt{m}}\sqrt{\log e\left(1+\frac{8\tau L\|\mathbf{A}\|_{2\to 2}\|\mathbf{Y}\|_F}{\sqrt{m}B_{\text{out}}}\right)} + 4(B_{\text{in}}+B_{\text{out}})\sqrt{\frac{2\log(4/\delta)}{m}},\end{aligned}$$

*where $K_L$ is the perturbation bound in (8.35).*

Theorem 8.2 follows immediately from Theorem 8.5 and Corollary 8.3. Indeed, we have seen in (8.27) that

$$\tau\|\mathbf{A}\|_{2\to 2}\|\mathbf{Y}\|_F \le \sqrt{m}B_{\text{in}}. \tag{8.49}$$

Using that $B_{\text{in}} = B_{\text{out}}$ by assumption, and, for simplicity also assuming that $L \ge 2$ such that $2 + 8L(L+3) \le (5L)^2$, we therefore have

$$\log\left(2+\frac{8L(L+3)\tau\|\mathbf{Y}\|_F\|\mathbf{A}\|_{2\to 2}}{\sqrt{m}B_{\text{out}}}\right) \le \log(2+8L(L+3)) \le 2\log(5L).$$

Plugging in these estimates and using that $\mathcal{H}_1^L \subseteq \mathcal{H}_2^L$ give the statement of Theorem 8.2.

As already pointed out above, the deep thresholding network we analyze is, due to the weight sharing, a recurrent neural network. The authors of [10] derive VC dimension bounds of recurrent networks for recurrent perceptrons with binary outputs. The VC dimension of recurrent neural networks for different classes of activation functions has been studied by the authors of [25]. However, their results do not immediately apply to our setup since they focus on one-dimensional inputs and outputs, which of course do not suit our vector-valued regression problem and, moreover, would correspond to taking just one single measurement. Even in the scenario that is closest to ours, namely, fixed piecewise polynomial activation functions with $n = 1$, their VC dimension bound scales between $O(Lw)$ and $O(Lw^2)$, where $L$ is the number of layers and $w$ is the number of trainable parameters in the network. In our case, the number of trainable parameters is equal to the dimension of the orthogonal group $O(N)$, which is $N(N-1)/2$. Therefore, their bounds scale between $O(LN^2)$ and $O(LN^4)$. In contrast, if $n = 1$, our bound scales only like $O(N\sqrt{\log(L)})$. Besides, we only make use of Lipschitzness of the activation function.

## 8.5 Thresholding Networks for Sparse Recovery

In Theorem 8.2 and Corollary 8.3, we have provided a worst-case bound on the sample complexity that holds uniformly over the hypothesis space and for any arbitrary data distribution. It is interesting to see if this bound can be improved for data distributions limited to low-complexity sets distributions, for example over the set of sparse vectors. ISTA is used mainly in sparse coding and recovery tasks; therefore, it is reasonable to ask if the generalization error behaves similarly when it is applied to sparse recovery tasks.

We consider a synthetic dataset as well as the MNIST dataset [27]. For both cases, the measurement matrix is a random Gaussian matrix properly normalized to guarantee convergence of soft-thresholding algorithms. The synthetic data is generated for different input and output dimensions and sparsity level. The original dictionary is a random orthogonal matrix. The default parameters are $N = 120$, $n = 80$, and sparsity equal to 10. Sparse vectors are generated by choosing their support uniformly randomly and then picking non-zero values according to the standard normal distribution. The experiments for the synthetic data are repeated at least 50 times, and the results are averaged over the repetitions. For both the MNIST and synthetic datasets, we sweep over $L$, $N$, and $n$ to see how the generalization error behaves.

There are different ways to implement the orthogonality constraint for weight matrices. One way [29] is based on the fact that the matrix exponential mapping provides a bijective mapping from the skew-symmetric matrices onto the special orthogonal group $SO(N)$. However, we use the alternative method of adding a regularization term $\|\mathbf{I} - \mathbf{\Phi}^\top \mathbf{\Phi}\|_F$ (or another matrix norm) to the loss function, which means to penalize $\mathbf{\Phi}$ that is far from being orthogonal.

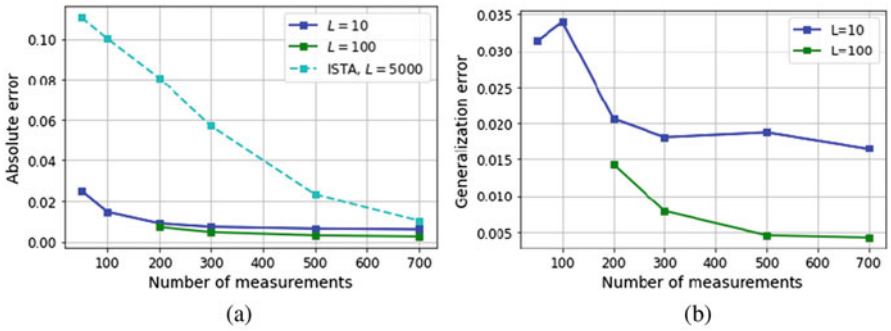



**Fig. 8.1** MNIST dataset. (**a**) Absolute reconstruction error for different measurements of MNIST. (**b**) Generalization error for different measurements of MNIST

We choose different number of measurements and layers for both datasets. For each one, the network is trained for a few epochs. Mostly not more than 10 epochs are required to get first promising results, and often times, the loss goes down very slowly after 10 epochs.

All experiments (see Fig. 8.1a) show that it is possible to recover the original vectors $\mathbf{x}$ with as few as 10 layers, which is less than typical when using ISTA (see supplementary materials for some visuals). Note that the error in the MNIST experiments is the pixel-based error normalized by the image dimension, and MNIST pixels are all normalized between 0 and 1. We have chosen ISTA with a similar structure and 5000 iterations. The result warrants the applicability of dictionary learning for sparse reconstruction.

Figure 8.2a confirms the dependence of the generalization error on the number of layers $L$. Increasing the number of layers increases the generalization error for a fixed number of measurements $n$. However, the generalization error decreases by increasing the number of layers for MNIST dataset. For both synthetic and MNIST datasets, it seems that increasing the number of measurements decreases the generalization error. See Figs. 8.1b, 8.2a,b. Besides, Fig. 8.2b shows that increasing $N$ increases the generalization error. Therefore, our bound scales correctly with the input dimension and the number of layers but incorrectly with the number of measurements. Although not predicted by our theoretical results, this is not unexpected. Note that the number of measurements $n$ is not essential here since it can always be upper bounded by $N$. Therefore, the theoretical bound on the generalization error (see (8.16), and Theorem 8.2 as well as Corollary 8.3 for more details) can be lower and upper bounded via

$$\sqrt{\frac{\log(L)}{m}}N \leq \sqrt{\frac{\log(L)}{m}}(N+\sqrt{Nn}) \leq 2\sqrt{\frac{\log(L)}{m}}N.$$

Furthermore, as mentioned above, the sample complexity is supposed to apply to all possible input distributions. If we restrict ourselves to distributions over low-

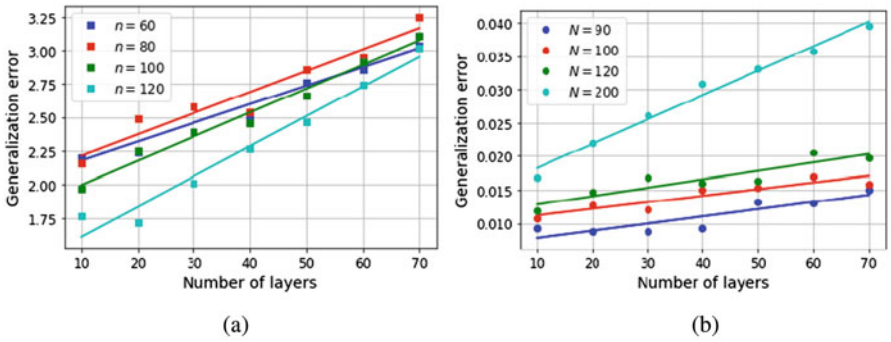



**Fig. 8.2** Generalization error for synthetic dataset. (**a**) Generalization error for different measurements of synthetic data ($N = 120$). (**b**) Generalization error for different input dimensions of synthetic data ($n = 80$)

complexity sets, then various worst-case bounds in our analysis might be improved. The experiments seem to confirm this intuition. Namely, for the MNIST dataset, there is a clear improvement with increasing the number of measurements and the number of layers. This is intuitive from a compressive sensing standpoint, as more number of layers in ISTA leads to better results and more measurements provide more information about the input.

On the other hand, the synthetic dataset shows that the generalization error increases with the input dimension and the number of layers. Note that the bound of this chapter is obtained for a very general setting where nothing is assumed on the data structure. Additional assumptions on the structure of the problem, i.e., sparsity, can be used to improve the current bound. Nonetheless, the linear dimension dependency of the current bound makes it a very good baseline for future comparisons.

The model that is used for our experiments shares the weights across layers conforming to our theoretical setup. However, we can improve the performance of this method by using ideas similar to LISTA literature. Many works on LISTA use a different dictionary at each layer, which eases the training procedure and can lead to potentially better results.

## 8.6 Conclusion and Outlook

In this chapter, we have derived a generalization bound for an unfolded ISTA algorithm where, similar to LISTA, the dictionary is learned via learning the reconstruction algorithm and interpreted as neural network with shared layers. To the best of our knowledge, this is the first result of its kind. Our proof utilizes a Rademacher complexity analysis and obtains generalization bounds with only linear dependence on the dimension. The comparison of our theoretical results and the

numerical results suggests that we might be able to obtain tighter generalization bounds of neural networks for structured input data. Future works also consist of considering more intricate structures with more flexible weight sharing between the layers and also learning parameters such as the stepsizes and thresholds simultaneously.

**Acknowledgments** The authors would like to thank Sebastian Lubjuhn for proofreading an earlier version of this paper and giving valuable suggestions for improvement. The third author acknowledges funding from the Deutsche Forschungsgemeinschaft (DFG) through the project *Structured Compressive Sensing via Neural Network Learning* (SCoSNeL, MA 1184/36-1) within the SPP 1798 *Compressed Sensing in Information Processing* (CoSIP).

## References

1. Aberdam, A., Golts, A., Elad, M.: Ada-LISTA: Learned solvers adaptive to varying models. Preprint. arXiv:2001.08456 (2020)
2. Arora, S., Ge, R., Neyshabur, B., Zhang, Y.: Stronger generalization bounds for deep nets via a compression approach. In: International Conference on Machine Learning, pp. 254–263 (2018)
3. Arridge, S., Maass, P., Öktem, O., Schönlieb, C.B.: Solving inverse problems using data-driven models. Acta Numerica **28**, 1–174 (2019)
4. Bartlett, P.L., Mendelson, S.: Rademacher and Gaussian complexities: risk bounds and structural results. J. Mach. Learn. Res. **3**(Nov), 463–482 (2002)
5. Bartlett, P.L., Foster, D.J., Telgarsky, M.J.: Spectrally-normalized margin bounds for neural networks. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems 30, pp. 6240–6249 (2017)
6. Behrens, F., Sauder, J., Jung, P.: Neurally augmented ALISTA. In: International Conference on Learning Representations (2021)
7. Chen, X., Liu, J., Wang, Z., Yin, W.: Theoretical linear convergence of unfolded ISTA and its practical weights and thresholds. In: Advances in Neural Information Processing Systems, pp. 9061–9071 (2018)
8. Chou, H.H., Gieshoff, C., Maly, J., Rauhut, H.: Gradient descent for deep matrix factorization: dynamics and implicit bias towards low rank. Preprint. arxiv:2011.13772 (2021)
9. Daras, G., Dean, J., Jalal, A., Dimakis, A.G.: Intermediate layer optimization for inverse problems using deep generative models. arXiv:2102.07364 [cs] (2021). http://arxiv.org/abs/2102.07364
10. DasGupta, B., Sontag, E.: Sample complexity for learning recurrent perceptron mappings. IEEE Trans. Inf. Theory **42**(5), 1479–1487 (1996)
11. Daubechies, I., Defrise, M., De Mol, C.: An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. Commun. Pure Appl. Math. J. Issued Courant Instit. Math. Sci. **57**(11), 1413–1457 (2004)
12. Foucart, S., Rauhut, H.: A Mathematical Introduction to Compressive Sensing. Applied and Numerical Harmonic Analysis. Springer, New York (2013)
13. Genzel, M., Macdonald, J., März, M.: Solving Inverse Problems With Deep Neural Networks – Robustness Included? arXiv:2011.04268 (2020)
14. Georgogiannis, A.: The generalization error of dictionary learning with Moreau envelopes. In: Dy, J., Krause, A. (eds.) Proceedings of the 35th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 80, pp. 1617–1625. PMLR, Stockholmsmässan, Stockholm (2018)

15. Golowich, N., Rakhlin, A., Shamir, O.: Size-independent sample complexity of neural networks. In: Conference on Learning Theory, pp. 297–299 (2018)
16. Gottschling, N.M., Antun, V., Adcock, B., Hansen, A.C.: The troublesome kernel: why deep learning for inverse problems is typically unstable. Preprint. arXiv:2001.01258 (2020)
17. Gregor, K., LeCun, Y.: Learning fast approximations of sparse coding. In: Proceedings of the 27th International Conference on International Conference on Machine Learning, pp. 399–406 (2010)
18. Gribonval, R., Schnass, K.: Dictionary identification – sparse matrix-factorisation via $\ell_1$-minimisation. IEEE Trans. Inf. Theory **56**(7), 3523–3539 (2010)
19. Gribonval, R., Jenatton, R., Bach, F., Kleinsteuber, M., Seibert, M.: Sample complexity of dictionary learning and other matrix factorizations. IEEE Trans. Inf. Theory **61**(6), 3469–3486 (2015)
20. Hasannasab, M., Hertrich, J., Neumayer, S., Plonka, G., Setzer, S., Steidl, G.: Parseval proximal neural networks. J. Fourier Anal. Appl. **26**(4), 59 (2020)
21. Jiang, Y., Neyshabur, B., Mobahi, H., Krishnan, D., Bengio, S.: Fantastic generalization measures and where to find them. In: International Conference on Learning Representations (2020)
22. Jung, A., Eldar, Y.C., Görtz, N.: Performance limits of dictionary learning for sparse coding. In: 2014 22nd European Signal Processing Conference (EUSIPCO), pp. 765–769 (2014)
23. Jung, A., Eldar, Y.C., Görtz, N.: On the minimax risk of dictionary learning. IEEE Trans. Inf. Theory **62**(3), 1501–1515 (2016)
24. Kamilov, U.S., Mansour, H.: Learning optimal nonlinearities for iterative thresholding algorithms. IEEE Signal Process. Lett. **23**(5), 747–751 (2016)
25. Koiran, P., Sontag, E.D.: Vapnik-Chervonenkis dimension of recurrent neural networks. Discret. Appl. Math. **86**(1), 63–79 (1998)
26. Koltchinskii, V.: Rademacher penalties and structural risk minimization. IEEE Trans. Inf. Theory **47**(5), 1902–1914 (2001)
27. LeCun, Y.: The MNIST database of handwritten digits. http://yann.lecun.com/exdb/mnist/
28. Ledoux, M., Talagrand, M.: Probability in Banach spaces: isoperimetry and processes. Classics in Mathematics. Springer, Berlin (2011)
29. Lezcano-Casado, M., Martınez-Rubio, D.: Cheap orthogonal constraints in neural networks: a simple parametrization of the orthogonal and unitary group. In: International Conference on Machine Learning, pp. 3794–3803. PMLR (2019)
30. Liu, J., Chen, X., Wang, Z., Yin, W.: ALISTA: Analytic weights are as good as learned weights in LISTA. In: International Conference on Learning Representations (2019)
31. Maurer, A.: A vector-contraction inequality for Rademacher complexities. In: Algorithmic Learning Theory, Lecture Notes in Computer Science, pp. 3–17 (2016)
32. Mousavi, A., Patel, A.B., Baraniuk, R.G.: A deep learning approach to structured signal recovery. In: 2015 53rd Annual Allerton Conference on Communication, Control, and Computing, Allerton, pp. 1336–1343. IEEE, Piscataway (2015)
33. Nagarajan, V., Kolter, J.Z.: Uniform convergence may be unable to explain generalization in deep learning. In: Advances in Neural Information Processing Systems, pp. 11611–11622 (2019)
34. Neyshabur, B., Tomioka, R., Srebro, N.: In search of the real inductive bias: on the role of implicit regularization in deep learning. In: ICLR (Workshop) (2015)
35. Neyshabur, B., Bhojanapalli, S., McAllester, D., Srebro, N.: Exploring generalization in deep learning. In: Advances in Neural Information Processing Systems, pp. 5947–5956 (2017)
36. Neyshabur, B., Bhojanapalli, S., Srebro, N.: A PAC-Bayesian approach to spectrally-normalized margin bounds for neural networks. In: International Conference on Learning Representations (2018)
37. Neyshabur, B., Li, Z., Bhojanapalli, S., LeCun, Y., Srebro, N.: The role of over-parametrization in generalization of neural networks. In: International Conference on Learning Representations (2019)

38. Rakhlin, A., Mukherjee, S., Poggio, T.: Stability results in learning theory. Anal. Appl. **03**(04), 397–417 (2005)
39. Rauhut, H., Schnass, K., Vandergheynst, P.: Compressed sensing and redundant dictionaries. IEEE Trans. Inf. Theory **54**(5), 2210–2219 (2008)
40. Schnass, K.: On the identifiability of overcomplete dictionaries via the minimisation principle underlying K-SVD. Appl. Comput. Harmonic Anal. (3), 37 (2014)
41. Shalev-Shwartz, S., Ben-David, S.: Understanding Machine Learning: From Theory to Algorithms. Cambridge University Press, New York (2014)
42. Shalev-Shwartz, S., Shamir, O., Srebro, N., Sridharan, K.: Learnability, stability and uniform convergence. J. Mach. Learn. Res. **11**, 2635–2670 (2010)
43. Sprechmann, P., Bronstein, A.M., Sapiro, G.: Learning efficient sparse and low rank models. IEEE Trans. Pattern Anal. Mach. Intell. **37**(9), 1821–1833 (2015)
44. Sreter, H., Giryes, R.: Learned convolutional sparse coding. In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2191–2195. IEEE, Piscataway (2018)
45. Talagrand, M.: Upper and Lower Bounds for Stochastic Processes: Modern Methods and Classical Problems. Ergebnisse der Mathematik und ihrer Grenzgebiete. 3. Folge/A Series of Modern Surveys in Mathematics. Springer, Berlin (2014)
46. Vainsencher, D., Mannor, S., Bruckstein, A.M.: The sample complexity of dictionary learning. J. Mach. Learn. Res. **12**(Nov), 3259–3281 (2011)
47. Vapnik, V.N.: The Nature of Statistical Learning Theory. Springer, New York. Imprint: Springer, New York (2000)
48. Vapnik, V.N., Chervonenkis, A.Y.: On the uniform convergence of relative frequencies of events to their probabilities. In: Vovk, V., Papadopoulos, H., Gammerman, A. (eds.) Measures of Complexity: Festschrift for Alexey Chervonenkis, pp. 11–30 (2015)
49. Wu, K., Guo, Y., Li, Z., Zhang, C.: Sparse coding with gated learned ISTA. In: International Conference on Learning Representations (2020)
50. Xin, B., Wang, Y., Gao, W., Wipf, D., Wang, B.: Maximal sparsity with deep networks? In: Advances in Neural Information Processing Systems, pp. 4340–4348 (2016)
51. Xu, H., Mannor, S.: Robustness and generalization. Mach. Learn. **86**(3), 391–423 (2012)
52. Yang, Y., Sun, J., Li, H., Xu, Z.: Deep ADMM-Net for compressive sensing MRI. In: Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., Garnett, R. (eds.) Advances in Neural Information Processing Systems, vol. 29 (2016)
53. Zhang, J., Ghanem, B.: ISTA-Net: interpretable optimization-inspired deep network for image compressive sensing. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1828–1837 (2018)
54. Zhang, C., Bengio, S., Hardt, M., Recht, B., Vinyals, O.: Understanding deep learning requires rethinking generalization. In: International Conference on Learning Representations (2017)

# Chapter 9
# Angular Scattering Function Estimation Using Deep Neural Networks


**Yi Song and Giuseppe Caire**

## 9.1 Introduction

Massive multiple-input multiple-output (MIMO) systems are a building block of the next generation of wireless networks, promising unprecedented increase in spatial multiplexing capability, data rate, and link reliability [5, 15, 17]. Employing multiple-antenna elements at the base station (BS), the Angles of Arrival (AoA) of each signal component is associated with a random gain that depends on the scattering properties of the environment. Following the well-known and widely accepted uncorrelated scattering assumption, the angular stochastic process is uncorrelated in the angle domain, and its power density as a function of the AoA variable is referred to as the angular scattering function (ASF) of the channel.[1]

The ASF encapsulates highly valuable information about the propagation environment. It can be used for the purpose of channel sounding, to determine the angular position of scatterers and reflectors and to measure their relative gain power [8, 24, 28]. It can also facilitate user grouping and spatial multiplexing. Given the ASF information of a set of users in a cell, the BS can partition the user population into groups with (approximately) the same angular power profile to simplify user scheduling and to design channel precoders that alleviate inter-user interference (see

[1] Notice that a non-uniform ASF in the angle domain induces a correlation of the elements of the channel vector in the antenna domain, in analogy to the classical theory of time-domain processes, where a non-uniform power spectral density in the frequency domain corresponds to a certain autocorrelation function in the time domain via Fourier transformation. In the context of antenna/angle domain, the transform is related to the antenna array manifold.

Y. Song (✉) · G. Caire
Technical University of Berlin, Berlin, Germany
e-mail: yi.song@tu-berlin.de; caire@tu-berlin.de



G. Kutyniok et al. (eds.), *Compressed Sensing in Information Processing*, Applied and Numerical Harmonic Analysis, https://doi.org/10.1007/978-3-031-09745-4_9

[1, 2] and references therein). In this chapter, the ASF is also directly related to the channel covariance through an integral transform, and exploiting certain properties of the ASF (such as non-negativity over the angular range) results in an improved estimate of the channel covariance in noisy and limited pilot dimension scenarios [14]. In addition, the ASF proves to be an invariant property of the channel over close frequency bands, for example, the frequency bands dedicated to uplink (UL) and downlink (DL) transmission in a frequency-division duplex (FDD) system. This property is known as *angular channel reciprocity* and can be used both for UL–DL covariance transformation and, in special cases, for FDD channel prediction [13, 18].

However, in this chapter, we do not focus specifically on the above-mentioned applications but consider the following general ASF estimation problem: during the training phase, the BS receives a number of (noisy) pilot symbols from a user. Using these pilot observations, the BS has to compute a high-resolution estimate of the ASF, i.e., the distribution of the received signal power over a fine grid of angular bins. The number of such bins can be several times larger than the array dimension. This suggests that ASF estimation can be seen as a type of inverse problem where a high-dimensional vector has to be recovered from low-dimensional observations. Various methods in the literature attempt to solve this inverse problem, the most relevant of which are mentioned in the sequel.

### *9.1.1 Related Work*

The problem of estimating the ASF from a set of noisy pilot samples can be seen as a spectral estimation problem. Conventional methods of spectral estimation include MUSIC, ESPRIT, Prony and Pisarenko's harmonic estimation techniques [22, 23], which rely on the assumption that the spectrum contains only spectral lines (i.e., Dirac delta functions in the angle domain). Besides, compressive sensing techniques are able to estimate the spectrum either over the continuous angular domain [4, 25] or over a fine discrete angular grid [9, 26]. These methods, however, do not generalize to cases where the ASF is not sparse. Exploiting the non-negativity of the spectrum is a natural leverage for lifting and replacing the sparsity assumption. A recent method suggests estimating the ASF via a non-negative least-squares (NNLS) convex program that minimizes the least-squares error while enforcing the non-negativity on the parametric coefficients vector associated with the ASF [10, 14]. A drawback of using NNLS is that, despite the absence of any explicit sparsity constraint, it tends to result in sparse estimates of the [21] and hence may fail in accurately estimating a non-sparse ASF. The authors of [18] have instead proposed to estimate the ASF by solving a feasibility problem. The feasibility set is non-negative, and enforces the set of functions that generate a given covariance, and feasiblity problem can be solved using an ietrative method. A recent publication has further improved this method based on a prior information on the shape of the ASF [7]. In this case, one assumes that in addition to the covariance, the estimator has

access to a set of i.i.d. ASF realizations (the data set). An algorithm is developed that, among all feasible spectra, selects the one with the minimum distance to the expected value of the data set.

### 9.1.2 Outline and Notation

In Sect. 9.2, we explain the detail information of MIMO system and describe the relationships between noisy sample channel covariance matrix and ASF. In order to make a better use of ASF, then in Sect. 9.3, we come up with a parametric form of approximating ASF especially for diffuse ASF using density functions. Then we introduce the two-step algorithm for ASF estimation in Section and Sect. 9.5. Last, in Sect. 9.6, we summarize the technical details of the another two methods and compare them to our proposed estimator.

We use small and capital bold-faced letters ($\mathbf{x}$ and $\mathbf{x}$) to denote vectors and matrices, respectively. We denote the $i$-th element of $\mathbf{x}$ by $[\mathbf{x}]_i$. The complex Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$ is denoted by $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. $\texttt{rect}_{\mathbf{I}}$ with argument $x$ is a function that is equal to one for $x \in \mathbf{I}$ and is equal to zero for $x \notin \mathbf{I}$. $\mathbb{S}_{+}^{M\times M}$ denotes the set of positive semidefinite (PSD) matrices of size $M$, and the space of functions with bounded $\ell_2$-norm is represented by $L_2$. Moreover, we denote Hadamard product as $\odot$, so $(A \odot B)_{ij} = (A)_{ij}(B)_{ij}$.

## 9.2 System Model

Consider a BS equipped with a uniform linear array (ULA) of $M \gg 1$ antennas, serving a single-antenna user as shown in Fig. 9.1. We assume the popular block-fading channel model [27], where the $M$-dimensional channel vector at a specific resource block $s$ is given by a superposition of signals impinging on the array over a continuum of Angles of Arrival (AoAs):

$$\mathbf{h}(s) = \int_{-\theta_{\max}}^{\theta_{\max}} w(d\theta; s)\, \mathbf{a}(\theta), \tag{9.1}$$

where $\theta \in [-\theta_{\max}, \theta_{\max}]$ is the AoA parameter, $\theta_{\max} \in [0, \frac{\pi}{2}]$ is the maximum array aperture, $w(d\theta; s)$ is the random channel gain over the infinitesimal angular interval $[\theta, \theta+d\theta]$, and $\mathbf{a}(\theta) \in \mathbb{C}^M$ is the *array response* vector whose $m$-th element is given as $[\mathbf{a}(\theta)]_m = e^{j\frac{2\pi d}{\lambda} m \sin(\theta)}$, where $d$ denotes the antenna spacing and $\lambda$ is the carrier wavelength. For convenience, we assume the antenna spacing to be $d = \frac{\lambda}{2\sin(\theta_{\max})}$ and introduce the change of variables $\xi = \frac{\sin(\theta)}{\sin(\theta_{\max})}$. The array response in terms of the "normalized" AoA, $\xi \in [-1, 1]$, is given by

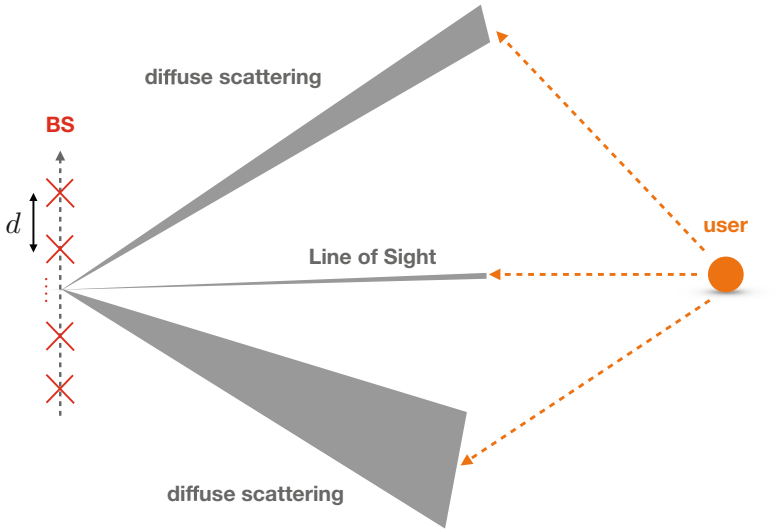



**Fig. 9.1** An example of a propagation environment with various types of scattering

$$\mathbf{a}(\xi) = [1, e^{j\pi\xi}, e^{j2\pi\xi}, \ldots, e^{j\pi(M-1)\xi}]^{\mathsf{T}} \in \mathbb{C}^M. \tag{9.2}$$

The channel representation in (9.1) is more general than the one typically assumed in the literature, in which the AoAs are restricted to belong to a finite, discrete, and not the continuum, implying that the channel is a superposition of signals coming from discrete, separable AoAs corresponding to specular and narrow scattering. In contrast, we further consider the possibility that diffuse scattering components exist in the environment that are associated with subsets of the angular continuum.

We take the channel gain $w(d\xi; s)$ to be a complex, circularly symmetric Gaussian process over the angular domain, with zero mean $\mathbb{E}[w(d\xi)] = 0$ and an autocorrelation function[2]

$$\mathbb{E}[w(d\xi)w(d\xi')^*] = \gamma(\xi)\delta(\xi - \xi'), \tag{9.3}$$

where $\gamma(\xi) : [-1, 1] \to \mathbb{R}_+ \cup \{0\}$ denotes the ASF, representing the power received at the array from the angular interval $[\xi, \xi + d\xi]$. The physical meaning of (9.3) is that the scattering gains over angles $\xi$ and $\xi'$ are uncorrelated. Using (9.3), we can compute the channel covariance matrix as follows:

[2] The parameter $s$ both in $w(d\theta; s)$ and $\mathbf{h}(s)$ highlights the fact that the coefficients may vary from one resource block to another. We assume the statistical properties of these random variables, such as mean and covariance, to be constant over a much longer time horizon i the order of tens or hundreds of resource blocks. Hence, when we study the statistical properties of these random variables, we drop the parameter $s$ from the argument.

$$\mathbf{\Sigma} = \mathbb{E}[\mathbf{h}\mathbf{h}^{\mathsf{H}}] = \mathbb{E}\left[\int_{-1}^{1}\int_{-1}^{1} w(d\xi)w(d\xi')^{*}\mathbf{a}(\xi)\mathbf{a}(\xi')^{\mathsf{H}}\right] \tag{9.4}$$

$$= \int_{-1}^{1} \gamma(\xi)\mathbf{a}(\xi)\mathbf{a}(\xi)^{\mathsf{H}} d\xi. \tag{9.5}$$

Note how the ASF $\gamma$ is mapped to the covariance $\mathbf{\Sigma}$ through an integral transform. More concretely, we have $\mathcal{M}(\gamma) = \mathbf{\Sigma}$, where $\mathcal{M} : \mathcal{D} \to \mathbb{S}_{+}^{M\times M}$, $f \to \int_{-1}^{1} f(\xi)\mathbf{a}(\xi)\mathbf{a}(\xi)^{\mathsf{H}} d\xi$ is a mapping that takes a distribution from the space of generalized functions $\mathcal{D}$ as input and outputs a PSD $M \times M$ covariance matrix through the integral transform in (9.4). Unfortunately, the inverse map $\mathcal{M}^{-1}$ from the covariance to the ASF is non-unique, which means that multiple ASFs can generate the same covariance. Nevertheless, with additional information on the ASFs, given the covariance, one can single out the true ASF. Broadly speaking, our goal in this work is to "learn" an inverse mapping from the channel covariance to the ASF by assuming the ASF to belong to a certain class of distributions. This class is not fixed and can be modified from one environment to another and from time to time.

In practice, the covariance must be estimated by the BS through observing the pilot signals received from the user that are given by $\mathbf{y}(s) = \mathbf{h}(s)x_s + \mathbf{z}(s)$, $s \in [T]$, where $T$ is the total pilot dimension, $x_s = 1$ is the pilot symbol here assumed to be equal to one for simplicity, and $\mathbf{z} \sim (\mathbf{0}, N_0\mathbf{I})$ is the additive white Gaussian noise (AWGN). Pilots are transmitted on resource blocks with sufficiently large separation in time and frequency so that we can safely assume the channel realizations $\mathbf{h}(s)$ $s \in [T]$ to be statistically independent. Given the noise variance $N_0$, a simple estimate of the channel covariance is given by the sample covariance matrix as

$$\widetilde{\mathbf{\Sigma}} = \frac{1}{T}\sum_{s\in[T]} \mathbf{y}(s)\mathbf{y}(s)^{\mathsf{H}} - N_0\mathbf{I}. \tag{9.6}$$

In the case of a ULA, one can improve the sample covariance estimator by imposing additional structure on the estimated covariance. The covariance of a ULA channel is a Toeplitz, Hermitian PSD matrix that is fully expressed by its first column, i.e., $\mathbf{\Sigma} = \mathcal{T}(\boldsymbol{\sigma})$, where $\mathcal{T}$ is an operator that outputs the Toeplitz, Hermitian matrix whose first column is the input $\boldsymbol{\sigma} \in \mathbb{C}^M$. Therefore, an improved estimate can be computed by solving the following semidefinite program:

$$\underset{\boldsymbol{\sigma}\in\mathbb{C}^M}{\text{minimize}} \quad \|\mathcal{T}(\boldsymbol{\sigma}) - \widetilde{\mathbf{\Sigma}}\|_F, \quad \text{subject to} \quad \mathcal{T}(\boldsymbol{\sigma}) \succeq \mathbf{0}. \tag{9.7}$$

Denoting the solution of (9.7) with $\widehat{\boldsymbol{\sigma}}$, we have an estimate of the covariance from noisy pilot observations as $\widehat{\mathbf{\Sigma}} = \mathcal{T}(\widehat{\boldsymbol{\sigma}})$. In what follows, we consider the problem of estimating the ASF $\gamma$ from the channel covariance estimate $\widehat{\mathbf{\Sigma}}$.

## 9.3 The Parametric Form of ASF

In order to learn the inverse mapping from the estimated channel covariance to the ASF, we should impose some structure on the ASF to restrict the set of admissible solutions. In general, the ASF is made up of two types of components: (1) discrete components, representing the power coming from Line-of-Sight (LoS) and specular scattering, denoted as spikes at specified locations and (2) continuous components,[3] associated with the diffuse scattering, denoted as a certain power distribution within the regime in theta domain. Therefore, we consider a decomposition of the ASF in the form

$$\gamma(\xi) = \gamma_c(\xi) + \gamma_d(\xi), \tag{9.8}$$

where $\gamma_c$ and $\gamma_d$ denote continuous and discrete ASF components, respectively.

### *9.3.1 The Continuous ASF Component*

The continuous component of the ASF can be expressed as a superposition of functions that do not contain delta impulses. Therefore, this component takes on the following form:

$$\gamma_c(\xi) = \sum_{k=1}^{K} f_k(\xi), \tag{9.9}$$

where, as explained, each component function $f_k$, $k \in [K]$ is associated with a diffuse scattering element (see Fig. 9.2).

In order to obtain a finite-dimensional representation of $\gamma_c$, we can approximate it with a linear combination of pre-defined, limited-support densities (kernels) and establish an equivalence between $\gamma_c$ and the coefficients of the approximation. Specifically, let us define a family of densities as $\Psi = \{\psi_i : i \in [G_c]\}$ with cardinality $G_c$. One can design such a family in various ways, and we choose the following simple option. Let $\psi^\star : [-1, 1] \to \mathbb{R}_+ \cup \{0\}$ be a real, positive function whose most support is limited to $[0, \frac{2}{G_c}]$, and define the density family to be consisting of shifted versions of $\psi^\star$, i.e., $\psi_i(\xi) = \psi^\star(\xi + 1 - \frac{2i}{G_c})$, $i \in [G_c]$, $\xi \in [-1, 1]$. In particular, we use one density function, i.e.,

[3] We refer to the part of ASF not containing delta functions as "continuous component" in analogy with the probability density function of continuous random variables. This does not mean that the density is continuous (e.g., a rectangular function is not), but that its anti-derivative function (i.e., the corresponding cumulative distribution function) is continuous.

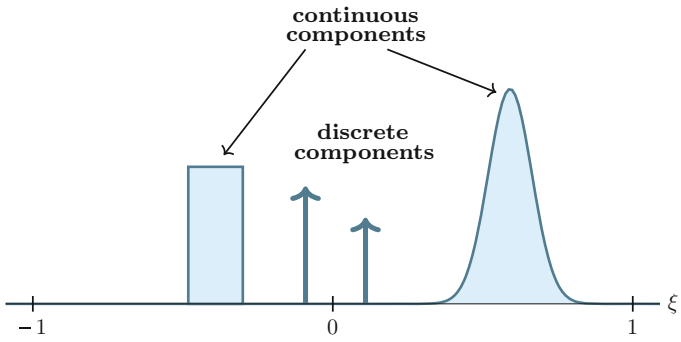



**Fig. 9.2** An example ASF with discrete components (the two spikes) and continuous components (the bell-shaped and rectangular functions)

- *Rectangular densities:* In this case, we define $\psi^\star$ as a rectangular pulse over $[0, \frac{2}{G_c}]$, that is, $\psi^\star(\xi) = \frac{G_c}{2}\mathbf{I}_{\{\xi \in [0, \frac{2}{G_c}]\}}$.

Now, given that the function is large enough ($G_c \gg 1$), we can closely approximate $\gamma_c$ as

$$\gamma_c(\xi) \approx \sum_{i \in [G_c]} b_i \psi_i(\xi), \tag{9.10}$$

where $b_i,\ i \in [G_c]$ are appropriate approximation coefficients.

**Generating Approximation Coefficients**
We can compute the continuous component's approximation coefficients by solving the following optimization problem:

$$\underset{b_i}{\text{minimize}} \quad \|\gamma_c - \sum_{i}^{G_c} b_i \psi_i\|_2 \tag{9.11}$$

$$\text{subject to} \quad b_i \geq 0,\ i = 1, \ldots, G_c. \tag{9.12}$$

When kernel functions $\psi$ are orthogonal, such as rectangular kernel, we can obtain the continuous component's coefficients simply by computing the inner product of the generated continuous ASF with each element of the density family. In other words, we have

$$b_i = \frac{\langle \gamma_c, \psi_i \rangle}{\langle \psi_i, \psi_i \rangle} = \frac{1}{\|\psi_i\|^2} \int_{-1}^{1} \gamma_c(\xi)\psi_i(\xi) d\xi, \tag{9.13}$$

where $b_i$ denotes the energy of ASF in $i$-th grid.

### 9.3.2 The Discrete ASF Component

Unlike the continuous component, the discrete ASF component has a finite-dimensional parametric expression as a train of weighted spikes, each representing the power coming from an LoS or specular scattering element. Formally, it can be written as

$$\gamma_d(\xi) = \sum_{\ell \in [L]} \rho_\ell \delta(\xi - \xi_\ell), \tag{9.14}$$

where $\rho_\ell \geq 0$, $\ell \in [L]$ are real, non-negative scalar coefficients and $\xi_\ell$, $\ell \in [L]$ represent the discrete AoAs.

The decomposition of the ASF into discrete and continuous components, as given by (9.10) and (9.14), results in a decomposition of the corresponding covariance matrix, given by (9.4). Due to the Hermitian, Toeplitz structure of the channel covariance of a ULA, we can reformulate the matrix identity (9.4) to the vector identity

$$\boldsymbol{\sigma} = \int_{-1}^{1} \gamma(\xi)\mathbf{a}(\xi)d\xi \tag{9.15}$$

$$\approx \sum_{i \in [G_c]} b_i \widetilde{\mathbf{a}}_i + \sum_{\ell \in [L]} \rho_\ell \mathbf{a}(\xi_\ell), \tag{9.16}$$

where $\boldsymbol{\sigma} = \boldsymbol{\Sigma}_{:,1}$ is the first column of the covariance matrix, and the vector $\widetilde{\mathbf{a}}_i \in \mathbb{C}^M$ is defined as $\widetilde{\mathbf{a}}_i = \int_{-1}^{1} \psi_i(\xi)\mathbf{a}(\xi)d\xi$.

A difference between the two representations in (9.10) and (9.14) is that the density functions $\psi_i$, $i \in [G_c]$ (correspondingly, the vectors $\mathbf{a}_i$, $i \in [G_c]$) are known a priori, while the discrete AoAs $\xi_\ell$, $\ell \in [L]$ (correspondingly, the vectors $\mathbf{a}(\xi_\ell)$) are not known. Given the discrete AoAs, the ASF estimation problem reduces to the task of estimating the coefficients $b_i$, $i \in [G_c]$ and $\rho_\ell$, $\ell \in [L]$ from an estimate of $\boldsymbol{\sigma}$, obtained from the noisy pilot measurements. In order to realize this simple form of the problem, as a pre-processing step, we propose first estimating the discrete AoAs.

## 9.4 Pre-processing: Discrete AoA Estimation via MUSIC

In order to estimate the support of the discrete ASF component $\gamma_d$, from the pilot measurements $\mathbf{y}(s)$, $s \in [T]$, we employ the well-known multiple signal classification (MUSIC) method [20, 22]. As a super-resolution method, MUSIC is typically used for estimating the frequencies of multiple sinusoids from their (possibly noisy) mixture. Similarly, here we use MUSIC to estimate the angular

tones $\xi_\ell,\ \ell \in [L]$. Recall the expression of the sample covariance matrix in (9.6) and define its eigendecomposition as $\widetilde{\mathbf{\Sigma}} = \widetilde{\mathbf{U}}\widetilde{\mathbf{\Lambda}}\widetilde{\mathbf{U}}^{\mathsf{H}}$, where $\widetilde{\mathbf{U}} = [\widetilde{\mathbf{u}}_1, \ldots, \widetilde{\mathbf{u}}_M]$ denotes the eigenvectors matrix, and $\widetilde{\mathbf{\Lambda}}$ is the diagonal eigenvalue matrix, with its diagonal elements ordered as $\widetilde{\lambda}_1 \geq \widetilde{\lambda}_2 \geq \ldots \geq \widetilde{\lambda}_M$. Assume that an estimate of the number of discrete AoAs ($\widehat{L}$) is given (normally, we can set $\widehat{L} > L$[4]). Then, we can define the so-called noise subspace as the subspace spanned by the columns of the $(M-\widehat{L})$ eigenvectors in $\widetilde{\mathbf{U}}$ corresponding to the smallest $(M-\widehat{L})$ eigenvalues in $\widetilde{\mathbf{\Lambda}}$, namely the columns of $\widetilde{\mathbf{U}}_{\text{noise}} = \left[\widetilde{\mathbf{u}}_{\widehat{L}+1}, \ldots, \widetilde{\mathbf{u}}_M\right]$ MUSIC estimate the discrete AoAs by finding the $\widehat{L}$ dominant minimizers of the *pseudo-spectrum* function

$$\eta(\xi) = \|\widetilde{\mathbf{U}}_{\text{noise}}^{\mathsf{H}}\mathbf{a}(\xi)\|^2 = \sum_{\ell=\widehat{L}+1}^{M} \left|\widetilde{\mathbf{u}}_\ell^{\mathsf{H}}\mathbf{a}(\xi)\right|^2. \tag{9.17}$$

We denote the estimated discrete AoAs as $\widehat{\xi}_\ell,\ \ell \in [\widehat{L}]$.

When the observations are generated by a noisy superposition of a finite number of weighted tones, MUSIC asymptotically gives consistent estimates of the tones. In the context of our problem, this scenario translates to the case in which the ASF consists of only a discrete component, and the tones are the discrete AoAs. However, in general, the channel is not only a product of the discrete ASF component, but a mixture of discrete and continuous components. A recent asymptotic result has shown that also in this case, under some mild conditions on the energy distribution of the discrete and continuous parts as well as the signal dimension, MUSIC is able to consistently estimate the discrete AoAs [19]. The following theorem states a slightly modified version of this result to justify the expected success of MUSIC in identifying discrete ASF AoAs from the noisy pilot observations.

**Theorem 9.1** *Consider an ASF* $\gamma(\xi) = \gamma_d(\xi) + \gamma_c(\xi) := \sum_{\ell=1}^{L} \rho_\ell^{(M)}\delta(\xi - \xi_\ell) + \gamma_c(\xi)$ *and assume that the weights* $\{\rho_\ell^{(M)} : \ell \in [L]\}$ *may depend on the number of antennas (M). Consider a scaling regime where the number of antennas M and the sample size T both approach infinity such that* $\frac{T}{M} \to \nu > 0$. *Then, MUSIC is asymptotically consistent, i.e.,* $\max_{\ell\in[L]} M\|\widehat{\xi}_\ell - \xi_\ell\| \to 0$ *provided that* $\limsup_{M\to\infty} \min_{\ell\in[L]} M\rho_\ell^{(M)} \geq \omega_0(\nu, \gamma_c)$, *where* $\omega_0(\nu, \gamma_c)$ *is a finite parameter that depends on* $\nu$ *and the continuous component* $\gamma_c$.

***Proof*** Further proof is presented in [14]. □

Given an estimate of the discrete AoAs $\xi_\ell,\ \ell \in [L]$, we have a model that relates the $(G_c + \widehat{L})$-dimensional vector of real, positive coefficients to the covariance matrix via (9.15). The goal now is to find the reverse mapping, mapping from the first column of the covariance matrix to the $(G_c + \widehat{L})$-dimensional coefficients vector.

[4] Since Remark 1 in the next section to see why our method can easily tolerate the a priori overestimation of the number of spikes in the MUSIC pre-processing phase.

**Estimating the Number of Discrete AoAs**

We employ a heuristic method for estimating the number of discrete AoAs [14]. This method uses the spectrum of the sample covariance matrix $\widetilde{\mathbf{\Sigma}}$ to find the number of discrete AoAs. First, let us normalize the eigenvalues of $\widetilde{\mathbf{\Sigma}}$ to their largest and define the new variables $\beta_i = (\frac{\widetilde{\lambda}_i}{\widetilde{\lambda}_1})^{\frac{1}{2}} \in [0, 1]$ for $i = 1, \ldots, M$. The exponent $\frac{1}{2}$ has the role of inducing a separation between large and small eigenvalues. The reason is that the function $f : [0, 1] \to \mathbb{R}$, $x \to x^{\frac{1}{2}}$ does not affect the largest eigenvalue or smallest eigenvalue but "soft-truncates" the eigenvalues by mapping them closer to $f(1) = 1$.

We apply the K-means clustering algorithm with $K = 2$ clusters to the set of (scalar) normalized parameters $\beta_i : i = 1, ..., M$. The cluster centers are initialized to two values uniformly drawn at random from the interval [0, 1]. Assume that K-means converges to clusters with centers $c_1^{(\infty)} \leq c_2^{(\infty)}$. We estimate the number of discrete AoAs by the cardinality of the cluster associated with the larger center $c_2^{(\infty)}$, i.e.,

$$\widehat{L} = \left| \{ i \in [M] : |\beta_i - c_2^{(\infty)}| \leq |\beta_i - c_1^{(\infty)}| \} \right|. \tag{9.18}$$

We repeat the K-means algorithm with newly generated initial centers, each time calculating a new $\widehat{L}$, and eventually we take the mode (most repeated member) of this sequence of estimates as the ultimate estimate of the number of discrete AoAs.

*Remark 9.1* Note that the precise estimation of the number of discrete AoAs is *not* critical, and in particular, it is better to overestimate the number of discrete AoAs, than to underestimate it. If we overestimate the number of discrete AoAs, there will be "fake" spikes identified in the support of the discrete ASF component. However, the network (as will be introduced shortly) will assign small coefficients to the fake spikes, which practically means that there is no spike. Underestimating the number of discrete AoAs can be more harmful since some of the existing spikes will not be represented to the network. Nevertheless, even in this case, the network can assign a non-zero coefficient to an element of the density dictionary $\Psi$ that has the highest correlation with the "missed" spike. This is obviously sub-optimal, due to the poor approximation of the delta function with a continuous density, but the induced error will be controlled.

## 9.5 A Deep Learning Approach to ASF Estimation

We address the problem of estimating the ASF parameters using a novel deep learning approach. Recall that the BS receives a number of $T$ noisy pilot observations in the form $\mathbf{y}(s) = \mathbf{h}(s) + \mathbf{z}(s)$, $s \in [T]$, upon which it computes an estimate of the first column of the covariance using (9.7). In addition, the discrete AoAs are

estimated using MUSIC, as explained in the previous section. Therefore, we can naturally consider a network that takes as input the estimated covariance column $\widehat{\boldsymbol{\sigma}}$ plus the estimated discrete AoAs $\xi_\ell,\ \ell \in [\widehat{L}]$ and outputs the coefficient parameters that approximate the ASF, namely the variables $b_i,\ i \in [G_c]$ and $\rho_\ell,\ \ell \in [\widehat{L}]$. For simplicity, we define a new vector variable that contains all the coefficients as $\mathbf{c} = [b_1, \ldots, b_{G_c}, \rho_1, \ldots, \rho_{\widehat{L}}]^\mathsf{T} \in \mathbb{R}_+^{G_c+\widehat{L}}$.

Recently, deep neural networks have been studied as tools for solving inverse problems by applying encoder–decoder network or generative adversarial network [3, 6, 16]. Therefore, a deep network can also be employed for the purpose of ASF estimation. Such a network takes in the covariance column as well as estimated locations of the discrete AoAs as input and outputs the ASF coefficients vector $\boldsymbol{\gamma}$. In what follows, we describe the details of the network architecture and the method used for constructing data (for training and validation).

### *9.5.1 Training Phase*

In the training phase, the true locations of spikes are given as input in DNN, while in the testing phase, the estimated locations of the spikes are provided by MUSIC as explained before. Therefore, the training examples consist of input–output pairs

$$\left(\{\widehat{\boldsymbol{\sigma}}^{(t)}, \{\xi_\ell^{(t)}\}_{\ell=1}^{\widehat{L}}\}, \boldsymbol{\gamma}^{(t)}\right), \quad t = 1, \ldots, T_{\text{train}}, \tag{9.19}$$

where the superscript $t$ denotes the example index and $\widehat{L}$ is a suitably chosen integer, which will be discussed shortly. Moreover, when $\widehat{L} > L$, some spurious locations other than the actual spikes generating $\widehat{\sigma}$ are given to the network, with corresponding power set to zero. This allows the DNN to learn also the cases when MUSIC overestimates the number of spikes. In order to generate an input–output training pair, we take the following steps:

1. **Generating an ASF:** We generate an example ASF in a semi-random fashion:

   - **Discrete ASF**: The discrete ASF component $\gamma_d^{(t)}$ is produced by choosing $L$ locations uniformly at random over the interval $[-1, 1]$ plus $\widehat{L} - L$ spurious locations with zero power. $L$ is chosen between $L_{\min}$ and $L_{\max}$, where $L_{\min}$, $L_{\max}$ denote a presumed value for the minimum and maximum numbers of discrete AoAs present in the ASF. To each random AoA, we assign a real, non-negative coefficient that is generated uniformly at random over $[\rho_{\min}, \rho_{\max}]$, where $\rho_{\max}$ is a pre-defined bound on the maximum amplitude for a single discrete ASF component, and $\rho_{\min}$ is the bound for minimum amplitude. More concretely, we have

$$\gamma_d^{(t)}(\xi) = \sum_{\ell=1}^{L} \rho_\ell^{(t)} \delta(\xi - \xi_\ell^{(t)}), \tag{9.20}$$

where $\xi_\ell^{(t)} \sim \mathbb{U}([-1, 1])$ and $\rho_\ell^{(t)} \sim \mathbb{U}([\rho_{\min}, \rho_{\max}])$, independently for $\ell = 1, \ldots, L$.

- **Diffuse ASF**: The continuous ASF component $\gamma_c^{(t)}$ is generated according to (9.9), i.e., by using the *synthesis* expression $\gamma_c^{(t)}(\xi) = \sum_{k=1}^{K} f_k^{(t)}(\xi)$. Similar to $L$, the integer $K$ is a pre-defined parameter on the number of diffuse scatterers ranging from $K_{\min}$ to $K_{\max}$. The functions $f_k$ are arbitrary non-negative, real, continuous functions. However, for the sake of training the network, it is practically easier to assume $f_k^{(t)}$ to take on a certain parametric shape. A natural choice is to allow $f_k^{(t)}$ to be either a rectangular or a Gaussian distribution. For each $k$, an equi-probable Bernoulli random variable decides whether $f_k^{(t)}$ has a rectangular or a Gaussian form. If $f_k^{(t)}$ is decided to be Gaussian, then we have

$$f_k^{(t)}(\xi) = a_k^{(t)} e^{-\frac{\left(\xi - \mu_k^{(t)}\right)^2}{2\sigma_k^{(t)\,2}}}, \tag{9.21}$$

where $a_k^{(t)} \sim \mathbb{U}([a_{min}, a_{\max}])$ is a random, uniformly generated, non-negative coefficient, $\mu_k^{(t)} \sim \mathbb{U}([\mu_{\min}, \mu_{\max}])$ is the randomly generated distribution mean, and $\sigma_k^{(t)\,2} \sim \mathbb{U}([\sigma_{\min}^2, \sigma_{\max}^2])$ is the distribution variance. The parameters $a_{\max}$ and $\sigma_{\max}^2$ represent the maximum coefficient amplitude and the maximum variance, respectively. In contrast, the minimum are defined as $a_{\min}$ and $\sigma_{\min}^2$. Similarly, if $f_k^{(t)}$ is decided to be rectangular, we have

$$f_k^{(t)}(\xi) = a_k^{(t)} \mathbf{I}_{\{\xi \in [\mu_k^{(t)} - \sigma_k^{(t)}/2, \mu_k + \sigma_k^{(t)}/2]\}}, \tag{9.22}$$

where in this case $\mathbf{I}$ is an indicator function; if the condition is fulfilled, it is one; otherwise, it is zero. $a_k^{(t)} \sim \mathbb{U}([a_{\min}, a_{\max}])$ is a random coefficient as before, $\sigma_k^{(t)}$ denotes the width of the rectangular function, generated as $\sigma_k^{(t)\,2} \sim \mathbb{U}([\sigma_{\min}^2, \sigma_{\max}^2])$, and $\mu_k^{(t)} \sim \mathbb{U}([\mu_{\min}, \mu_{\max}])$ is its support mean.
We can set the parameters involved in the semi-random generation of the ASF based on a priori information about the communication environment. For example, the BS can learn the number of LoS and specular scattering paths and set a value for $L_{\max}$. Also, an upper bound on the sparsity order of the channels in an environment translates to a bound on $K_{\max}$ as well as the maximum scatterer (effective) width $\sigma_k$.
- **Normalization**: Once $\gamma_d^{(t)}$ and $\gamma_c^{(t)}$ are generated, we normalized them to a scalar factor such that they both have unit integral over the interval $[-1, 1]$. In other words, we have $\int_{-1}^{1} \frac{1}{Z_d} \gamma_d^{(t)}(\xi) d\xi = 1$ and $\int_{-1}^{1} \frac{1}{Z_c} \gamma_c^{(t)}(\xi) d\xi = 1$, where

$Z_d$ and $Z_c$ are the normalization factors for the discrete and continuous ASF components, respectively. For simplicity, we drop the normalization factors from here onward, assuming that each component is normalized.

- Finally, the ASF is given by the superposition of the discrete and continuous parts as

$$\gamma^{(t)}(\xi) = \alpha^{(t)}\gamma_c^{(t)}(\xi) + (1-\alpha^{(t)})\gamma_d^{(t)}(\xi), \tag{9.23}$$

where $\alpha^{(t)} \in [\alpha_{\min}, \alpha_{\min}]$ is a parameter that controls the contribution of each component: if $\alpha^{(t)} = 0$, the ASF is purely discrete, and if $\alpha^{(t)} = 1$, the ASF is purely continuous, and for all other values, the ASF is a convex combination of discrete and continuous parts.

In this way, there are a variety of examples, where in some the discrete component is dominant, in some the continuous component is dominant, and in the two components, they have a balanced contribution to the overall ASF.

2. **Generating the Associated Coefficients Vector:** According to section "Generating Approximation Coefficients", the coefficient vectors can be computed, and then $[\mathbf{c}^{(t)}]_i = b_i^{(t)}$ for $i = 1, \ldots, G_c$ and $[\mathbf{c}^{(t)}]_i = \rho_{i-G_c}$ for $i = G_{c+1}, \ldots, G_{c+\widehat{L}}$. In this way, after normalization, it is determined that the sum of each components in $\mathbf{c}^{(t)}$ is 1, i.e., $\sum_i^{G_c+\widehat{L}}[\mathbf{c}^{(t)}]_i = 1$.
3. **Generating the Noisy Covariance Column:** With the generated ASF, the covariance column $\boldsymbol{\sigma}^{(t)}$ corresponding to it can be computed by simply using (9.15), but with the true ASF instead of dictionary-based ASF. The corresponding noisy sample covariance based on sample number $T$ can be computed from (9.6). In order to reduce the complexity of DNN, the transformed column $\widehat{\boldsymbol{\sigma}}$ of noisy sample covariance matrix is obtained from (9.7), which is the part of input in DNN.

With these three steps, the training input–output examples in (9.19) can be produced.

### 9.5.2 Network Architecture

We propose a fully connected neural network with three components for ASF estimation, as illustrated in Fig. 9.3. First, we learn an intermediate representation from noisy sample covariance matrix by a linear transformation $f : \mathbb{R}^{2\times M+\widehat{L}} \to \mathbb{R}^N$, which is made of one-layer neural network with input denoted as $[Real(\widehat{\boldsymbol{\sigma}}), Imag(\widehat{\boldsymbol{\sigma}})] \in \mathbb{R}^{2M}$ plus the estimated locations of spikes, and then the intermediate representation is denoted as

$$\mathbf{v} = f(\widehat{\boldsymbol{\sigma}}, \{\xi_\ell\}_{\ell=1}^{\widehat{L}})$$

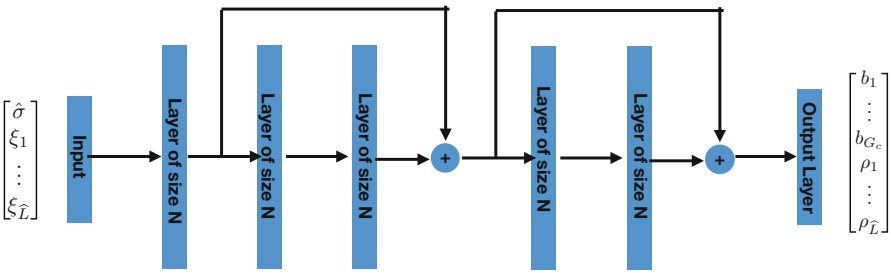



**Fig. 9.3** A schematic of our proposed network

$$= \mathbf{W}_i * [\widehat{\boldsymbol{\sigma}}, \{\xi_\ell\}_{\ell=1}^{\widehat{L}}] + \mathbf{b}_i. \tag{9.24}$$

The second component is the residual learning block, a small modification in the network architecture introduced in [11], which consists of two-layer non-linear function $h : \mathbb{R}^N \to \mathbb{R}^N$ and an addition operation. Therefore, the output of residual block can be denoted as $R(\mathbf{x}) = ReLU(\mathbf{x} + h(\mathbf{x}))$, where $ReLU$ is an activation function as $\max(\mathbf{x}, 0)$. Specifically, the non-linear function $h$ can be denoted as

$$h(\mathbf{x}) = BN(\mathbf{W}_2 * (ReLU(BN(\mathbf{W}_1 * \mathbf{x})))), \tag{9.25}$$

where $BN$ represents the batch normalization [12], which first normalizes the output into standard Gaussian distribution and rescales the output during training. In the experiment, we will use two residual blocks.

Last, the estimated coefficients $\hat{\mathbf{c}}$ can be obtained from output layer denoted as

$$\hat{\mathbf{c}} = g(\mathbf{v}) = Sigmoid(\mathbf{W_o} * \mathbf{y} + \mathbf{b_o}), \tag{9.26}$$

where $\mathbf{y}$ is the output of residual block, and $g : \mathbb{R}^N \to \mathbb{R}^{G_c+\widehat{L}}$ represents one-layer transformation function with sigmoid activation function as $Sigmoid(\mathbf{x}) = \frac{1}{\exp(-\mathbf{x})+1}$. Moreover, we apply sigmoid function to obtain estimate coefficients ranging between 0 and 1.

Furthermore, the binary cross-entropy loss function is applied to the output layer to compare the differences between $\hat{\mathbf{c}}$ and $\mathbf{c}$, as shown in

$$Loss = \sum_{i}^{G_c+\widehat{L}} -c_i * \log(\hat{c}_i) - (1 - c_i) * \log(1 - \hat{c}_i). \tag{9.27}$$

During training, we use the widely adopted batch gradient descent (Batch GD) for optimizer, and Adam optimizer proposed in [26] to update model parameters, whose learning rate is 0.0002.

### 9.5.3 Test Phase

Once the network parameters are optimized, we test it by feeding input examples designed in the following way. First, a number of $T$ noisy pilot samples are received $\mathbf{y}(s) = \mathbf{h}(s) + \mathbf{z}(s)$, $s = 1, \ldots, T$, where $\mathbf{h}(s) \sim (\mathbf{0}, \boldsymbol{\Sigma})$ and $\mathbf{z}(s) \sim (\mathbf{0}, N_0\mathbf{I})$, where the component-wise noise variance $N_0$ relates to the SNR as $N_0 = \text{trace}(\boldsymbol{\Sigma})/(M\,\text{SNR})$. Assuming that the noise variance is known, we estimate the channel covariance by first computing the sample covariance $\widehat{\boldsymbol{\Sigma}}$ according to (9.6) and then improving the estimate via (9.7). The outcome is an estimate of the first column of the channel covariance, which we denote by $\widehat{\boldsymbol{\sigma}}$. Besides the covariance column, we use the sample covariance matrix to obtain estimates of the discrete AoAs, using the MUSIC method as explained in Sect. 9.4. The number of estimated discrete AoAs is upper-bounded by $\widehat{L}$, which is a fixed network parameter. Therefore, the input to the network during testing is the set of parameters $\left(\widehat{\boldsymbol{\sigma}}, \{\widehat{\xi}_\ell\}_{\ell=1}^{\widehat{L}}\right)$.

The output of the network is an estimate of the ASF coefficient parameters, namely the vector $\boldsymbol{\gamma}$ of dimension $G_c + \widehat{L}$, where the first $G_c$ elements are estimates of the coefficients in the approximation formula (9.10) and the last $\widehat{L}$ elements are estimates of the spike coefficients in (9.14). The ASF estimate is then given as

$$\widehat{\gamma}(\xi) = \sum_{i=1}^{G_c} [\hat{\mathbf{c}}]_i \psi_i(\xi) + \sum_{\ell=1}^{\widehat{L}} [\hat{\mathbf{c}}]_{G_c+\ell} \delta(\xi - \widehat{\xi}_\ell). \tag{9.28}$$

We assess the performance of the network in terms of quantitative and qualitative measures with respect to this estimate of the ASF.

## 9.6 Simulation Results

In this section, we provide empirical results to compare the performance of our proposed DNN-based ASF estimator with the state-of-the-art methods in the existing literature.

**1. Non-negative Least Squares (NNLS)** In the method proposed in [14] by some of the authors of the present paper, given $\boldsymbol{\sigma}$, the vector of ASF coefficient parameters $\boldsymbol{\gamma}$ is estimated by solving the following NNLS program:

$$\text{minimize}_{\mathbf{x}} \; \|\mathbf{W}\mathbf{D}\mathbf{x} - \boldsymbol{w} \odot \boldsymbol{\sigma}\| \;\; \text{subject to} \;\; \mathbf{x} \geq 0, \tag{9.29}$$

where $\mathbf{D} = \left[\widetilde{\mathbf{a}}_1, \ldots, \widetilde{\mathbf{a}}_M, \mathbf{a}(\widehat{\xi}_1), \ldots, \mathbf{a}(\widehat{\xi}_{\widehat{L}})\right]$, where $\widetilde{\mathbf{a}}_m = \int_{-1}^{1} \psi_m(\xi)\mathbf{a}(\xi)d\xi$, $m = 1, \ldots, M$ and $\{\widehat{\xi}_\ell\}_{\ell=1}^{\widehat{L}}$ are the discrete AoAs estimated by MUSIC. Moreover, since $\boldsymbol{\sigma}$ refers to the first column of channel covariance matrix, in order to

reconstruct the whole matrix, some weights $\boldsymbol{w}$ are added in the optimization problem. Therefore, $\boldsymbol{w} = [\sqrt{M}, \sqrt{2*(M-1)}, \sqrt{2*(M-2)}, ..., \sqrt{2}]$, and $\mathbf{W}$ is a diagonal matrix whose diagonal entries are $\boldsymbol{w}$. Then the estimated ASF is given by $\widehat{\gamma} = \sum_{i=1}^{M} [\mathbf{x}]_m \psi_m(\xi) + \sum_{\ell=1}^{\widehat{L}} [\mathbf{x}]_{M+\ell} \delta(\xi - \widehat{\xi}_\ell)$.

**2. Convex Projection Method** As mentioned in Sect. 9.1, in [18] a method for the ASF estimation is proposed by solving a convex feasibility problem of the form

$$\widehat{\gamma} = \text{ find } \mu \quad \text{subject to } \gamma \in \mathcal{S}, \tag{9.30}$$

where

$$\mathcal{S} = \{\gamma : \int_{-1}^{1} \gamma(\xi) e^{j\pi m\xi} d\xi = [\widehat{\sigma}]_m, \ m = 0, \ldots, M-1, \\ \gamma(\xi) \geq 0 \text{ for all } \xi \in [-1, 1]\}, \tag{9.31}$$

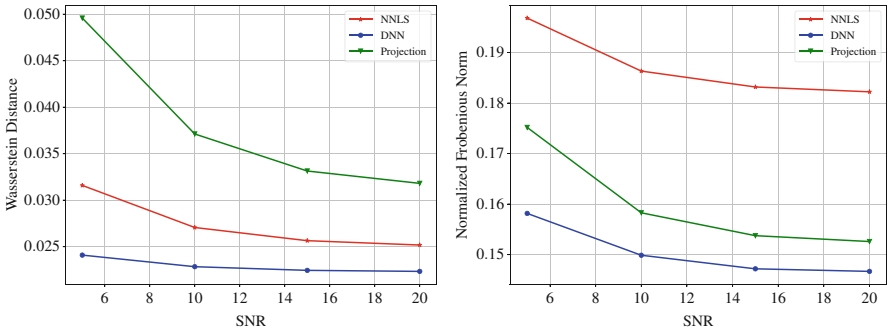
which can be solved by applying an iterative projection algorithm. Such algorithms produce a sequence of functions in $L_2$ that converges to a function that satisfies the constraints of (9.30), namely, consistency with the estimated covariance column derived from (9.7) and non-negativity.

### 9.6.1 Metrics for Comparison

In order to access the reconstruction performance of ASF, on one hand, we can compare those methods directly by computing the Wasserstein distance between estimate ASF and true ASF. On the other hand, we can map estimate ASF into the channel covariance and compare the estimate channel covariance by computing normalized Frobenius norm error, denoted as $\frac{\|\mathbf{\Sigma} - \hat{\mathbf{\Sigma}}\|_2}{\|\mathbf{\Sigma}\|_2}$.

### 9.6.2 Performance with Different SNRs

In order to evaluate the performance of reconstructing ASF based on different SNRs, we generate 5000 pairs of data when sample number $T = 1 \times M$, and SNR $= \{5, 10, 15, 20\}$ dB. Moreover, the estimate ASFs obtained by our proposed algorithm are produced by a pre-trained model that is trained on $T/M = 1$ and SNR $= 10$ dB. As illustrated in Fig. 9.4, where Fig. 9.4a shows the Wasserstein distance on estimate ASFs and Fig. 9.4b depicts the normalized Frobenius norm on reconstructed channel covariance matrix produced by estimate ASFs, it shows that when sample number is $M$, our proposed algorithm based on rectangular dictionary outperforms others not only in Wasserstein distance but also for the performance of reconstructed channel




(a) $SNR = [5, 10, 15, 20]$ (b) $SNR = [5, 10, 15, 20]$

**Fig. 9.4** ASF estimation quantitative comparison when sample number $T = 1 \times M$, and SNR $= \{5, 10, 15, 20\}$ dB: (**a**) Wasserstein distance for estimating ASFs, (**b**) normalized Frobenius norm error for reconstructed channel covariance matrix

covariance matrix. Besides, compared with NNLS, projection method is better at reconstructing channel covariance matrix.

**ASF Test Sample**

We take one sample from test set as an example to visually find out how the performance of ASF estimation is. As illustrated in Fig. 9.5, the first row is from DNN based on rectangular dictionary, where the red line is the true ASF needed to reconstruct, the green line is the approximate ASF adopting rectangular dictionary, and the blue line is the estimate ASF from DNN. The latter two rows are estimate ASF from NNLS and projection method, respectively. It is shown that MUSIC algorithm is able to detect the locations of spikes even if in the regime of low sample number and low SNR. Furthermore, since projection is targeted at reconstructing continuous function, it is hard to reconstruct spikes in the exact location but to spread energy in regime of spike locations with large amplitudes, where we only show its truncated version. In general, compared with NNLS and projection method, our proposed algorithm is able to distinguish between diffuse ASF and discrete ASF and thus can have a better estimate ASF.

### *9.6.3 Performance with Different Sample Numbers*

In order to evaluate the performance of reconstructing ASF based on different sample numbers $T$, we generate 5000 pairs of data when SNR $= 5$ dB and $\frac{T}{M} = \{0.125, 0.25, 0.5, 1, 2\}$, where the estimate ASFs of our proposed algorithm come from a pre-trained model trained on SNR $= 10$ dB, and $\frac{T}{M} = 1$. As illustrated in Fig. 9.6, it shows that our proposed algorithm outperforms other methods in low sample number regime. However, as sample number is increasing, the other approaches are getting better.

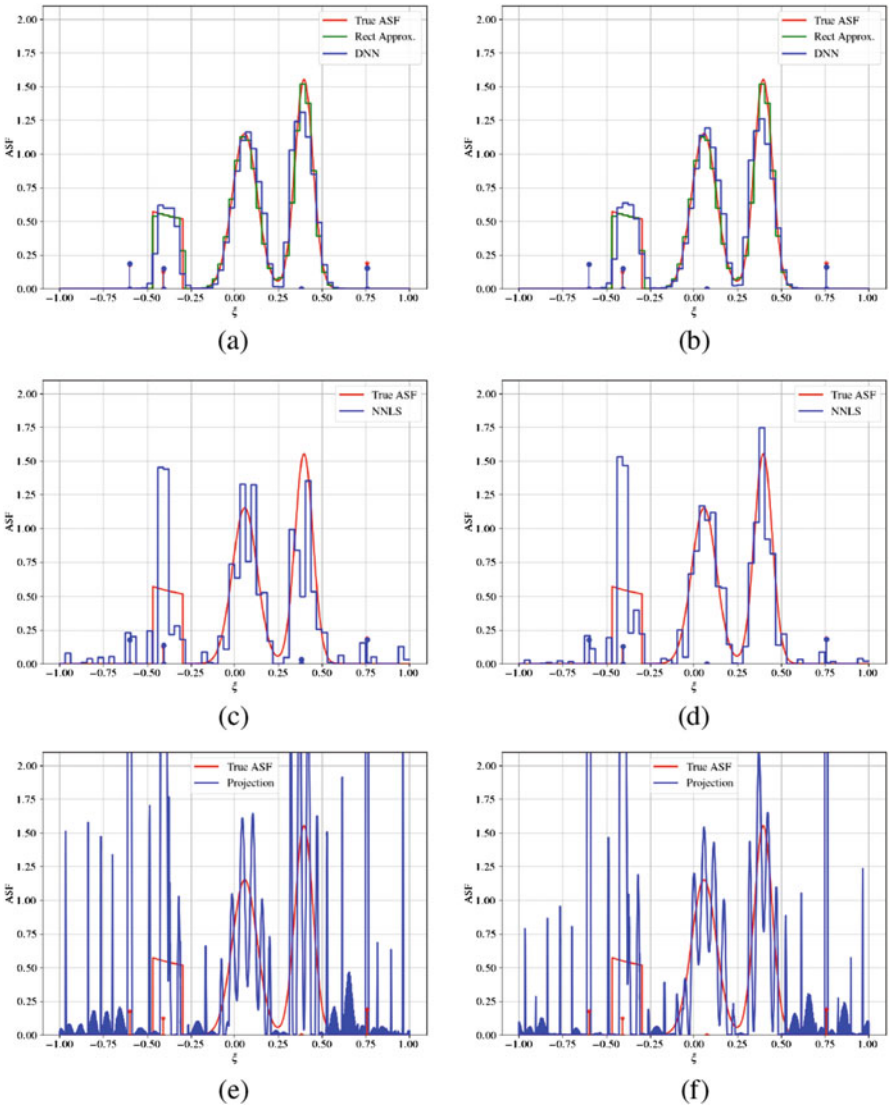



**Fig. 9.5** One sample from test set, consisting of two Gaussian diffuse scatterings and one rectangular scattering as well as three spikes, is estimated from different approaches, when sample number $T = 1 \times M$ and SNR= {5, 10} dB. (**a**) $SNR = 5dB$. (**b**) $SNR = 10dB$. (**c**) $SNR = 5dB$. (**d**) $SNR = 10dB$. (**e**) $SNR = 5dB$. (**f**) $SNR = 10dB$

### ASF Test Sample

In this section, one example from test set is illustrated in Fig. 9.7 when SNR = 5 dB, and sample number $T = \{1, 2\} \times M$. In the experiment, MUSIC will produce $\widehat{L} = 4$ locations of spikes, while there are only $L = 3$ spikes. As we can see, our

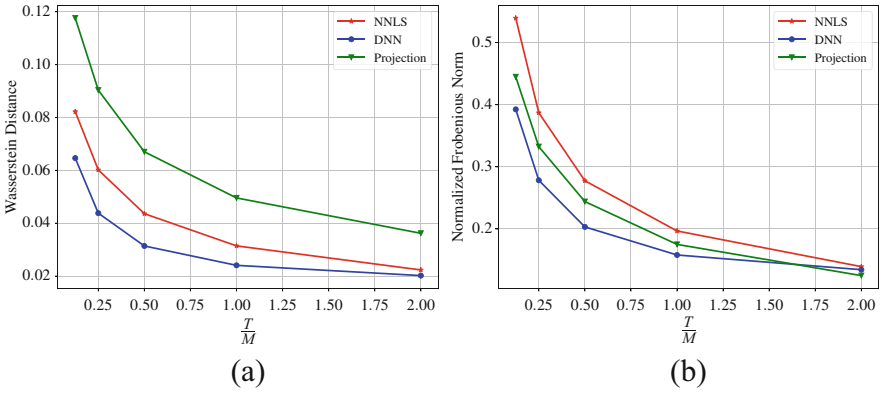



**Fig. 9.6** ASF estimation quantitative comparison when SNR=5 dB, and the number of noisy samples $T = [0.125, 0.25, 0.5, 1, 2] \times M$: (**a**) Wasserstein distance for ASF estimation, (**b**) normalized Frobenius norm error for reconstructed channel covariance matrix. (**a**) $\frac{T}{M} = [0.125, 0.25, 0.5, 1, 2]$. (**b**) $\frac{T}{M} = [0.125, 0.25, 0.5, 1, 2]$

proposed algorithm is able to set the energy of extra location as close as to zero, while in NNLS, a small amount of energy will be assigned to the extra locations, thus suppressing the energy of diffuse part when the location is within the diffuse scattering. However, as sample number is increasing, the ASF estimation for NNLS is more accurate. However, even if in the small sample number and noisy case, our proposed algorithm is still able to distinguish the diffuse part and discrete part separately. It is visually shown that our proposed algorithm outperforms others in the low sample regime.

## 9.7 Conclusion

In this chapter, a DNN-based algorithm is proposed for ASF estimation in MIMO systems. Unlike conventional approaches, we consider there are not only Line-of-Sight signals, or specular reflections, but also diffuse scatterings during propagation. Therefore, we introduce a two-step method for high-resolution ASF estimation. After the locations of spikes are estimated from MUSIC algorithm in the first step, DNN is applied for estimating the coefficients of diffuse ASF as well as the energy of spikes with estimate locations and noisy sample covariance matrix as input. Moreover, we make a solid comparison, and it shows that our proposed method outperforms other methods in low sample number regime in both qualitative and quantitative terms.

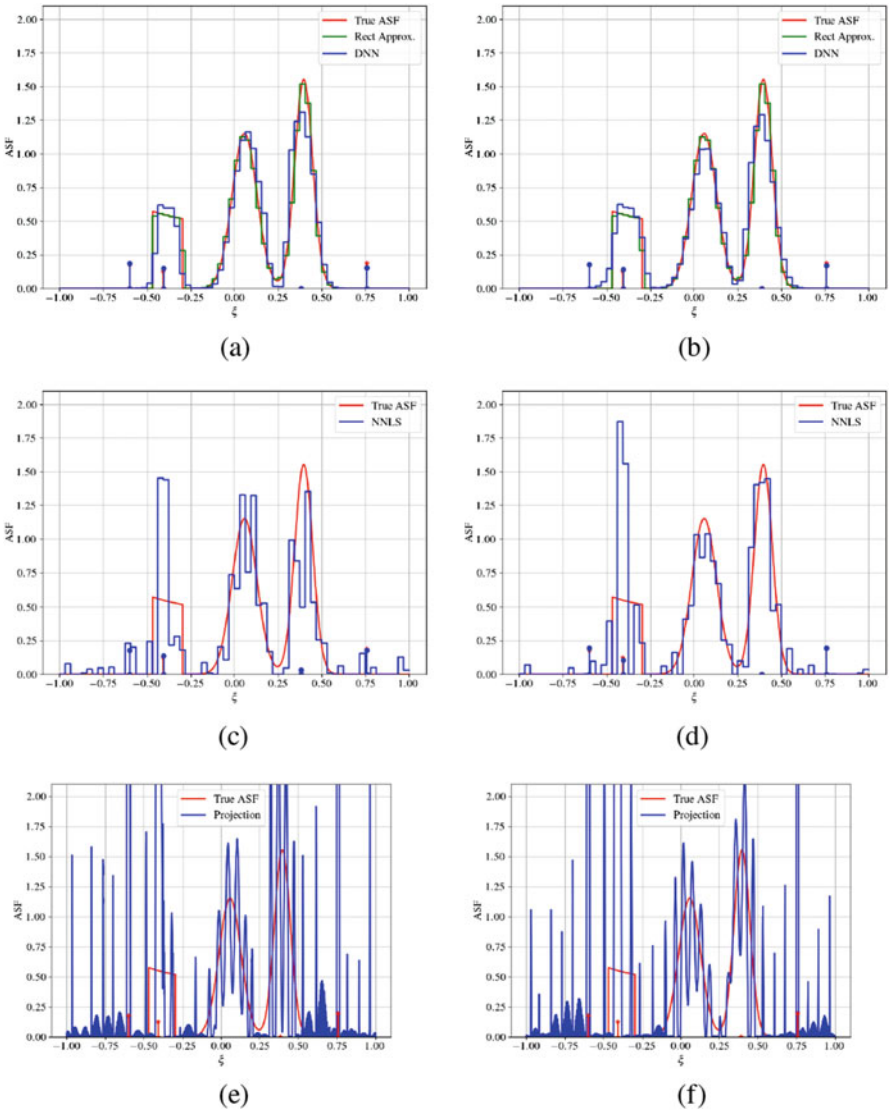



**Fig. 9.7** One sample from test set, which consists of two Gaussian diffuse scatterings, one rectangular scattering, and three spikes, is estimated from different approaches when SNR=5 $dB$, and the number of noisy samples $T = [1, 2] * M$. (**a**) $\frac{T}{M} = 1$. (**b**) $\frac{T}{M} = 2$. (**c**) $\frac{T}{M} = 1$. (**d**) $\frac{T}{M} = 2$. (**e**) $\frac{T}{M} = 1$. (**f**) $\frac{T}{M} = 2$

## References


1. Adhikary, A., Nam, J., Ahn, J.Y., Caire, G.: Joint spatial division and multiplexing: the large-scale array regime. IEEE Trans. Inf. Theory **59**(10), 6441–6463 (2013)
2. Adhikary, A., Al Safadi, E., Samimi, M.K., Wang, R., Caire, G., Rappaport, T.S., Molisch, A.F.: Joint spatial division and multiplexing for mm-wave channels. IEEE J. Sel. Areas Commun. (JSAC) **32**(6), 1239–1255 (2014)
3. Behboodi, A., Rauhut, H., Schnoor, E.: Compressive sensing and neural networks from a statistical learning perspective. arXiv preprint arXiv:2010.15658 (2021)
4. Bhaskar, B.N., Tang, G., Recht, B.: Atomic norm denoising with applications to line spectral estimation. IEEE Trans. Signal Process. **61**(23), 5987–5999 (2013)
5. Björnson, E., Larsson, E.G., Debbah, M.: Massive MIMO for maximal spectral efficiency: how many users and pilots should be allocated? IEEE Trans. Wirel. Commun. **15**(2), 1293–1308 (2016)
6. Bora, A., Jalal, A., Price, E., Dimakis, A.G.: Compressed sensing using generative models. In: Precup, D., Teh, Y.W. (eds.) Proceedings of the 34th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 70, pp. 537–546. PMLR, New York (2017). http://proceedings.mlr.press/v70/bora17a.html
7. Cavalcante, R.L.G., Stanczak, S.: Hybrid data and model driven algorithms for angular power spectrum estimation. arXiv preprint arXiv:2005.14003 (2020)
8. Choi, T., Rottenberg, F., Luo, P., Zhang, J., Molisch, A.F.: How many antennas do we need for massive MIMO channel sounding?-validating through measurement. arXiv preprint arXiv:1903.08207 (2019)
9. Donoho, D.L.: Compressed sensing. IEEE Trans. Inf. Theory **52**(4), 1289–1306 (2006)
10. Haghighatshoar, S., Khalilsarai, M.B., Caire, G.: Multi-band covariance interpolation with applications in massive MIMO. In: 2018 IEEE International Symposium on Information Theory (ISIT), pp. 386–390. IEEE, New York (2018)
11. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016). https://doi.org/10.1109/CVPR.2016.90
12. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: Proceedings of the 32nd International Conference on International Conference on Machine Learning, vol. 37, ICML'15, pp. 448–456. JMLR.org (2015)
13. Khalilsarai, M.B., Haghighatshoar, S., Yi, X., Caire, G.: FDD massive MIMO via UL/DL channel covariance extrapolation and active channel sparsification. IEEE Trans. Wirel. Commun. **18**(1), 121–135 (2018)
14. Khalilsarai, M.B., Yang, T., Haghighatshoar, S., Caire, G.: Structured channel covariance estimation from limited samples in massive MIMO. In: IEEE International Conference on Communications (ICC), pp. 1–7 (2020)
15. Larsson, E., Edfors, O., Tufvesson, F., Marzetta, T.: Massive MIMO for next generation wireless systems. IEEE Commun. Mag. **52**(2), 186–195 (2014)
16. Lucas, A., Iliadis, M., Molina, R., Katsaggelos, A.K.: Using deep neural networks for inverse problems in imaging: Beyond analytical methods. IEEE Signal Process. Mag. **35**(1), 20–36 (2018). https://doi.org/10.1109/MSP.2017.2760358
17. Marzetta, T.L.: Noncooperative cellular wireless with unlimited numbers of base station antennas. IEEE Trans. Wirel. Commun. **9**(11), 3590–3600 (2010)
18. Miretti, L., Cavalcante, R.L., Stanczak, S.: FDD massive MIMO channel spatial covariance conversion using projection methods. In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 3609–3613. IEEE, New York (2018)
19. Najim, O., Vallet, P., Ferré, G., Mestre, X.: On the statistical performance of music for distributed sources. In: 2016 IEEE Statistical Signal Processing Workshop (SSP), pp. 1–5. IEEE, New York (2016)

20. Schmidt, R.O.: Multiple emitter location and signal parameter estimation. IEEE Trans. Antennas Propag. **34**(3), 276–280 (1986)
21. Slawski, M., Hein, M., et al.: Non-negative least squares for high-dimensional linear models: consistency and sparse recovery without regularization. Electronic Journal of Statistics **7**, 3004–3056 (2013)
22. Stoica, P., Nehorai, A.: MUSIC, maximum likelihood, and Cramer-Rao bound. IEEE Trans. Acoust. Speech Signal Process. **37**(5), 720–741 (1989)
23. Stoica, P., Moses, R.L., et al.: Spectral Analysis of Signals, vol. 452. Pearson Prentice Hall, Upper Saddle River (2005)
24. Tamir, J.I., Rappaport, T.S., Eldar, Y.C., Aziz, A.: Analog compressed sensing for RF propagation channel sounding. In: 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5317–5320 (2012)
25. Tang, G., Bhaskar, B.N., Shah, P., Recht, B.: Compressed sensing off the grid. IEEE Trans. Inf. Theory **59**(11), 7465–7490 (2013)
26. Tropp, J.A.: Greed is good: Algorithmic results for sparse approximation. IEEE Trans. Inf. Theory **50**(10), 2231–2242 (2004)
27. Tse, D., Viswanath, P.: Fundamentals of Wireless Communication. Cambridge University, Cambridge (2005)
28. Wang, R., Bas, C.U., Cheng, Z., Choi, T., Feng, H., Li, Z., Ye, X., Tang, P., Sangodoyin, S., Gómez-Ponce, J., et al.: Enabling super-resolution parameter estimation for mm-wave channel sounding. IEEE Trans. Wirel. Commun. **19**(5), 3077–3090 (2020)

# Chapter 10
# Fast Radio Propagation Prediction with Deep Learning


**Ron Levie, Çağkan Yapar, Giuseppe Caire, and Gitta Kutyniok**

## 10.1 Introduction

In wireless communications, the pathloss is a quantity that measures the loss of signal strength (reduction in power or attenuation) between a transmitter (Tx) and a receiver (Rx) due to large-scale effects. The signal power attenuation may be caused by different factors, such as free-space propagation loss, reflections and diffraction from buildings, waveguide effects in street canyons, and obstacles blocking line of sight between Tx and Rx. The pathloss function (sometimes referred to as *path gain* function or *radio map*) is a function that assigns to each Tx–Rx pair of locations $x, y$ the corresponding large-scale signal attenuation $G(x, y)$.

In this chapter, we introduce *RadioUNet* [27, 28]—a deep learning method for estimating radio maps, based on UNets. UNets are a special type of convolution networks, ubiquitous in imaging and computer vision applications. Radio maps can be represented as images, in which the pixels represent spatial locations and the pixel values represent pathloss values. From this point of view, the radio map

The two authors "Ron Levie and Çağkan Yapar" contributed equally to this work.

R. Levie (✉)
Faculty of Mathematics, Technion – Israel Institute of Technology, Haifa, Israel
e-mail: levieron@technion.ac.il

Ç. Yapar · G. Caire
Department of Telecommunication Systems, Technische Universität Berlin, Berlin, Germany
e-mail: cagkan.yapar@tu-berlin.de; caire@tu-berlin.de

G. Kutyniok
Department of Mathematics, Ludwig-Maximilians-Universität München, München, Germany

Department of Physics and Technology, University of Tromsø, Tromsø, Norway
e-mail: kutyniok@math.lmu.de



G. Kutyniok et al. (eds.), *Compressed Sensing in Information Processing*, Applied and Numerical Harmonic Analysis, https://doi.org/10.1007/978-3-031-09745-4_10

estimation task can be seen as the problem of generating a radio map image from some information about the physical environment in which the wireless communication system operates. Hence, the idea of using UNets for estimating radio maps, which was first introduced in [27, 28], is quite natural, and the follow-up works repeated the idea under slightly different assumptions and with slightly different methodologies [37, 44, 64].

### *10.1.1 Applications of Radio Maps*

Many applications in wireless communication explicitly rely on the knowledge of the pathloss function, and thus, estimating pathloss is a crucial task. For example, in device-to-device (D2D) link scheduling, there exists a set of wireless devices that transmit signals to each other in pairs. A pair of devices that communicate defines a Tx–Rx link. The signal sent by a Tx is generally received by multiple Rx's beyond its intended destination, creating mutual interference between the links. While the general information-theoretic setting for this problem is the Gaussian interference channel, whose capacity region and optimal coding techniques are still an open problem in general, a huge amount of work have been devoted to the problem of scheduling subsets of links to be active on the same time slot and frequency sub-band, such that their mutual interference is sufficiently weak and the multiuser interference can be treated as Gaussian noise. It turns out that in a particular regime of weak interference, *Treating Interference as Noise* (TIN) is information-theoretic approximately optimal [14]. Furthermore, efficient link scheduling and power control combined with TIN yield very good performance in comparison with classical interference avoidance schemes such as CSMA [6]. A practical such link scheduling algorithm developed by Qualcomm is FlashLinQ [59]. Recent works on information-theoretic inspired D2D link scheduling include [34, 61], which significantly improve upon FlashLinQ. A recent more direct approach based on fractional programming optimization is provided in [52]. All these schemes somehow assume that the pathloss function between every Tx–Rx location is known or can be accurately estimated via some probing scheme. A deep learning approach to D2D link scheduling is proposed in [12], which is implicitly based on the fact that interference is a decreasing function of distance and therefore that the pathloss function has a radial symmetry. Therefore, such scheme does not directly apply to more complicated urban propagation scenarios as considered in the present chapter. From the above works, it is clear that an accurate knowledge of the radio map for a specific environment is very important for efficient D2D links scheduling.

Another classical use-case example of radio maps is base station assignment, or user-cell site association, where the goal is to assign a set of wireless devices to a set of cellular base stations. In order to decide which device to assign to which station, it is important to know the radio map (e.g., see [5] and references therein).

Some additional applications that rely on the knowledge of the pathloss function are fingerprint-based localization [35], physical-layer security [57], power control

in multi-cell massive MIMO systems [9], user pairing in MIMO–NOMA systems [40], precoding in multi-cell large-scale antenna systems [2], path planning [63], and activity detection [8].

### *10.1.2 Radio Map Prediction*

A multitude of approaches for estimating the pathloss function have been proposed in the literature. For the sake of clarity, we can group these approaches into three categories.

*Data-driven interpolation methods* assume that some measurements of the pathloss function are given at certain locations. These methods estimate the pathloss function at non-measured locations via some signal processing approach (e.g., *Kriging* [55]) and do not rely—or rely only lightly—on a model of the physical phenomenon. Beyond Kriging, other examples of such approaches are radial basis function interpolation [7, Sect 5.1], tensor completion [50], support vector regression [56], and matrix completion [10].

*Model-based data-fitting methods* combine measurements of the pathloss function with a priori assumptions on the physical system to estimate the pathloss function at non-measured locations. For example, in tomography methods, the attenuation due to shadowing can be derived under some modeling assumptions from the so-called spatial loss field (SLF), which in turn can be estimated from the measurements. Here, various assumptions on the underlying SLF can be imposed, e.g., low-rank structure [25], sparsity [47], and piecewise homogeneity [24, 26].

Last, *model-based prediction* estimates the pathloss function based only on the available prior knowledge, e.g., physical considerations, without taking any measurements from the area of interest. We focus in this chapter on simulations based on tracking the signal along rays. One well-known class of methods is *ray tracing* [46], where the signal is modeled as rays that are cast from the transmitter, travel in straight lines in homogeneous medium (such as free space), and undergo rules of reflection, refraction, and diffraction when the medium changes (e.g., when hitting an obstacle). One efficient ray-tracing method is called intelligent ray tracing (IRT). This algorithm starts with a pre-processing step that takes the considered city map, and structures the data in a manner that allows later on to compute the interactions of the rays with the geometry efficiently. Dominant path model [58] is a simplification of ray tracing, where only diffractions are considered, which allows a more efficient implementation. Another class of methods are empirical models, e.g., [65].

### 10.1.3 Radio Map Prediction Using Deep Learning

Two recent papers proposed deep learning approaches for estimating radio maps [19, 49]. There, the neural network is a function that returns an estimate of the pathloss for each input Tx–Rx location. The network is trained on a fixed map and simulated pathloss values at a set of Tx–Rx locations. This procedure is a data-fitting method for the 4-dimensional (4D) function $G(x, y)$.[1] Different city maps require re-training the network, and each trained network describes a specific map. In contrast, our RadioUNet learns to estimate the underlying physical phenomenon and executes a type of implicit simulation, given by the operations of its underlying convolution network, which interacts with any Tx source and city map. Even when the map is fixed, we show that RadioUNet significantly outperforms previous deep learning proposed methods.

There are several more papers on pathloss prediction that use fully connected neural networks, which do not take the city map information into consideration, and use additional information such as the height of the transmitter/receiver or the distance between them. For example, see the survey [43], and the papers [42, 53, 54].

Another recent work based on data fitting to radio maps via deep learning, in the above fashion, is [39]. The authors of [39] also proposed a transfer learning approach to learn a radio map estimator corresponding to some antenna tilt $T_B$ from a radio map estimator of another tilt $T_A$. There, it is assumed that there is a large amount of data to train the tilt $T_A$ and a small amount of data for the tilt $T_B$. We also consider a transfer learning approach, in which we train a radio map estimator on a large dataset of simulations, and transfer it to real life with the aid of a small dataset of real-life measurements.

## 10.2 Introduction to Radio Map Prediction with RadioUNet

In this chapter, we propose several versions of a radio map estimation method based on deep learning, which we term *RadioUNet*.[2] In our setting, we consider mobile devices/base stations in an urban environment. Our deep-learning-based methods are efficient, estimating the whole radio map within an area of $256^2\text{m}^2$ in an order of $10^{-3}$sec to $10^{-2}$sec, with root mean square accuracy of order 1 dB, where the range of pathloss values from the noise floor to the maximal gain is 100 dB. This is a mean accuracy of 1% (RMSE divided by the range).

[1] Notice that when $x$ and $y$ are points on the plane $\mathbb{R}^2$, the function $G(\cdot)$ has domain in $\mathbb{R}^4$.

[2] The source code of RadioUNet can be found at https://github.com/RonLevie/RadioUNet.

### 10.2.1 RadioUNet Methods

Our radio map estimation methods are based on UNets [48] and their compositions. One version of RadioUNet (called RadioUNet$_C$) only uses as input the geometry of the urban environment, which may be perturbed, the Tx location, and no pathloss measurements. Thus, this method can be categorized as model-based simulation. However, as opposed to classical-model-based simulation, our model is learned from training data and does not have an explicit physically interpretable formulation. Another model that we propose (called RadioUNet$_S$, with S for samples) takes as an additional input variable some measurements of the pathloss at a few locations. Thus, this method can be categorized as a model-based data-fitting method. Another optional input variable are the locations of cars along the streets, which help predict the shadowing effect due to the penetration of the signal through cars.

### 10.2.2 The Training Data

We present a new dataset, called *RadioMapSeer*,[3] of 56,000 simulated radio maps in different city locations and different Tx locations. Each simulation has a number of versions, generated using different types of coarse simulations. The coarse simulations we use are dominant path model (DPM) [58] and intelligent ray tracing [45] based on 2 interactions of the rays with the geometry, called IRT2. The coarse simulations are saved as dense measurements of the radio map in a 2D dense grid of $256 \times 256\,\mathrm{m}^2$. We also consider two more coarse simulation types, based on DPM and IRT2, in which cars are generated along the streets, and affect the simulation. The cars serve as unpredictable obstacles perturbing the received signal strength. Alongside each simulation, the map of the city, the Tx locations, and cars are also provided.

In addition, we present a smaller dataset of 1400 high-accuracy simulations, called IRT4 (IRT with 4 interactions). In our setting, IRT4 serves as a surrogate for real-life measured radio maps, i.e., the effective ground truth with respect to which we calculate the prediction error. A second version of this dataset has IRT4 simulations including the effect of cars. To imitate a realistic scenario where the 1400 IRT4 simulations represent real-life measurements collected during a measurement campaign, or even in real time from user devices, we suppose that each of the 1400 radio maps is only measured sparsely, e.g., we only have 300 receiver locations per map. We note that we are not trying to study the accuracy of IRT4, and we do not even have to assume that IRT4 is a high-accuracy method. The idea is that DPM, IRT2, and IRT4 all share a basic coarse behavior, namely, they roughly represent the basic underlying propagation phenomenon, but IRT4 has additional

[3] The dataset can be found at https://RadioMapSeer.github.io.

finer details not present in DPM or IRT2. IRT4 shares this property with real-life radio maps. One goal is then to develop methods to predict the fine details of IRT4, even though in training we have access to a large dataset of DPM and IRT2, but only to a small set of sparse measurements of IRT4. Of course, when RadioUNet is employed in practice, the refined phenomenon should be taken as the actual real-life measurements.

### *10.2.3 Generalizing What Was Learned to Real-Life Scenarios*

As discussed above, one important aspect that we address in this chapter is how to generalize the RadioUNet, trained on coarse simulations, to real life. To assess the performance of a trained RadioUNet in real life, we test it on the small dataset of high-accuracy IRT4 simulations, serving as a surrogate to real-life measurements. The ultimate goal is to transfer what RadioUNet learned to real-life deployment. Our methods learn the "big-picture" coarse phenomenon from the large DPM and IRT2 dataset and use the additional IRT4 sparse samples to refine and adapt the RadioUNet from simulation to "real-life," using a relatively small number of trainable parameters. We thus demonstrate that a RadioUNet, trained on coarse simulations, can learn to estimate the fine details of a more complex phenomenon. When RadioUNet is employed in practice, the refined phenomenon should be taken as real-life measurements.

A second approach for transferability is training a RadioUNet that estimates radio maps from three *input feature channels*, the city map data, the Tx location, and some pathloss measurements. The method is trained to estimate coarse simulations by combining the data from the city map and the measurements. However, once trained, the RadioUNet can be employed in real life, where real-life input measurements of the pathloss are now taken.

### *10.2.4 Applications*

Our RadioUNet can be directly applied to any of the problems mentioned before, where an accurate knowledge of the pathloss function between any Tx–Rx pair of locations is useful. In a dynamic environment, the set of refined measurements can be provided in real time from the mobile devices, along with their position. For the sake of space limitation, in this chapter, we demonstrate the potential of our radio map estimation method with two applications.

**Coverage Classification** We show how to predict the service area of a Tx, and conversely, show how to estimate the domain where the Tx creates small interference with other devices. This example is taken from [28].

**Pathloss Fingerprint-Based Localization** Using the estimated radio maps of a set of devices/base stations with known location, the location of some other device $d$ can be accurately computed if $d$ reports the received signal gains from the base stations. An extension of the approach presented here can be found in [60].

## 10.3 Background and Preliminaries

### *10.3.1 Wireless Communication*

The pathloss function assigns to each Tx–Rx pair of locations $x, y$ the corresponding large-scale signal attenuation $G(x, y)$. Notice that in addition to the large-scale effects, wireless propagation is also subject to small-scale fading, due to the superposition of scattered wavefronts with different phases at the Rx location. Such small-scale effects are typically modeled as a Gaussian random variable $H$ that, without loss of generality, can be normalized with unit second moment. Therefore, if we denote by $Y = \sqrt{G(x, y)}HX + Z$ a signal sample at the Rx baseband output, where $X$ is the transmitted signal sample with power $\mathrm{P_{Tx}}$, $H$ is the normalized small-scale fading, and $Z$ is the additive noise with power spectral density $N_0$, the received energy per sample is generally given by $\mathbb{E}[|Y|^2] = G(x, y)\mathrm{P_{Tx}}/W + N_0$. Here, $W$ is the signal bandwidth, and the signal-to-noise ratio (SNR) at the input of the Rx baseband processor is given by $\mathsf{SNR} = \frac{G(x,y)\mathrm{P_{Tx}}}{N_0 W}$.

Consider a general Gaussian interference network with $K$ Tx and $N$ Rx devices located over a certain region of the 2D plane. Following the *generalized degrees of freedom* (GDoF)-oriented model in [14], it is useful to normalize the received signal such that the variance of the noise samples $N_0$ and the signal energy per symbol $\mathrm{P_{Tx}}/W$ are both equal to 1, and define a parameter $P$ such that the normalized received signal at each $j$-th Rx is given by

$$Y_j = \sum_{i=1}^{K} \sqrt{P^{\alpha_{i,j}}} X_i + Z_j, \tag{10.1}$$

where $\alpha_{i,j} = \frac{\log \mathsf{SNR}_{i,j}}{\log P}$ and $\mathsf{SNR}_{i,j}$ is the SNR between Tx $i$ and Rx $j$ as defined in Sect. 10.1. It turns out that the *GDoF region* of the underlying Gaussian interference network (i.e., a high-SNR representation of the capacity region) is defined by the exponents $\alpha_{i,j}$. Furthermore, under certain conditions (see [14, 61]), the GDoF region yields the actual *capacity region* within a one bit gap. These facts provide a strong evidence that the relevant scale to estimate the pathloss function is logarithmic, i.e., on a dB scale. Furthermore, from the theory in [14], it follows that negative values of these exponents are irrelevant, that is, for the GDoF region, it is sufficient to take the positive part of the $\alpha_{i,j}$'s. In practice, this means that we do not have to spend much effort in estimating very large negative values (in dB) of the

pathloss function. As a matter of fact, it makes sense to truncate such function such that the received signal power is not too much smaller than the noise floor.

Driven by the above considerations, we define the pathloss in dB scale as $P_L = (P_{Rx})_{dB} - (P_{Tx})_{dB}$, where $P_{Tx}$ and $P_{Rx}$ denote the transmitted power and received power at the Tx and Rx locations, respectively. The truncation and rescaling of the pathloss function in dB scale in order to make it suitable for the proposed deep learning estimation method are given in Sects. 10.4.2 and 10.4.3.

## *10.3.2 Deep Learning*

In this subsection, we go over the required material from deep learning.

### 10.3.2.1 An Interpretation of Deep Learning

In this chapter, we use a deep learning approach for simulating radio maps. To explain what deep learning is, let us present one point of view that we find constructive, namely, seeing deep learning as an approach for algorithm design.

Traditional algorithms are designed "manually," where each step is specified to the last detail to achieve the end goal of the algorithm. In contrast, deep learning can be seen as the practice of designing algorithms by laying down the general outline of the different steps and specifying the types of computational tools to be used in the algorithm. The choice of the general blueprint of the algorithm is called the *architecture*. In deep learning, the fine details of the algorithm are automatically tuned by optimization to achieve the end goal of the algorithm and not explicitly designed by humans. This optimization is called *training* in the machine learning jargon.

In this interpretation, a *layer* in a deep learning architecture means a step in the algorithm. The term *deep* in deep learning means that there are many layers, or equivalently many steps. The number of layers is called the *depth* of the architecture. The fine details to be optimized in each layer are given as free parameters of the architecture and are called the *learnable parameters*, or *weights* in some architectures. A deep learning algorithm receives inputs and produces outputs. The algorithm is written down, or *unfolded* in the deep learning jargon, as the end-to-end function that transforms the input to the output [17, 38]. This function is sometimes called the *network*. Unfolding is done by composing the different steps, or layers, one on top of the other.

The vast majority of deep learning methods are trained using some variant of gradient descent on the learnable parameters (e.g., see [21]). One step in gradient descent is called an *Euler step*. Since in gradient descent the gradient of the network is computed at each Euler step, and the network is the composition of all of the layers, the chain rule plays an imported role. Using the chain rule in gradient descent is called *back-propagation* in the deep learning jargon.

A deep learning architecture keeps enough parameters free to be able to express a large class of algorithms. This is called the *expressive capacity* of the network. The more expressive the network, the more versions of the algorithm there are to explore during training, and thus the harder optimization is. On the other hand, if the network is not expressive enough, there might not be any choice of the parameters that constitutes an adequate algorithm. A good deep learning architecture is designed by choosing general steps that are suitable to the specific problem. Choosing steps that are natural for solving the problem means that the network does not have to learn these steps as combinations of more basic steps. This helps in reducing the amount of learnable parameters. The idea that a network is predisposed to certain algorithmic approaches, or has some built-in functionalities, is sometimes called the *innateness* of the network [31].

#### 10.3.2.2 Convolutional Neural Networks

A *convolutional neural network* (CNN) is a popular deep learning architecture, typically used in machine learning applications in imaging science [22, 23]. In our context, a *feature map* is a function from a 2D grid to some $\mathbb{R}^N$, where $N$ is called the number of feature channels of the feature map. If $N = 1$, we call the feature map a *gray-level image*. A CNN is defined by aggregating the following five basic computational steps as the layers of the network.

A *convolution layer* is a step where an input feature map is convolved with a filter kernels and added to some scalars called the *bias*. The numbers of input feature channels and output feature channels need not coincide. More accurately, let $N$ be the number of input feature channels and $M$ the number of output feature channels. Let $f_1, \ldots, f_N$ be the feature channels of the input feature map. Note that each $f_n$ is a gray-level image, not a scalar. The feature channels of the output feature map, $g_m$, are defined for every $m = 1 \ldots, M$ by

$$g_m = \sum_{n=1}^{N} f_n * y_{n,m} + b_m, \tag{10.2}$$

where $*$ denotes convolution, and for each $m = 1, \ldots, M$ and $n = 1, \ldots, N$, $y_{n,m}$ is a gray-level filter kernel, and $b_m$ is the $m$-th component of the bias. We emphasize here that for each $(n, m)$, $y_{n,m}$ is a filter kernel, not a scalar.

An *activation function* is any function applied on the entries of a feature map, and a typical choice is ReLU, defined by $r(z) = \max\{0, z\}$. A *pooling layer* takes a feature map and downsamples it, e.g., by assigning the maximal entry of each $2 \times 2$ patch to the corresponding entry of the down-sampled feature map. An *up-sampling layer* upsamples lower-resolution feature maps to higher-resolution ones. Last, a *fully connected layer* is a general linear operator/matrix applied on the feature map and added to some pre-defined bias. A CNN architecture is defined by choosing how to combine the above layers, choosing the number of feature channels, and

choosing the shapes of the filter kernels. The trainable parameters are the filters, the fully connected matrices, and the biases.

#### 10.3.2.3 UNets

UNet is a special CNN architecture, introduced in [48], and used in a multitude of applications, including image segmentation [3, 11, 33, 51], video predicting [32], super-resolution/image inpainting [29], inverse problems in imaging [20], image-to-image translation [62], and medical image analysis [30] to name a few.

UNets consist of convolution, pooling, up-sampling, and activation function layers, without fully connected layers. The UNet architecture is divided into two paths. The first portion of the layers gradually contracts the image as the layers deepen and gradually increases the number of feature channels. This path—also called the *encoder*—is interpreted as a procedure for extracting "concepts" that become more complex/high level and less spatially localized along the layers. The second portion of the layers—also called the *decoder*—expands the image as the layers deepen and reduces the number of feature channels gradually. This path is interpreted as a procedure of combining/synthesizing the concepts, layer by layer, to lower-level concepts, and eventually to an output image. The decoder layers are derived by up-sampling lower-resolution images and thus lack high-resolution information on their own. To provide high-resolution information to the decoder layers, the feature map in the feature channels of the encoder layers is copied and concatenated to the corresponding feature channels of the decoder layers having the same resolution. This copying between non-neighboring layers is called *skip connection*.

We write down UNets explicitly as follows. Consider a UNet $U$ based on $L$ layers. Let $\mathbf{p}_l$ denote the vector of all learnable parameters of layer $l$ of the UNet. Namely, $\mathbf{p}_l$ is a list that concatenates all of the entries of the different filters and the different biases of layer $l$. For any $l = 1, \ldots, L$, denote by $U^l$ the function that maps the feature map of layer $l$ to the feature map of layer $l+1$. Namely, $U^l$ applies a convolution plus bias step of the form (10.2), followed by an activation function and optionally pooling or up-sampling. To emphasize the reliance of $U^l$ on $\mathbf{p}_l$, we denote $U^l_{\mathbf{p}_l}$, and $U^l_{\mathbf{p}_l}$ applied on the feature map $\mathbf{f}$ of layer $l$ is denoted by $U^l_{\mathbf{p}_l}(\mathbf{f})$.

Let $\mathbf{p} = (\mathbf{p}_1, \ldots, \mathbf{p}_L)$ denote the concatenation of all learnable parameters of the UNet. The end-to-end unfolded UNet can be written as the composition

$$U_{\mathbf{p}} = U^1_{\mathbf{p}_1} \circ U^2_{\mathbf{p}_2} \circ \ldots \circ U^L_{\mathbf{p}_L}.$$

The output of the UNet on the input feature map $\mathbf{f}$ is given by

$$U_{\mathbf{p}}(\mathbf{f}) = U^L_{\mathbf{p}_L}\left(U^{L-1}_{\mathbf{p}_{L-1}}\left(\ldots U^1_{\mathbf{p}_1}(\mathbf{f})\ldots\right)\right). \tag{10.3}$$

#### 10.3.2.4 Supervised Learning of UNets via Stochastic Gradient Descent

In supervised learning, a *training set* of many example input images $\mathbf{f}_k$ and the corresponding desired output images $\mathbf{g}_k$ are given, where $k = 1, \ldots, K$ and $K$ is the size of the dataset. The goal is to finetune the parameters $\mathbf{p}$ of $U_{\mathbf{p}}$ so that $U_{\mathbf{p}}(\mathbf{f}_k) \approx \mathbf{g}_k$ for every $k = 1, \ldots, K$. The hope is that if the dataset is a good enough representation of the distribution of all possible input–output pairs, the UNet will successfully predict the correct output of examples outside of the training set. Namely, for unseen inputs $\mathbf{f}$ and desired output $\mathbf{g}$, we will have $U_{\mathbf{p}}(\mathbf{f}) \approx \mathbf{g}$. The success of the network on new examples is called *generalization*. If the network performs well on the training set but not on new examples, we say that the network *overfits* the training set.

In practice, a finite dataset of examples $\{(\mathbf{f}_k, \mathbf{g}_k)\}_{k=1}^{K}$ is given and is split artificially into three subsets. The first subset serves as the *training* set. The *validation set* is a subset on which training is not performed and is used for assessing the generalization capability of the network during the research and development process. The *test* set is saved aside, and generalization is tested on it only for the finalized architecture. The reason the test set is separate from the validation set is that during development the researcher, or some automated system, makes design choices to maximize the performance of the network on the validation set. It is thus possible that the final version of the network overfits both the training and validation sets.

The loss function to be optimized is typically of the form

$$\mathcal{L}(\mathbf{p}) = \frac{1}{K} \sum_{k=1}^{K} \left\| \mathbf{g}_k - U_{\mathbf{p}}(\mathbf{f}_k) \right\|^p \tag{10.4}$$

for some norm, e.g., the root mean square norm, and power $p > 0$, which is typically 2. In *stochastic gradient descent* (SGD), a stochastic version of $\mathcal{L}$ is given. Namely, for some $J \ll K$, let $\{k_1, \ldots, k_J\}$ denote a random selection of $J$ indices in $\{1, \ldots, K\}$ and define the random variable

$$\mathcal{L}_J(\mathbf{p}) = \frac{1}{J} \sum_{j=1}^{J} \left\| \mathbf{g}_{k_j} - U_{\mathbf{p}}(\mathbf{f}_{k_j}) \right\|^p . \tag{10.5}$$

Here, $J$ is called the *batch size*. In optimization, $\mathcal{L}_J(\mathbf{p})$ is realized at each Euler step independently, and the gradient with respect to $\mathbf{p}$ is computed for this specific realization of $\mathcal{L}_J(\mathbf{p})$. The random selection of indices is constructed in such a way that after $\lceil K/J \rceil$ iterations the batches go through the whole training set.

The gradient is computed using the chain rule on the sum of norms and the composition representation of the UNet (10.3), which gives a so-called back-propagation formula for the Euler step. When implementing UNets in modern deep learning libraries, such as PyTorch [41] or TensorFlow [1], there is no need to derive

a closed-form formula for the gradient of the loss function. Instead, if the loss function is written using built-in operations, the gradient is computed automatically.

#### 10.3.2.5 Curriculum Learning

The SGD optimization procedure (and its variants) explores configurations of the parameters only along the 1D path of descent, which might miss good configurations. Namely, SGD searches the parameter space in a highly non-exhaustive manner. This observation supports the principle that high expressive capacity does not guarantee converges of the deep network to a good solution. Thus, the expressive capacity of a network does not guarantee high-quality trained networks. It is thus often important to lead gradient descent in a more deliberate way and in some sense to "micro manage" the exploration of parameter configurations in the optimization process. One approach for achieving this is called *curriculum learning* [4]. In curriculum learning, training is divided into a *curriculum*, namely, a list of optimization problems, where the optimal solution of the previous problem is used as the initial guess for the next optimization problem. The idea is to first teach the network how to solve an easy-to-learn simplified version of the problem and gradually to increase the complexity of the problem until reaching the original formulation of the loss function.

#### 10.3.2.6 Out-of-Domain Generalization

In some learning scenarios, the training data does not represent real-life data completely faithfully. It is thus important to know whether the network, trained on one data distribution, performs well for another data distribution. The idea of training in one domain and testing in another domain is called *out-of- domain generalization*. The capacity of a network to perform well in new domains is called its *transferability*.

## 10.4 The RadioMapSeer Dataset

In this section, we introduce RadioMapSeer, a dataset of city maps with corresponding simulated radio maps that we have created and made available for this work.

### *10.4.1 General Setting*

The RadioMapSeer dataset consists of 700 maps, 80 transmitter locations per map, and corresponding coarsely simulated radio maps (using DPM and IRT2).

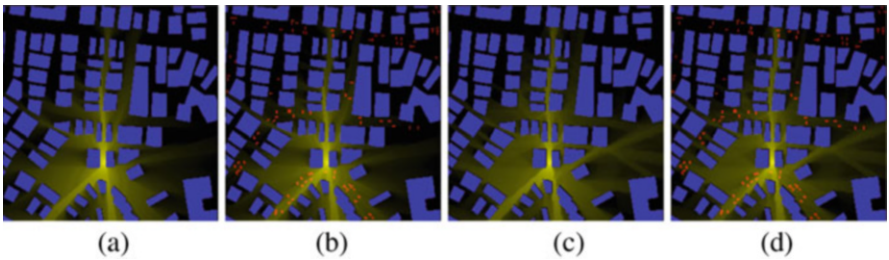



**Fig. 10.1** RadioMapSeer examples. Buildings are blue, cars red, and pathloss yellow. (**a**) DPM. (**b**) DPM with cars. (**c**) IRT4. (**d**) IRT4 with cars

The fine simulations (IRT4) are given for the first two transmitters of each map. Maps are taken from *OpenStreetMap* [36] in the cities Ankara, Berlin, Glasgow, Ljubljana, London, and Tel Aviv. We set the heights of the transmitters, receivers, and buildings as 1.5 m, 1.5 m, and 25 m, respectively, which is relevant to device-to-device scenarios (see Sect. 10.4.2 for more details). All simulations were computed using the software *WinProp* [18]. Some example radio maps from the dataset are shown in Fig. 10.1. All simulations are saved as dense measurements of the radio map in a 2D dense grid of $256 \times 256\,\mathrm{m}^2$.

#### 10.4.1.1 Maps and Transmitters

Each map covers $256 \times 256\,\mathrm{m}^2$ where buildings and roads are saved in the dataset as polygons. Each map is also converted into a morphological 2D image (namely, binary black and white) of $256 \times 256$ pixels, where each pixel represents one square meter. The interior of the buildings is white (pixel value $= 1$), and the exterior of the buildings is black (pixel value $= 0$). The transmitter locations are stored as numerical 2D values and also given as morphological images, where the pixel in which the transmitter is located is white and the rest is black.

Along with the city maps, roads are given both as polygonal lines and as morphological images with 1 on the road and 0 outside. Cars are generated along and aside roads, to represent driving and parking car, and given as separate morphological images.

#### 10.4.1.2 Coarsely Simulated Radio Maps

The simulated radio maps were generated using two types of simulations, namely, dominant path model (DPM) and intelligent ray tracing (IRT2), with the radio network planning software *WinProp* [18]. IRT2 is performed with two interactions of the rays. Each simulated radio map stores at each pixel the pathloss between the pixel location and the transmitter location in dB.

To represent uncertainty in the dataset, we consider two cases. First, a set of simulations on all city maps including the cars is produced using DPM and IRT2. These simulations are perturbations of the simulations based on the city map alone. We moreover provide separate datasets of perturbed city maps, where in each map of the original dataset $m$ buildings are missing. We provide four such datasets with $m = 1, \ldots, 4$.

#### 10.4.1.3 Higher-Accuracy Simulations

An additional smaller dataset of higher-accuracy simulations is provided. Here, for each of the 700 maps, we consider two transmitter locations and simulate the radio map using IRT with 4 interactions of the rays (IRT4).

The goal of the higher-accuracy simulations is to provide means of testing whether the network, trained on simulations, performs well in real life. Hence, the high-accuracy simulation serves as a surrogate for the real-life physical phenomenon.

#### 10.4.1.4 Pathloss Scale

The pathloss values $P_{\mathrm{L}}$ are converted into *gray-level* pixel values between 0 and 1 (see Sect. 10.4.3). Hence, each radio map is represented as a gray-level image of size $256 \times 256$.

### *10.4.2 System Parameters*

In this chapter, we stick to the current regulations for safety-related communications in intelligent transportation systems (ITS), which is based on the IEEE 802.11p standard. Accordingly, we consider a signal bandwidth $W$ of 10 MHz in the 5.9 GHz band. We choose the transmitter power and thermal noise power spectral density as $(\mathrm{P_{Tx}})_{\mathrm{dB}} = 23\,\mathrm{dBm}$ and $N_0 = -174\,\mathrm{dBm/Hz}$ in compliance with IEEE 802.11p and assume an idealistic noise figure of 0 dB at receivers (cf. Table 10.1 for a summary of the system parameters).

We express by $(\mathcal{N})_{\mathrm{dB}} = 10\log_{10} W + N_0 + \mathrm{NF}$ the noise floor in dB, with NF being the noise figure. We consider the points where the received signal power $(\mathrm{P_{Rx}})_{\mathrm{dB}} = \mathrm{P_L} + (\mathrm{P_{Tx}})_{\mathrm{dB}}$ yields a signal-to-noise ratio above a desired SNR level, i.e., the points where $(\mathrm{SNR})_{\mathrm{dB}} = (\mathrm{P_{Rx}})_{\mathrm{dB}} - (\mathcal{N})_{\mathrm{dB}} \geq \mathrm{SNR_{thr}}$ holds. Solving this for $\mathrm{P_L}$, we get the threshold $\mathrm{P_{L,thr}}$ for the pathloss

$$\mathrm{P_L} \geq \mathrm{P_{L,thr}} = -(\mathrm{P_{Tx}})_{\mathrm{dB}} + \mathrm{SNR_{thr}} + (\mathcal{N})_{\mathrm{dB}}. \tag{10.6}$$

**Table 10.1** RadioMapSeer Dataset parameters

| Parameter | Value |
|---|---|
| Number of transmitters | 80 |
| Frequency | 5.9 GHz |
| Bandwidth | 10 MHz |
| Pixel length | 1 meter |
| Noise power spectral density | −174 dBm/Hz |
| Transmit power | 23 dBm |
| Noise figure | 0 dB |

We call $\mathrm{P}_{\mathrm{L,thr}}$ the *pathloss threshold*. Consider for example the SNR requirement that the received signal power should be above the noise floor, i.e., when $\mathrm{SNR}_{\mathrm{thr}} = 0$. With the choice of parameters in Table 10.1, we find $\mathrm{P}_{\mathrm{L,thr}} = -127\,\mathrm{dB}$.

One task of RadioUNet is to extract the area in the city map above the noise floor, given an input city map and transmitter location. To do this, the network must learn the physical phenomenon both above and below the noise floor. We thus truncate the pathloss values below another threshold $\mathrm{P}_{\mathrm{L,trnc}} < \mathrm{P}_{\mathrm{L,thr}}$. We choose $\mathrm{P}_{\mathrm{L,trnc}}$ such that the difference between the maximum pathloss $M_1$ in the dataset and $\mathrm{P}_{\mathrm{L,thr}}$ is approximately four times greater than the difference between $\mathrm{P}_{\mathrm{L,thr}}$ and $\mathrm{P}_{\mathrm{L,trnc}}$, i.e., $M_1 - \mathrm{P}_{\mathrm{L,thr}} = 4(\mathrm{P}_{\mathrm{L,thr}} - \mathrm{P}_{\mathrm{L,trnc}})$. The maximum and minimum pathlosses in the dataset are −47.84 dB and −186.41 dB, respectively. Note that the maximum is −47.84 dB and not 0 dB since the pathloss is integrated over $1\mathrm{m}^2$ pixels. To meet the previously mentioned condition, we set $\mathrm{P}_{\mathrm{L,trnc}} = -147\,\mathrm{dB}$. Since any signal below $\mathrm{P}_{\mathrm{L,thr}}$ cannot be detected in practice and is only used in simulation for theoretical reasons, we call $\mathrm{P}_{\mathrm{L,trnc}}$ the *analytic noise floor*. Note that by (10.6) we have $\mathrm{P}_{\mathrm{L,thr}} = -\mathrm{P}_{\mathrm{Tx}} + \mathrm{SNR}_{\mathrm{thr}} + N_0 + \mathrm{NF} + 10\log_{10} W$. Hence, any choice of the parameters on the RHS that results in the same pathloss threshold $\mathrm{P}_{\mathrm{L,thr}}$ has the same radio map.

### 10.4.3 Gray-Level Conversion

We convert the pathloss values $\mathrm{P}_{\mathrm{L}}$ into pixel values between 0 and 1 as follows. Denote by $M_1$ the maximal pathloss in all radio maps in the dataset, and define $f = \max\{\frac{\mathrm{P_L} - \mathrm{P_{L,trnc}}}{M_1 - \mathrm{P_{L,trnc}}}, 0\}$. Here, $f = 0$ represents anything below the analytic noise floor, and $f = 1$ represents the maximal gain at the transmitter. Any intermediate value is referred to as a *gray level*.

Let us explain the importance of our gray-level conversion when evaluating the performance of any pathloss estimation. We evaluate performance of any approximation $\tilde{f} : \mathcal{D} \to \mathbb{R}$ of a signal/image $f : \mathcal{D} \to \mathbb{R}$, where $\mathcal{D} = \{x_n\}_n$ is some finite grid in $\mathbb{R}^2$, via the relative error

$$E = \frac{\sum_n |\tilde{f}(x_n) - f(x_n)|^2}{\sum_n |f(x_n)|^2}. \tag{10.7}$$

We also call (10.7) normalized mean square error (NMSE). The numerator in (10.7) represents the absolute error, and the denominator represents the global magnitude of $f$. The coefficients $|\tilde{f}(x_n) - f(x_n)|^2$ and $|f(x_n)|^2$ having larger values affect the outcome of $E$ the most, and small values are negligible. It is thus crucial to express the signal $f$ in a representation in which the important parts of the signal obtain large values.

In our case, the representation of the radio map should be constructed in such a way that small powers contribute small values to $E$. Indeed, locations of small power represent a weak signal. If we represent the radio map as standard pathloss, in dB, the smaller the power in a certain location, the higher the magnitude of the pathloss, with negative sign. When the power goes to zero, the pathloss diverges to $-\infty$. In this representation, locations of a weak signal dominate the global magnitude of the radio map and in general define a misleading concept of the "size" of the radio map. A similar situation occurs for the absolute error (the numerator of (10.7)).

As discussed in Sect. 10.3.1, motivated by the GDoF region of a Gaussian interference network, we know that very large negative values of the pathloss are effectively irrelevant and should not dominate the overall error. Our gray-level conversion resolves this issue. Indeed, anything below the noise floor, or more generally, below $\mathrm{P}_{\mathrm{L,trnc}}$, is deemed to be "too small to be interesting," and set to zero. In contrast, the values of higher power, which are most important, are transformed to levels close to 1. We note that papers like [10, 25, 50] suffer from the aforementioned shortcoming, and it is thus difficult to interpret their reported performance.

When root mean square error (RMSE) is used, the gray-level error is simply a scaling of the RMSE of the pathloss in dB (up to the truncation below the analytic noise floor). More precisely, we have

$$\sqrt{\sum_n |\tilde{\mathrm{P}}_{\mathrm{L}}(x_n) - \mathrm{P}_{\mathrm{L}}(x_n)|^2} = C\sqrt{\sum_n |\tilde{f}(x_n) - f(x_n)|^2},$$

where $\mathrm{P}_{\mathrm{L}}$ is the pathloss in dB. For $\mathrm{SNR}_{\mathrm{thr}} = 0$, we have $C = 80$.

## 10.5 Estimating Radio Maps via RadioUNets

In this section, we introduce a number of methods, collectively called RadioUNet, that learn to estimate radio maps in different scenarios. We evaluate the accuracy of the proposed methods and compare them to state of the art.

### *10.5.1 Motivation for RadioUNet*

UNets have been extensively applied to imaging problems in the past few years with resounding success and are considered to be a baseline method for image-to-image tasks [16]. Our problem can be seen as mapping an image representing the city and Tx to an image representing the radio map, and hence using UNets is a natural choice. One advantage in using UNets in our case is that they respect the translation invariance symmetry of the physical phenomenon. Namely, this symmetry is built into RadioUNet and requires no training. Another strong point of UNets is the encoder–decoder interpretation, as we discuss next.

In Fig. 10.2, we show an example of a ground truth radio map generated by simulation and the estimated radio map computed by the $\text{RadioUNet}_{\text{C}}$ and $\text{RadioUNet}_{\text{S}}$. Aside from the low quantitative error, RadioUNet seems to synthesize radio maps from the urban geometry that qualitatively captures the correct shadow patterns. Note that the results in Fig. 10.2 are representative of the general quality

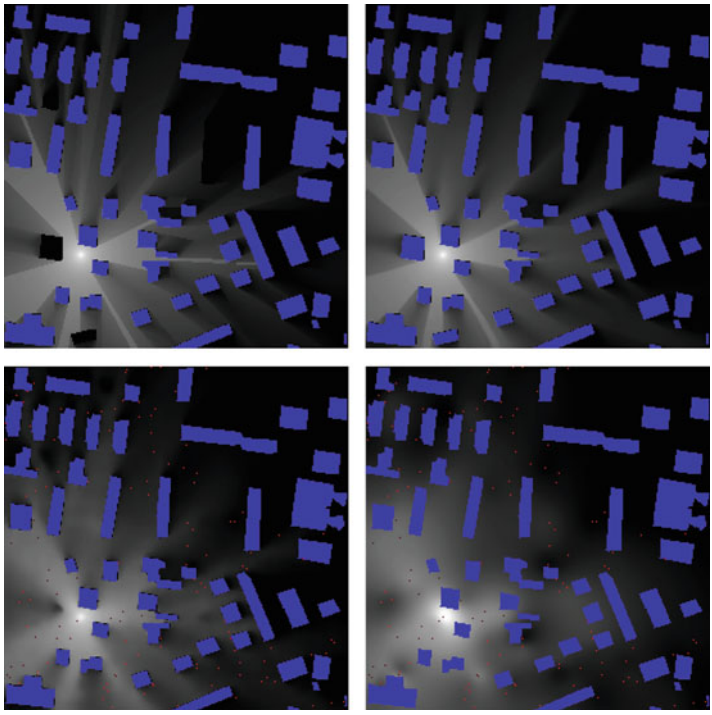


**Fig. 10.2** Comparison of RadioUNet with RBF with four missing buildings in the input. Top left: ground truth radio map. Top right: $\text{RadioUNet}_{\text{c}}$ with all buildings. Bottom left: $\text{RadioUNet}_{\text{S}}$ with missing buildings. Bottom right: RBF. The measured 127 locations for both RadioUNetS and RBF are marked in red. For RBF, the transmitter is also a measurement, and the known buildings are set to zero post-processing. Known buildings are marked in blue

of RadioUNet. One might naively interpret the success of the RadioUNet by postulating that it learns to mimic a physical model, such as ray tracing, or some differential equation such as Maxwell's equations. However, we believe that this is a misleading viewpoint. A more reasonable interpretation follows from the encoder–decoder description of general UNets. In the encoder path, the RadioUNet extracts complicated concepts about the geometry of the urban environment and the mutual relationship between the different geometric features, their location, and the location of the transmitter. Then, in the decoder path, the RadioUNet uses these concepts to synthesize the radio map. Thus, RadioUNet is based on extracting and analyzing *global* information about the urban environment, as opposed to classical physical models that are based on *local* information, such as collisions with the geometry in ray tracing and derivatives in differential equations. In this viewpoint, it is more fitting to compare RadioUNet to a highly skilled artist that draws radio maps from his/her perception of the urban environment as a whole, rather than comparing to a classical local physical model.

## *10.5.2 Different Settings in Radio Map Estimation*

We consider the following scenarios for the input of the UNet, the map of the city, the learning setting, and the properties of the simulated dataset. The problem setting can be any combination of the choices presented in Sects. 10.5.2.1 and 10.5.2.2. We remind here again that, in the following, the term "feature channel" is understood in the deep learning sense as a gray-level image, not to be confused with the notion of communication channel.

### 10.5.2.1 Network Input Scenarios

**City Map and Transmitter Location** In the first case, the UNet receives as input the map of the city and the Tx location as morphological images. The Tx feature channel is an image where the pixel in which the Tx is located is white, and the rest is black. From these two input feature channels, the network estimates the radio map.

In this *accurate map scenario*, if the simulated dataset without cars is used, then the map without cars is given as input, while if the simulated dataset includes cars, then the map without cars is given as one feature channel, and the cars in an additional feature channel.

When the map is accurate and the simulated data used for training is assumed to represent reality accurately, the radio map is uniquely determined by the map and the Tx location. Thus, these two input feature channels are sufficient for high-quality radio map reconstruction.

**City Map, Transmitter Location, and Measurements** In the second case, the UNet receives as input the two/three feature channels of map and Tx location as before, and an additional feature channel of measurements of the "true" radio map. The measurements are taken at some locations on the true map, i.e., their values are sampled from the target "ground truth." This third feature channel is given as a gray-level image, where in the pixels corresponding to the locations of the measurements the gray-level value is the measurement. Non-measured pixels are set to zero. The network simulates/estimates the radio map from these three/four input feature channels.

This scenario is useful when the "nominal" map given as input feature channel does not represent reality completely accurately. Hence, the network learns a hybrid of a radio map estimation method based on the given map, which is not completely reliable, and an interpolation method of the accurate pathloss measurements. In this *non-accurate maps scenario*, a perturbed version of the ground truth maps is given as input to the UNet. We consider two types of perturbations: (1) the map is given with a one to four missing buildings; (2) the map is given without cars, but the ground truth simulation is computed with the cars.

Another source of inaccuracy, for which relying on measurements is useful, is the fact that training is done against simulations, which are only approximations of reality, or in our setting, approximations of IRT4.

#### 10.5.2.2 Learning Scenarios

**Large and Dense Simulation Dataset** Here, the network is trained in supervised learning to predict a large dataset of 2D gray-level images representing dense measurements of radio maps on a fine grid. The images are the DPM simulations, the IRT2 simulations, both with or without cars, or random combinations of DPM and IRT2. In particular, the goal in the randomized simulation is to push the network to learn that it can only rely on the simulations for the big-picture behavior of radio maps, shared both by DPM and IRT2, but not on the fine details. This pushes the network to use additional information for refining the estimations, such as the input measurements if given, or the smaller dataset of sparse IRT4 if given.

Transferring the trained network to the ground truth (representing real-life maps) is a *zero-shot generalization*. Namely, the network only learned to estimate the coarser simulations, not ever seeing the ground truth phenomenon, and we rely on the accuracy of the simulations, and optionally on the measurements, to predict the ground truth radio maps.

In case measurements are given as an input feature channel to the RadioUNet, real-life measurements would be given to the RadioUNet in the real-time operations, even though measurements from the crude simulation are used in training. Real-life measurements can be provided in real time directly from the deployed devices, e.g., from the beacon signals of the transmitters, in the same way current systems report "Channel Quality Indicators" as measurements of the received signal strength. Hence, no costly measurement campaign is needed for training. The network

can generalize well to real-life radio maps since it learned to interpolate the measurements, which are now accurate, while what was learned from the crude simulations roughly guides the interpolation procedure to be physically feasible. We demonstrate this experimentally by training on coarse simulations and using IRT4 samples and targets (as a proxy for real-life measurements) in testing.

**Large and Dense Simulation Dataset + Small Sparse Measured Dataset** Here, in addition to the large dataset of dense measurements, we also assume that we have a small dataset of sparse measurements taken from real life (in our case, the high-accuracy ground truth IRT4 simulations). For each of the 700 maps of the RadioMapSeer dataset, we consider two transmitter locations and measurements in $K$ receiver locations, where $K$ is fixed, e.g., $K = 300$. Higher $K$ leads to better performance, but also to a more extensive and costly measurement campaign. The choice of $K = 300$ was taken to crudely balance the trade-off.

In this scenario, we first train a large network that estimates the crude simulations, using the large simulation dataset. Then, we improve the network output, using a smaller network, to match the small dataset of real-life measurements.

### *10.5.3 RadioUNet Architectures*

The simplest RadioUNet comprises one UNet. The input of the UNet has two, three, or four feature channels, depending on if measurements and cars are used, and the output is the one feature channel-estimated radio map. In most architectures of RadioUNet, we compose a second UNet on the first one. We call such an architecture a WNet (U+U makes a W). The inputs of the second UNet are the same as the inputs of the first UNet, plus an additional feature channel, the output of the first UNet. The architectures of our proposed UNets are reported in Table 10.2. The second UNet can be used for three different purposes, summarized in the following three subsections.

#### 10.5.3.1 Retrospective Improvement

The idea here is to give RadioUNet a chance to improve its estimation in retrospective. The first UNet learns implicitly an algorithm for estimating the radio map from the input, by extracting high-level concepts from the map and synthesizing a radio map from them. The philosophy here is that it would be beneficial to inspect the resulting estimation and correct visible inconsistencies with the map and with the physical phenomenon. To inspect the output of the first UNet, a second UNet extracts high-level concepts from the estimated radio map and the city map and synthesizes from these concepts an improved estimation of the radio map. We observe that the retrospective improvement yields better performance especially

**Table 10.2** RadioUNet architecture. *Resolution* is the number of pixels of the image in each feature channel along the $x$, $y$ axis. *Filter* is the number of pixels of each filter kernel along the $x$, $y$ axis. The input layer is concatenated in the last two layers

| First UNet | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Layer | In | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | Out |
| Resolution | 256 | 256 | 128 | 64 | 64 | 32 | 32 | 16 | 8 | 4 | 8 | 16 | 32 | 32 | 64 | 64 | 128 | 256 | 256 | 256 |
| Channel | 2/3/4 | 6 | 40 | 50 | 60 | 100 | 100 | 150 | 300 | 500 | $300+300$ | $150+150$ | $100+100$ | $100+100$ | $60+60$ | $50+50$ | $40+40$ | $20+6+2/3/4$ | $20+2/3/4$ | 1 |
| Filter size | 3 | 5 | 5 | 5 | 5 | 3 | 5 | 5 | 5 | 4 | 4 | 4 | 3 | 6 | 5 | 6 | 6 | 5 | 5 | – |
| Second UNet | | | | | | | | | | | | | | | | | | | | |
| Layer | In | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | Out |
| Resolution | 256 | 256 | 128 | 64 | 64 | 32 | 32 | 16 | 8 | 4 | 8 | 16 | 32 | 32 | 64 | 64 | 128 | 256 | 256 | 256 |
| Channel | 3/4/5 | 20 | 30 | 40 | 50 | 60 | 70 | 90 | 110 | 150 | $110+110$ | $90+90$ | $70+70$ | $60+60$ | $50+50$ | $40+40$ | $30+30$ | $20+20+3/4/5$ | $20+3/4/5$ | 1 |
| Filter size | 3 | 5 | 5 | 5 | 5 | 3 | 5 | 5 | 5 | 4 | 4 | 4 | 3 | 6 | 5 | 6 | 6 | 5 | 5 | – |

when the first UNet is small (see Fig. 10.3a). This WNet is thus a technique for reducing the size of the RadioUNet without degrading performance.

The WNet is trained in a curriculum. The first UNet is trained first to estimate the coarse radio maps, with MSE loss. In the second phase, the weights of the first UNet are frozen, and the second UNet is trained to estimate the ground truth radio maps with MSE loss.

#### 10.5.3.2 Adaptation to Real Measurements

Here, we first train the first UNet to estimate coarse simulations from the large dataset with MSE loss. The simulations may be randomized or deterministic. After training, the weights of the first UNet are frozen, and the second UNet is trained to improve the estimation of the first UNet on the small dataset of IRT4.

The IRT4 training consists of sparse images, namely, for each map, there are $K$ Rx locations $\{x_k\}_{k=1}^K$, and the pathloss $f(x_k)$ is only known for these locations. We typically take $K = 300$. The loss function for the second UNet is the weighted MSE, with weights $W_k = \frac{1}{K}$ for the points $\{x_k\}_{k=1}^K$, and weight 0 for the unmeasured points. We train the adaptation UNet in two steps. First, we train a retrospective improvement UNet on the coarse dataset, and then we further train this UNet on the sparse IRT4 dataset.

#### 10.5.3.3 Thresholder

A thresholder second UNet is used in the service area classification method. The goal of the second UNet here is to take the estimated radio map of the first UNet and to produce a service map from it. The service map is, roughly speaking, the area in the city in which the signal strength is high enough to be detected. More details are given in Sect. 10.7.1.

### *10.5.4 Training*

The 700 maps of the RadioMapSeer dataset are split into 500 training maps, 100 validation maps, and 100 test maps. The realization of the random split is fixed and available in the project web page.[4] We perform supervised learning on the RadioMapSeer dataset. The loss function is the MSE between the inferred radio maps by RadioUNet and the simulation radio maps from the training set. Training of all methods was performed with Adam [21], with a learning rate of $10^{-4}$. We take 50 epochs for each UNet, no regularization, and batch size 15. To alleviate overfitting,

[4] https://github.com/RonLevie/RadioUNet.

out of the 50 epochs, we pick the model with smallest error in the validation set. Lastly, the models are tested either on the coarse simulations on the test maps or on the IRT4 simulations on the test maps. Performance is evaluated by RMSE on the gray levels and by NMSE (normalized MSE). Note that the RMSE in dB is 80 times the RMSE of gray level.

### *10.5.5 RadioUNet Performance*

In Table 10.3, we report the results in all of the above settings. Recall that RadioUNet$_{\mathrm{C}}$ and RadioUNet$_{\mathrm{S}}$ denote the RadioUNet based on no input measurements and input measurements, respectively. From the table, we can observe that both the adaptation method to sparse IRT4 samples and the training with randomized coarsely simulated maps promote transferability. All accuracies are given in both NMSE and RMSE. RMSE is the square root of the MSE on the whole test set. The pathloss threshold is taken as $\mathrm{P_{L,thr}} = -127\,\mathrm{dB}$. The best results on IRT4 for each category are marked in bold face. RadioUNet$_S$ was trained and tested with a random number of input measurements between 1 and 300. *Zero-shot IRT4* means testing the methods, trained on coarse simulations, on IRT4. *Adaptation to IRT4* means training a second small UNet to match the sparse IRT4 measurements. All architectures are based on the WNets of Table 10.2, where for zero-shot transfer, the second UNet is a retrospective improvement, and for adaptation to sparse IRT4, the second UNet is the adaptor. The receiver points of the sparse IRT4 dataset are randomly generated for each map and fixed forever. For RadioUNet$_C$, the sparse IRT4 dataset has 300 receivers per transmitter. For RadioUNet$_S$, the sparse IRT4 dataset has 600 receivers per transmitter, and out of them 1 to 300 random points are taken as input points of the RadioUNet$_S$. The training loss is computed for all 600 points. To show that the higher transferability of the random simulations is not simply because IRT2 is closer to IRT4 than DPM, we also include the scenario where the deterministic simulation is IRT2. This produces inferior results to the random simulations.

In Fig. 10.3a, we compare RadioUNet$_{\mathrm{C}}$ with and without retrospective improvement for different pathloss thresholds. The results demonstrate that the retrospective improvement UNet is effective when the first UNet is small, thus making it a useful strategy for reducing the network size for the same accuracy. In Fig. 10.3b, we compare the performance of different RadioUNet$_{\mathrm{S}}$ methods on maps with various numbers of missing buildings. We observe that the strategy of combining random coarse simulations with an adaptor UNet to IRT4 promotes transferability.

In Fig. 10.4, we show some examples of RadioUNet$_{\mathrm{S}}$ with input maps missing 4 buildings and adapted to sparsely measured IRT4 simulations.

**Table 10.3** Comparison of RadioUNet accuracy in different scenarios

| RadioUNet Accuracy | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | RadioUNet$_c$ Test Error | | | | | | RadioUNet$_s$ Test Error | | | | | |
| | Coarse simulations | | Zero-Shot IRT4 | | Adaptation to IRT4 | | Coarse simulations | | Zero-Shot IRT4 | | Adaptation to IRT4 | |
| Setting | NMSE | RMSE | NMSE | RMSE | NMSE | RMSE | NMSE | RMSE | NMSE | RMSE | NMSE | RMSE |
| Accurate map | | | | | | | | | | | | |
| Deterministic DPM simulation | 0.0075 | 0.02 | 0.0284 | 0.0384 | 0.0166 | 0.0292 | 0.0052 | 0.0164 | 0.0183 | 0.0307 | 0.0135 | 0.0262 |
| Deterministic IRT2 simulation | 0.0219 | 0.032 | – | – | 0.0143 | 0.0271 | – | – | – | – | – | – |
| Non-deterministic simulation | 0.0152 | 0.0272 | 0.0324 | 0.0405 | **0.0135** | **0.0262** | 0.0068 | 0.0183 | 0.0122 | 0.0245 | **0.0086** | **0.0209** |
| Missing Four Buildings | | | | | | | | | | | | |
| Deterministic simulation | 0.102 | 0.0742 | 0.1205 | 0.0759 | 0.1015 | 0.0735 | 0.0321 | 0.0415 | 0.0409 | 0.0474 | 0.04 | 0.043 |
| Non-deterministic simulation | 0.1156 | 0.0769 | 0.1153 | 0.0783 | **0.1013** | **0.0726** | 0.0443 | 0.039 | 0.0417 | 0.0437 | **0.0372** | **0.041** |
| Cars | | | | | | | | | | | | |
| Deterministic simulation with unknown cars | 0.0132 | 0.0256 | 0.0357 | 0.0412 | **0.0249** | **0.0343** | 0.0072 | 0.0187 | 0.0197 | 0.0304 | **0.0156** | **0.0269** |
| Deterministic simulation with input cars | 0.0092 | 0.0207 | 0.0315 | 0.0385 | **0.0201** | **0.0308** | 0.0062 | 0.0173 | 0.0195 | 0.0305 | **0.0156** | **0.027** |

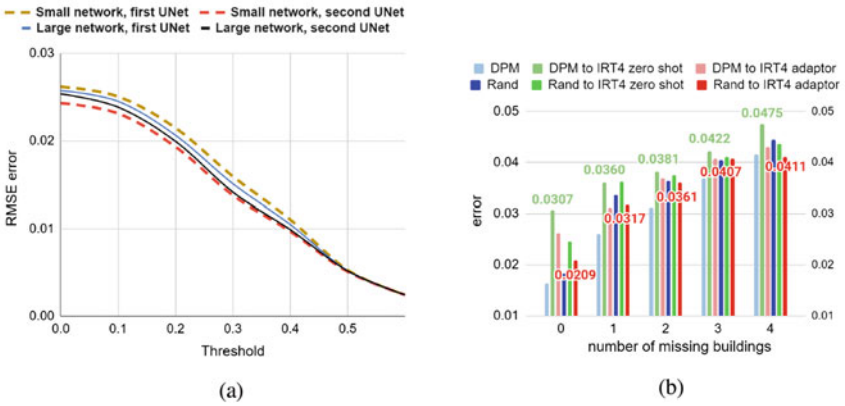



**Fig. 10.3** RadioUNet performance. (**a**) Test error in RadioUNet$_C$ of difference network sizes and different pathloss thresholds $P_{L,thr}$. The small network has 6,109,271 parameters, and large has 25,411,831 parameters. We plot the error of the RadioUNets with and without the retrospective improvement. Small networks outperform large networks when both have retrospective improvement. The error of RadioUNet with pathloss thresholds at pixel value 0.6 is comparable to the quantization error of the .png image file. (**b**) Test error in RadioUNet$_S$ with different numbers of missing buildings, different types of coarse simulations, and different transfer methods to sparse IRT4

## 10.6 Comparison of RadioUNet to State of the Art

In Fig. 10.5, we present the performance of different methods of radio map estimation. For methods that depend on samples, we use an input map with four missing buildings, and for methods that do not rely on samples, we use the full map. Apart from the fact the RadioUNet outperforms the data-driven interpolation methods, the tomography method, and the previously proposed deep learning approach significantly, these other methods need a separate training/optimization to fit the model to *each* map. Particularly, variations in the environment, such as moving cars, require re-computing the methods, which is not efficient. RadioUNets, in comparison, are trained offline only once and are then employed in any environment very efficiently. RadioUNet can deal with cars by using the measurements input, where the network is trained on a dataset of simulations with cars. All GPU methods ran on Nvidia Quadro GP100, and CPU methods on Intel Core i7-8750H.

### *10.6.1 Comparison to Model-Based Simulation*

We compare the run time of RadioUNet[5] with DPM, IRT2, and IRT4. To penalize RadioUNet, we compare run time on an Intel Core i7-8750H CPU, which is a highly

[5] Notice that the run time is the computation time of the *trained* network. This does not include the training, which is done offline and once for all.

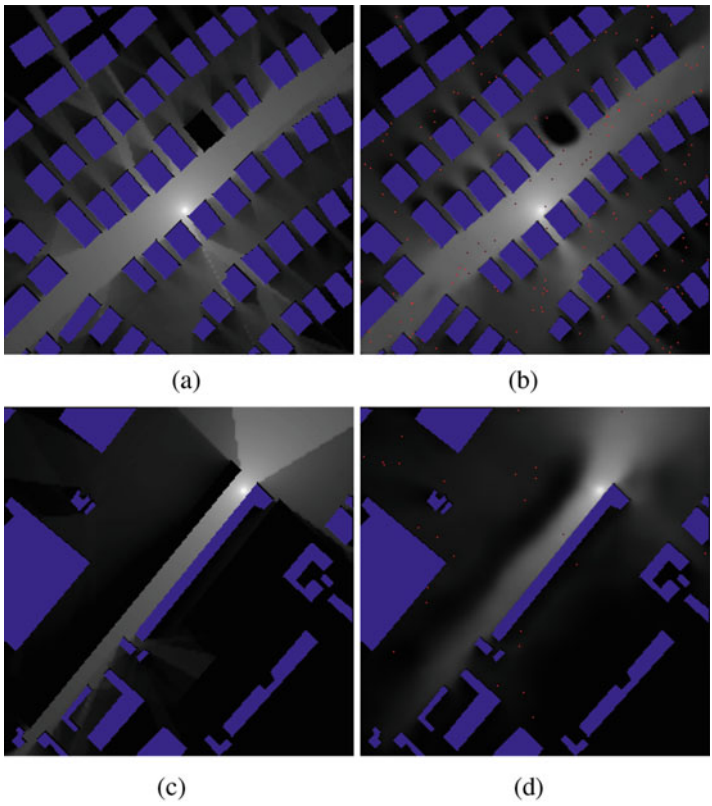



**Fig. 10.4** RadioUNet$_S$ test results on four missing buildings and adaptation to IRT4 simulation. The input buildings are in blue and measurements in red. (**a**) IRT4 target. (**b**) IRT4 prediction. (**c**) IRT4 target. (**d**) IRT4 prediction

non-optimal platform for convolution networks. RadioUNet estimates radio maps roughly one to three orders of magnitude faster than the simulation methods. In our experiments, WinProp completes a DPM simulation in roughly an order of 1 s on the CPU, and IRT2 and IRT4 take orders of 10 s and $10^2$ s, respectively. RadioUNet takes an order of $10^{-1}$ s on the CPU, and $10^{-3}$ s to $10^{-2}$ s on NVIDIA Quadro GV100 GPU.

### *10.6.2 Comparison to Data-Driven Interpolation*

Next, we compare RadioUNet$_C$ and RadioUNet$_S$ with data-driven interpolation methods: radial basis function (RBF) interpolation using multiquadric function [7, Sect 5.1] and tensor completion [50]. For the data-driven methods, we set to zero

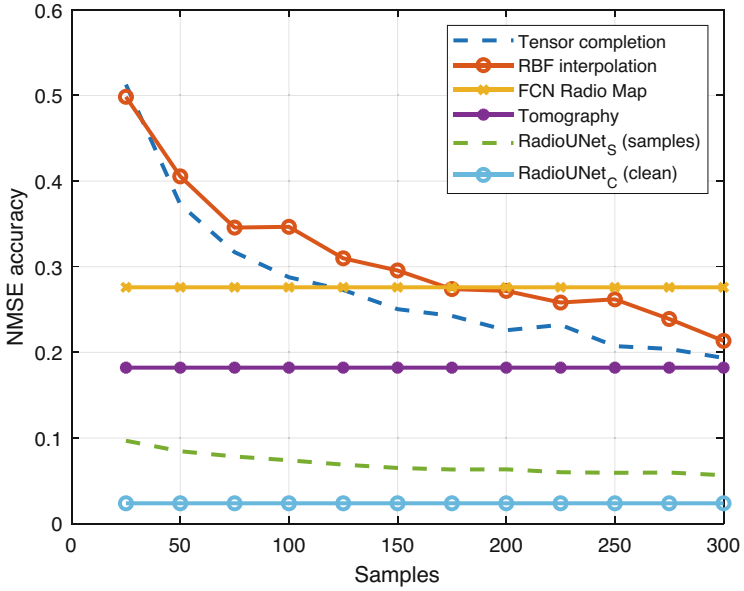



**Fig. 10.5** Estimation error of the radio map reconstruction methods as a function of the number of measurements. We chose Map 12 from the test set, on which RadioUNet performs worse than the average test map (three times the average NMSE). RadioUNet$_C$, tomography, and deep learning one step (FCN) are based on no samples and are given as horizontal baselines

the gray-level values inside the known buildings of the map post-processing, thus using the urban geometry data. Without this step, data-driven interpolation methods obtain a very poor accuracy since they are not able to recover the sharp building edges. In Fig. 10.5, we plot the average NMSE over 80 Tx's of RadioUNet$_S$ and of the two data-driven interpolation methods as a function of the number of samples. Both versions of RadioUNet clearly outperform state of the art. Aside from that, RadioUNet is roughly three orders of magnitude faster than RBF interpolation and five orders of magnitude faster than tensor completion interpolation.

### *10.6.3 Comparison to Model-Based Data Fitting*

We compare RadioUNet with a tomography method. In general, tomography methods model the attenuation in the channel strength as the sum of a distance-dependent pathloss and a shadowing term that models the attenuation due to obstructions. To model shadowing, a spatial loss field $L : \mathbb{R}^2 \to \mathbb{R}$ (SLF) is defined. For each spatial location $y$, the value $L(y)$ in a sense models the transparency of $y$, where $L(y) = 0$ models free space, and $L(y) > 0$ represents a "translucent" obstacle. The shadowing term from the Tx location $x$ to the Rx location $y$ is

computed as the integral of $L$ in a narrow oval for which the transmitter and receiver sit on the edges of the largest diameter. More generally, the oval can be replaced by some other shape, which may be trainable.

Note that as opposed to ray-tracing methods, tomography methods do not consider at all wave propagation phenomena such as diffraction and reflections and only model the attenuation due to the penetration of the signal through material. For high-frequency signals, the attenuation due to penetration in urban environments is very large, which make tomography method less realistic than DPM and IRT.

In tomography methods (e.g., [15, 24–26, 47]), the SLF is typically estimated from observed pathloss values between samples' transmitter–receiver pairs, by solving an inverse problem. In our situation, the problem is easier, since we are given the city map. Thus, the SLF outside the buildings, in free space, is known to be zero. Moreover, the building material is constant, and thus it is natural to consider an SLF with value $f$ inside buildings and 0 outside. Hence, the computation of the SLF is reduced to finding the scalar $f$ for which the tomography method gives a radio map as close as possible to the ground truth radio map. This method takes an order of $10^2$ s to run.

### *10.6.4 Comparison to Deep Learning Data Fitting*

We compare RadioUNet to the deep learning one-step prediction approach of [49]. We note that the two-step prediction approach of [49] did not perform well in our setting. As explained in Sect. 10.1.3, this method is a data fitting of a fully connected neural network to a 4D radio map of a specific city map. The network receives the transmitter and receiver 2D locations and returns the estimation of the pathloss for this pair. The network architecture is reported in Table 10.4. For a fixed map, the 80 transmitters are split into 60 training, 10 validation, and 10 test transmitters. The network is trained and tested against all receiver locations in the 256×256 grid. This method takes an order of 10sec to estimate all 256×256 pixels, which must be computed separately.

**Table 10.4** Architecture of the fully connected network of [49]

| Fully connected radio map network | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Layer | In | 1 | 2 | 3 | 4 | 5 | 6 | 7 | Out |
| Neurons | 4 | 64 | 200 | 2000 | 4000 | 4000 | 2000 | 64 | 1 |

## 10.7 Applications

In this section, we demonstrate the usefulness of RadioUNet with two simple applications and also discuss some future applications as future work.

### *10.7.1 Coverage Classification*

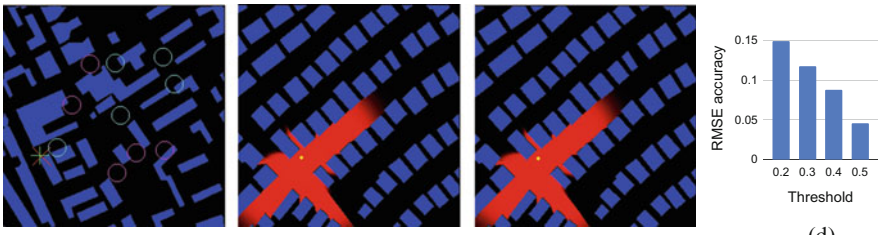
Service area classification shows up in two situations. In the first problem, given a Tx–Rx link, we would like to know if the received signal strength is large enough. In the second problem, given two Tx–Rx links, we would like to know if the interference caused by one link on the other is low enough. In both cases, the goal is to classify if the pathloss of a certain Tx is above or below some threshold at the location of some Rx. For a fixed Tx location $x$, let $f(y)$ denote the radio map at location $y$. We define the *coverage map* as the thresholding function

$$C(y) = \begin{cases} 0 & \text{if } f(y) \leq T, \\ 1 & \text{if } f(y) > T, \end{cases} \tag{10.8}$$

where $T$ is a threshold in gray scale. For the first problem, depending on the system requirements, $T$ is some value above the noise floor. For example, for high bit rates, the signal has to arrive with high SNR, so a typical value for $T$ might be pixel value 0.5 (see, e.g., [13]). For the second problem, a typical choice for $T$ is the noise floor, which is pixel value 0.2 for us.

Our goal is to predict the coverage map from the input city and transmitter location. Note that in principle UNets are expressive enough to predict coverage maps, since coverage maps are a sub-phenomenon of radio maps, and UNets are expressive enough to predict radio maps. However, this naive point of view disregards the fact that the gradient descent optimization procedure is highly non-exhaustive and only searches parameter configurations along a 1D path. As it turns out, simple UNets fail to learn meaningful predictions of coverage maps. Intuitively, radio maps are more predictable than coverage maps since shadow patterns are always associated with simple concepts such as building corners and spatial relations between buildings, receiver locations, and the location of the transmitter. In contrast, in the coverage map, most shadow edges disappear and are "absorbed" by one of the domains above or below $T$.

For the architecture to successfully predict the coverage map, it must first understand the underlying phenomenon of radio maps. We thus consider a WNet architecture. The first UNet is RadioUNet, which predicts the radio map from the city and transmitter inputs. The second UNet receives the predicted radio map as input, along with the map and the transmitter location, and computes the coverage map from them. We call the second UNet the thresholding UNet or TUNet. We call this architecture the Coverage WNet, or CWNet in short.




(a) (b) (c) (d)

**Fig. 10.6** **Left**: Localization result. Green +: true Rx position, red X: estimated Rx position, yellow: pixels of the localization intersection, magenta circles: Tx's of the best localization result out of the $R$, green circles: the rest of the Tx's. **Middle**: Coverage map results with threshold 0.5. Red: coverage map. Blue: city map. Yellow: transmitter. **Right**: accuracy of service map estimation for different thresholds in RMSE. (**a**) Localization. (**b**) True truth coverage map. (**c**) Estimated coverage map. (**d**) Accuracy of service map estimation

To train the CWNet, we use curriculum learning. We first train the RadioUNet as before. We then freeze the RadioUNet and train the TUNet in a curriculum as explained next. As it turns out, the discontinuous nature of the coverage map is still too challenging for the TUNet to learn directly. Instead, we relax the coverage map to a soft coverage map $C_\alpha(y) = \text{sigmoid}\Big(\alpha\big(f(y) - T\big)\Big)$, where $\alpha$ is a parameter that determines how soft the transition between 0 and 1 is. We interpret $C_\alpha(y)$ as the probability of location $y$ being in the coverage area. In the curriculum, we first train the TUNet to predict $C_\alpha(y)$ with $\alpha = 1$ and gradually increase $\alpha$. We end up with $\alpha = 128$, which we judge to be high enough to represent a sharp transition.

The accuracy of SWNet for different thresholds and an example service map are presented in Fig. 10.6.

### *10.7.2 Pathloss-Based Fingerprint Localization*

Suppose that a device is simultaneously in the coverage area of several base stations located at Tx points $x_1, \ldots, x_K$ and reports the strengths $g_k$ (converted into gray scale) of their corresponding beacon signals. Let $f_k(y)$ denote the estimated radio map for Tx location $x = x_k$, for $k = 1, \ldots, K$. For some $\epsilon > 0$, we define the $\epsilon$-level set for level $g_k$ as

$$L_k^\epsilon = \{z \in \Gamma \ : \ |f_k(z) - g_k| \leq \epsilon\}, \tag{10.9}$$

where $\Gamma$ is the discrete grid (domain of the radio map, in our case the 256×256 grid). Then, in order to identify the location of the receiver $y$, we can consider the intersection of the $\epsilon$-level sets $S = \bigcap_{k=1}^K L_k^\epsilon$. If this set is localized about a single point, then we have located $y$ with high probability.

Assuming that the reported values $\{g_k\}_k$ are equal to the true radio map values, if for some $k$ the radio map prediction error satisfies $|f_k(y) - g_k| > \epsilon$, then $y \notin L_k^\epsilon$ and $y$ will not be contained in the intersection $S$. We call such $k$ an outlier. In contrast, if we choose $K$ too small, then $S$ will contain multiple points and the localization is ambiguous. Hence, the method works well when the estimated radio maps are accurate and the number of reported signal strengths $K$ is large enough but not too large.

To alleviate the effect of outliers, instead of computing a single intersection, we can select random subsets of $J < K$ Tx's and consider the intersection of the corresponding $\epsilon$-level sets. We also take random $\epsilon$ values for each map since different maps have different unknown accuracies. Repeating this random selection $R$ times, we generate $R$ candidate sets, some of which may be empty and some of which may contain multiple points. For the $R'$ non-empty outcomes, we compute a score for the quality of the result and pick the outcome with the best score. For example, we use the variance of the localization outcome. Let $S_t$ be the localization outcome of sample $t$, where $t = 1, \dots R'$. Then, we define the expected position given $S_t$ as $\widehat{y}_t = \sum_{z \in S_t} \frac{z}{|S_t|}$, and the associated variance

$$V_t = \sum_{z \in S_t} \frac{|z - \widehat{y}_t|^2}{|S_t|},$$

where $|z - \widehat{y}|$ is the Euclidean distance between $z$ and $\widehat{y}_t$ in $\mathbb{R}^2$ and $|S_t|$ is the area of $S_t$. Since smaller variance means better localization, we pick the non-empty localization outcome with smallest variance. In this chapter, we mention this approach just as an example of the use of accurate radio map estimation. In future work, we will deal with improving the pathloss-based localization with more sophisticated localization extraction and using additional signal fingerprints.

In Fig. 10.6a, we present an example localization result with $K = 10$, $J = 5$, $R = 5$, $\epsilon = 0.03$. The best outcome has a standard deviation of 0.5 meters. The distance between the estimated and true receiver location is 1.58 meters.

## 10.8 Conclusion

In this chapter, we introduced RadioUNet, a deep learning method for simulating radio maps given a city geometry, Tx location, and optionally some pathloss measurements and car locations. For training RadioUNet, we introduced the new dataset RadioMapSeer, which we hope will be used for developing deep learning methods for pathloss prediction by other researchers as well. We developed approaches for transferring what was learned on the large dataset of coarsely simulated radio maps to real life and demonstrated the superior performance of our methods with respect to state of the art, both in run time and accuracy.

**Acknowledgments** The work presented in this chapter was funded by the DFG Grant DFG SPP 1798 "Compressed Sensing in Information Processing" through Project Massive MIMO-II. G.K. and G.C. acknowledge support by the Deutsche Forschungsgemeinschaft (DFG) through Projects KU 1446/21-2 and CA 1340/1-2, respectively, within SPP 1798.

## References

1. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X.: TensorFlow: Large-scale Machine Learning on Heterogeneous Systems (2015). https://www.tensorflow.org/. Software available from tensorflow.org
2. Ashikhmin, A., Marzetta, T.: Pilot contamination precoding in multi-cell large scale antenna systems. In: Proceedings of the IEEE International Symposium Information Theory, pp. 1137–1141. Cambridge, MA, USA (2012)
3. Badrinarayanan, V., Kendall, A., Cipolla, R.: Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **39**(12), 2481–2495 (2017). https://doi.org/10.1109/TPAMI.2016.2644615
4. Bengio, Y., Louradour, J., Collobert, R., Weston, J.: Curriculum learning. In: Proceedings of the 26th Annual International Conference Machine Learning (ICML '09), pp. 41–48. Association for Computing Machinery, New York (2009)
5. Bethanabhotla, D., Bursalioglu, O.Y., Papadopoulos, H.C., Caire, G.: Optimal user-cell association for massive MIMO wireless networks. IEEE Trans. Wireless Comm. **15**(3), 1835–1850 (2016)
6. Bianchi, G.: Performance analysis of the IEEE 802.11 distributed coordination function. IEEE J. Sel. Areas Commun. **18**(3), 535–547 (2000)
7. Bishop, C.M.: Neural Networks for Pattern Recognition. Oxford University Press, Inc., New York (1995)
8. Chen, Z., Sohrabi, F., Yu, W.: Sparse activity detection for massive connectivity. IEEE Trans. Sig. Proc. **66**(7), 1890–1904 (2018). https://doi.org/10.1109/TSP.2018.2795540
9. Chien, T.V., Canh, T.N., Björnson, E., Larsson, E.G.: Power control in cellular massive MIMO with varying user activity: A deep learning solution. CoRR **abs/1901.03620** (2019). http://arxiv.org/abs/1901.03620
10. Chouvardas, S., Valentin, S., Draief, M., Leconte, M.: A method to reconstruct coverage loss maps based on matrix completion and adaptive sampling. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 6390–6394. Shanghai, China (2016). https://doi.org/10.1109/ICASSP.2016.7472907
11. Çiçek, Ö., Abdulkadir, A., Lienkamp, S.S., Brox, T., Ronneberger, O.: 3D U-Net: Learning dense volumetric segmentation from sparse annotation. In: Medical Image Computing and Computer-Assisted Intervention (MICCAI 2016), pp. 424–432. Springer, Berlin (2016)
12. Cui, W., Shen, K., Yu, W.: Spatial deep learning for wireless scheduling. IEEE J. Sel. Areas Commun. **37**(6), 1248–1261 (2019)
13. ETSI: Intelligent Transport Systems (ITS); access layer specification for Intelligent Transport Systems operating in the 5 GHz frequency band. EN 302 663 V1.2.1, ETSI (2013)
14. Geng, C., Naderializadeh, N., Avestimehr, A.S., Jafar, S.A.: On the optimality of treating interference as noise. IEEE Trans. Inf. Theory **61**(4), 1753–1767 (2015)

15. Gutierrez-Estevez, M.A., Cavalcante, R.L.G., Stanczak, S.: Nonparametric radio maps reconstruction via elastic net regularization with multi-kernels. In: Proceedings of the IEEE International Workshop Signal Processing Advances Wireless Communication (SPAWC), pp. 1–5. Kalamata, Greece (2018). https://doi.org/10.1109/SPAWC.2018.8445843
16. Hauptmann, A., Adler, J.: On the unreasonable effectiveness of CNNS. arXiv preprint arXiv:2007.14745 (2020)
17. Hershey, J.R., Le Roux, J., Weninger, F.: Deep unfolding: Model-based inspiration of novel deep architectures. CoRR **abs/1409.2574** (2014)
18. Hoppe, R., Wölfle, G., Jakobus, U.: Wave propagation and radio network planning software WinProp added to the electromagnetic solver package FEKO. In: Proceedings of the International Application Computational Electromagnetics Society Symposium Italy (ACES), pp. 1–2. Florence, Italy (2017). https://doi.org/10.23919/ROPACES.2017.7916282
19. Imai, T., Kitao, K., Inomata, M.: Radio propagation prediction model using convolutional neural networks by deep learning. In: Proceedings of the European Conference Antennas and Propagation (EuCAP), pp. 1–5. Krakow, Poland (2019)
20. Jin, K.H., McCann, M.T., Froustey, E., Unser, M.: Deep convolutional neural network for inverse problems in imaging. IEEE Trans. Image Proc. **26**(9), 4509–4522 (2017). https://doi.org/10.1109/TIP.2017.2713099
21. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: Proceedings of the International Conference Learning Representation (ICLR). San Diego, CA, USA (2015)
22. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998)
23. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **512**, 436–444 (2015)
24. Lee, D., Giannakis, G.B.: A variational Bayes approach to adaptive channel-gain cartography. In: Proceedings of the IEEE International Conference Acoustics, Speech and Signal Processing (ICASSP), pp. 8434–8438. Brighton, United Kingdom (2019). https://doi.org/10.1109/ICASSP.2019.8683300
25. Lee, D., Kim, S., Giannakis, G.B.: Channel gain cartography for cognitive radios leveraging low rank and sparsity. IEEE Trans. Wireless Commun. **16**(9), 5953–5966 (2017). https://doi.org/10.1109/TWC.2017.2717822
26. Lee, D., Berberidis, D., Giannakis, G.B.: Adaptive Bayesian radio tomography. IEEE Trans. Sig. Proc. **67**(8), 1964–1977 (2019). https://doi.org/10.1109/TSP.2019.2899806
27. Levie, R., Yapar, C., Kutyniok, G., Caire, G.: Pathloss prediction using deep learning with applications to cellular optimization and efficient D2D link scheduling. In: ICASSP 2020—2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 8678–8682 (2020)
28. Levie, R., Yapar, C., Kutyniok, G., Caire, G.: RadioUNet: Fast radio map estimation with convolutional neural networks. IEEE Trans. Wirel. Commun. pp. 1–1 (2021). https://doi.org/10.1109/TWC.2021.3054977
29. Lim, B., Son, S., Kim, H., Nah, S., Lee, K.M.: Enhanced deep residual networks for single image super-resolution. In: Proceedings of the Conference Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 1132–1140. Honolulu, HI, USA (2017). https://doi.org/10.1109/CVPRW.2017.151
30. Litjens, G., Kooi, T., Bejnordi, B.E., Setio, A.A.A., Ciompi, F., Ghafoorian, M., van der Laak, J.A.W.M., van Ginneken, B., Sanchez, C.I.: A survey on deep learning in medical image analysis. Med. Image Anal. **42**, 60–88 (2017). https://doi.org/10.1016/j.media.2017.07.005
31. Marcus, G.: Innateness, AlphaZero, and artificial intelligence. arXiv preprint arXiv:1801.05667 (2018)
32. Mathieu, M., Couprie, C., LeCun, Y.: Deep multi-scale video prediction beyond mean square error. In: International Conference Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2–4, 2016. Conference Track Proceedings (2016). http://arxiv.org/abs/1511.05440
33. Milletari, F., Navab, N., Ahmadi, S.: V-Net: Fully convolutional neural networks for volumetric medical image segmentation. In: Proceedings of the International Conference 3D Vision (3DV), pp. 565–571. Stanford, CA, USA (2016). https://doi.org/10.1109/3DV.2016.79

34. Naderializadeh, N., Avestimehr, A.S.: ITLinQ: A new approach for spectrum sharing in device-to-device communication systems. IEEE J. Sel. Areas Commun. **32**(6), 1139–1151 (2014)
35. Nikitaki, S., Tsagkatakis, G., Tsakalides, P.: Efficient training for fingerprint based positioning using matrix completion. In: Proceedings of the 20th European Signal Processing Conference (EUSIPCO), pp. 195–199. Bucharest, Romania (2012)
36. OpenStreetMap contributors: Planet dump retrieved from https://planet.osm.org. https://www.openstreetmap.org (2017)
37. Ozyegen, O., Mohammadjafari, S., El Mokhtari, K., Cevik, M., Ethier, J., Başar, A.: Deep learning approaches for fast radio signal prediction. arXiv preprint arXiv:2006.09245 (2020)
38. Papyan, V., Romano, Y., Elad, M.: Convolutional neural networks analyzed via convolutional sparse coding. J. Mach. Learn. Res. **18**, 83:1–83:52 (2017)
39. Parera, C., Liao, Q., Malanchini, I., Tatino, C., Redondi, A.E.C., Cesana, M.: Transfer learning for tilt-dependent radio map prediction. IEEE Trans. Cognitive Commun. Netw. **6**(2), 829-843 (2020)
40. Park, S., Truong, A.Q., Nguyen, T.H.: Power control for sum spectral efficiency optimization in MIMO-NOMA systems with linear beamforming. IEEE Access **7**, 10593–10605 (2019)
41. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in PyTorch (2017)
42. Popescu, I., Nafomita, I., Constantinou, P., Kanatas, A., Moraitis, N.: Neural networks applications for the prediction of propagation path loss in urban environments. In: IEEE VTS 53rd Vehicular Technology Conference, Spring 2001. Proceedings (Cat. No.01CH37202), vol. 1, pp. 387–391 (2001)
43. Popoola, S.I., Jefia, A., Atayero, A.A., Kingsley, O., Faruk, N., Oseni, O.F., Abolade, R.O.: Determination of neural network parameters for path loss prediction in very high frequency wireless channel. IEEE Access **7**, 150462–150483 (2019)
44. Ratnam, V.V., Chen, H., Pawar, S., Zhang, B., Zhang, C.J., Kim, Y.J., Lee, S., Cho, M., Yoon, S.R.: FadeNet: Deep learning-based mm-wave large-scale channel fading prediction and its applications. IEEE Access **9**, 3278–3290 (2021). https://doi.org/10.1109/ACCESS.2020.3048583
45. Rautiainen, T., Wölfle, G., Hoppe, R.: Verifying path loss and delay spread predictions of a 3d ray tracing propagation model in urban environment. In: Proceedings IEEE 56th Vehicular Technology Conference, vol. 4, pp. 2470–2474. Vancouver, BC, Canada (2002)
46. Rizk, K., Wagen, J.F., Gardiol, F.: Two-dimensional ray-tracing modeling for propagation prediction in microcellular environments. IEEE Trans. Veh. Technol. **46**(2), 508–518 (1997)
47. Romero, D., Lee, D., Giannakis, G.B.: Blind radio tomography. IEEE Trans. Sig. Proc. **66**(8), 2055–2069 (2018). https://doi.org/10.1109/TSP.2018.2799169
48. Ronneberger, O., Fischer, O., Brox, T.: U-Net: Convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) Medical Image Computing and Computer-Assisted Intervention (MICCAI 2015), pp. 234–241. Springer, Cham (2015)
49. Saito, K., Jin, Y., Kang, C., Takada, J., Leu, J.S.: Two-step path loss prediction by artificial neural network for wireless service area planning. In: IEICE Communications Express (2019). https://doi.org/10.1587/comex.2019GCL0038
50. Schäufele, D., Cavalcante, R.L.G., Stanczak, S.: Tensor completion for radio map reconstruction using low rank and smoothness. In: Proceedings of the IEEE International Workshop Signal Processing Advances Wireless Communication (SPAWC), pp. 1–5. Cannes, France (2019). https://doi.org/10.1109/SPAWC.2019.8815495
51. Shelhamer, E., Long, J., Darrell, T.: Fully convolutional networks for semantic segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **39**(4), 640–651 (2017). https://doi.org/10.1109/TPAMI.2016.2572683
52. Shen, K., Yu, W.: FPLinQ: A cooperative spectrum sharing strategy for device-to-device communications. In: Proceedings of the IEEE International Symposium Information Theory (ISIT), pp. 2323–2327. Aachen, Germany (2017)

53. Sotiroudis, S.P., Siakavara, K.: Mobile radio propagation path loss prediction using artificial neural networks with optimal input information for urban environments. AEU Int. J. Electron. Commun. **69**(10), 1453–1463 (2015). https://doi.org/https://doi.org/10.1016/j.aeue.2015.06.014. http://www.sciencedirect.com/science/article/pii/S1434841115001855
54. Sotiroudis, S.P., Goudos, S.K., Gotsis, K.A., Siakavara, K., Sahalos, J.N.: Application of a composite differential evolution algorithm in optimal neural network design for propagation path-loss prediction in mobile communication systems. IEEE Antennas Wirel. Propag. Lett. **12**, 364–367 (2013)
55. Stein, M.L.: Interpolation of Spatial Data: Some Theory for Kriging. Springer, Berlin (2012)
56. Timoteo, R., Cunha, D., Cavalcanti, G.: A proposal for path loss prediction in urban environments using support vector regression. In: Proceedings of the Advance International Conference Telecommunication (AICT), vol. 2014, pp. 119–124. Paris, France (2014)
57. Utkovski, Z., Agostini, P., Frey, M., Bjelakovic, I., Stanczak, S.: Learning radio maps for physical-layer security in the radio access. In: Proceedings of the IEEE International Workshop Signal Processing Advances Wireless Communication (SPAWC), pp. 1–5. Cannes, France (2019). https://doi.org/10.1109/SPAWC.2019.8815467
58. Wahl, R., Wölfle, G., Wildbolz, P., Landstorfer, F.: Dominant path prediction model for urban scenarios. In: Proceedings of the IST Mobile and Wireless Communications (2005)
59. Wu, X., Tavildar, S., Shakkottai, S., Richardson, T., Li, J., Laroia, R., Jovicic, A.: FlashLinQ: A synchronous distributed scheduler for peer-to-peer ad hoc networks. IEEE/ACM Trans. Networking **21**(4), 1215–1228 (2013)
60. Yapar, C., Levie, R., Kutyniok, G., Caire, G.: Real-time localization using radio maps. arXiv preprint arXiv:2006.05397 [eess.SP] (2020)
61. Yi, X., Caire, G.: Optimality of treating interference as noise: A combinatorial perspective. IEEE Trans. Inf. Theory **62**(8), 4654–4673 (2016)
62. Yi, Z., Zhang, H., Tan, P., Gong, M.: DualGAN: Unsupervised dual learning for image-to-image translation. In: Proceedings of the IEEE International Conference Computer Vision (ICCV), pp. 2868–2876 (2017). https://doi.org/10.1109/ICCV.2017.310
63. Zhang, S., Zhang, R.: Radio map based path planning for cellular-connected UAV. In: Proceedings of the IEEE Global Communication Conference (GLOBECOM), pp. 1–6. Waikoloa, HI, USA (2019)
64. Zhang, X., Shu, X., Zhang, B., Ren, J., Zhou, L., Chen, X.: Cellular network radio propagation modeling with deep convolutional neural networks. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '20, pp. 2378–2386. Association for Computing Machinery, New York (2020)
65. Zugno, T., Drago, M., Giordani, M., Polese, M., Zorzi, M.: Towards standardization of millimeter wave vehicle-to-vehicle networks: Open challenges and performance evaluation. arXiv preprint arXiv:1910.00300 (2019)

# Chapter 11
# Active Channel Sparsification: Realizing Frequency-Division Duplexing Massive MIMO with Minimal Overhead


**Mahdi Barzegar Khalilsarai, Saeid Haghighatshoar, Xinping Yi , Giuseppe Caire, and Gerhard Wunder**

## 11.1 Introduction

Multiuser *multiple-input multiple-output* (MIMO) consists of exploiting multiple antennas at the base station (BS) side, in order to multiplex over the spatial-domain several data streams to a number of users sharing the same time–frequency transmission resource (channel bandwidth and time slots). For a block-fading channel with spatially independent fading and a coherence block of $T$ symbols,[1] the high-SNR sum capacity behaves as $C(\mathrm{SNR}) = M^*(1 - M^*/T)\log \mathrm{SNR} + O(1)$, where $M^* = \min\{M, K, T/2\}$, $M$ denotes the number of BS antennas, and $K$ denotes the number of single-antenna users [1, 39, 61]. When $M$ and the number

[1] This is the number of signal dimensions over which the fading channel coefficients can be considered constant over time and frequency [56].

M. Barzegar Khalilsarai (✉) · G. Caire
Department of Electrical Engineering and Computer Science, Technische Universität Berlin, Berlin, Germany
e-mail: m.barzegarkhalilsarai@tu-berlin.de; caire@tu-berlin.de

S. Haghighatshoar
SynSense AG, Zurich, Switzerland
e-mail: saeid.haghighatshoar@sysense.ai

X. Yi
Department of Electrical Engineering and Electronics, University of Liverpool, Liverpool, United Kingdom
e-mail: xinping.yi@liv.ac.uk

G. Wunder
Freie Universität Berlin, Berlin, Germany
e-mail: g.wunder@fu-berlin.de



G. Kutyniok et al. (eds.), *Compressed Sensing in Information Processing*, Applied and Numerical Harmonic Analysis, https://doi.org/10.1007/978-3-031-09745-4_11

of users are potentially very large, the system *pre-log factor*[2] is maximized by serving $K = T/2$ data streams (users). While any number $M \geq K$ of BS antennas yields the same (optimal) pre-log factor, a key observation made in [40] is that, when training a very large number of antennas comes at no additional overhead cost, it is indeed convenient to use $M \gg K$ antennas at the BS. In this way, at the cost of some additional *hardware complexity*, very significant benefits at the system level can be achieved. These include: (i) energy efficiency (due to the large beamforming gain); (ii) inter-cell interference reduction; (iii) a dramatic simplification of user scheduling and rate adaptation, due to the inherent large-dimensional channel hardening [34]. Systems for which the number of BS antennas $M$ is much larger than the number of DL data streams $K$ are generally referred to as *massive MIMO* (see [34, 40, 41] and references therein). Massive MIMO has been the subject of intense research investigation and development and is expected to be a cornerstone of the 5th generation of wireless/cellular systems [5].

In order to achieve the benefits of massive MIMO, the BS must learn the downlink (DL) channel coefficients for $K$ users and $M \gg K$ BS antennas. For time-division duplexing (TDD) systems, due to the inherent UL–DL channel reciprocity [39], this can be obtained from $K$ mutually orthogonal UL pilots transmitted by the users. Unfortunately, the UL–DL channel reciprocity does not hold for frequency-division duplexing (FDD) systems, since the UL and DL channels are separated in frequency by much more than the channel coherence bandwidth [56]. Hence, unlike TDD systems, in FDD, the BS must actively probe the DL channel by sending a common DL pilot signal and request the users to feed their channel state back.

In order to obtain a "fresh" channel estimate for each coherence block, $T_{\mathrm{dl}}$ out of $T$ symbols per coherence block must be dedicated to the DL common pilot. Assuming (for simplicity of exposition) a delay-free channel-state feedback, the resulting DL pre-log factor is given by $K \times \max\{0, 1 - T_{\mathrm{dl}}/T\}$, where $K$ is the number of served users, and $\max\{0, 1 - T_{\mathrm{dl}}/T\}$ is the penalty factor incurred by DL channel training. Conventional DL training consists of sending orthogonal pilot signals from each BS antenna. Thus, in order to train $M$ antennas, the minimum required training dimension is $T_{\mathrm{dl}} = M$. Hence, with such scheme, the number of BS antennas $M$ cannot be made arbitrarily large. For example, consider a typical case taken from the LTE system [53], where groups of users are scheduled over resource blocks spanning 14 OFDM symbols $\times$ 12 subcarriers, for a total dimension of $T = 168$ symbols in the time–frequency plane. Consider a typical massive MIMO configuration serving $K \sim 20$ users with $M \geq 200$ antennas (e.g., see [37]). In this case, the entire resource block dimension would be consumed by the DL pilot, leaving no room for data communication. Furthermore, feeding back the $M$-

[2] With this term, we indicate the number of spatial-domain data streams supported by the system, such that each stream has spectral efficiency that behaves as an interference-free Gaussian channel, i.e., $\log \mathrm{SNR} + O(1)$. In practice, although the system may be interference-limited (e.g., due to inter-cell interference in multicell cellular systems), a well-designed system would exhibit a regime of practically relevant SNR for which its sum rate behaves as an affine function of $\log \mathrm{SNR}$ [36].

dimensional measurements (or estimated/quantized channel vectors) also burdens the system with a significant UL feedback overhead [7, 27, 32, 35, 60].

While the argument above is kept informal on purpose, it can be made information-theoretically rigorous. The central issue is that, if one insists to estimate the $K \times M$ channel matrix in an "agnostic" way, i.e., without exploiting the channel fine structure, a hard dimensionality bottleneck kicks in and fundamentally limits the number of data streams that can be supported in the DL by FDD systems. It follows that gathering "massive MIMO gains" in FDD systems is a challenging problem. On the other hand, current wireless networks are mostly based on FDD. Such systems are easier to operate and more effective than TDD systems in situations with symmetric traffic and delay-sensitive applications [9, 26, 47]. In addition, converting current FDD systems into TDD would represent a non-trivial cost for wireless operators. With these motivations in mind, a significant effort is recently dedicated to reduce the common DL training dimension and feedback overhead in order to materialize the numerous benefits of massive MIMO also for FDD systems.

The focus of this chapter is to put forth an efficient scheme for massive MIMO in FDD systems. Our goal is to be able to serve as many users as possible even with a very small number of DL pilots, compared to the inherent channel dimension. Similar to previous works [11, 15, 47], we consider a scheme where each user sends back its $T_{\mathrm{dl}}$ noisy pilot observations per slot, using non-quantized analog feedback (see [7, 32]). Hence, achieving a small $T_{\mathrm{dl}}$ yields both a reduction of DL training and UL feedback overhead. It turns out that we have to *artificially* reduce each user channel dimension in a clever way, such that a single common DL pilot of assigned dimension $T_{\mathrm{dl}}$ is sufficient to estimate a large number of user channels. In the CS-based works mentioned above, the pilot dimension depends on the channel sparsity level $s$ (the number of non-zero components in the angle/beam domain). In fact, standard CS theory states that stable sparse signal reconstruction is possible using $T_{\mathrm{dl}} = O(s \log M)$ measurements.[3] In a rich scattering environment, $s$ is large or may in fact vary from user to user or in different cell locations. Even if the channel support is known, one needs at least $s$ measurements for a stable channel estimation. Hence, these CS-based methods (including the ones having access to support information) may or may not work well, depending on the propagation environment. In order to allow channel estimation with a given pilot dimension $T_{\mathrm{dl}}$, we use the DL covariance information in order to design an optimal *sparsifying precoder*. This is a linear transformation that depends only on the (estimated) channel covariances that impose that the effective channel matrix (including the precoder) has large rank and yet each of its columns has sparsity not larger than $T_{\mathrm{dl}}$. In this way, our method is not at the mercy of nature, i.e., it is flexible with respect to various types of environments and channel sparsity orders. We cast the optimization

[3] As commonly defined in the CS literature, we say that a reconstruction method is stable if the resulting MSE vanishes as 1/SNR, where SNR denotes the signal-to-noise ratio of the measurements.

of the sparsifying precoder as a mixed-integer linear program (MILP), which can be efficiently solved using standard off-the-shelf software.

We note that the material of this chapter is based on a number of publications by the authors [22, 29, 30].

**Notation**

We denote scalars, vectors, matrices, and sets by lower case letters, lower case bold letters, upper case bold letters, and calligraphic letters, i.e., $x$, $\mathbf{x}$, $\mathbf{X}$, $\mathcal{X}$, respectively. We refer to the $i$-th element of a vector $\mathbf{x}$ by $[\mathbf{x}]_i$, and to the $(i, j)$-th element of a matrix $\mathbf{X}$ by $[\mathbf{X}]_{i,j}$. For a non-negative integer $M$, we define the set $[M] = \{0, 1, \ldots, M-1\}$. Superscripts $(\cdot)^*$, $(\cdot)^{\mathsf{T}}$, $(\cdot)^{\mathsf{H}}$, $(\cdot)^{-1}$, and $(\cdot)^{\dagger}$ represent the complex conjugate, transpose, conjugate transpose, inverse, and Moore–Penrose pseudoinverse, respectively. For a vector $\mathbf{x}$, the symbol $\mathrm{diag}(\mathbf{x})$ denotes a matrix with $\mathbf{x}$ as its main diagonal. The $\ell_p$-norm of a vector $\mathbf{x}$ is referred to as $\|\mathbf{x}\|_p$, where for simplicity we drop the subscript for the case of $p = 2$. The Frobenius norm of a matrix $\mathbf{X}$ is denoted by $\|\mathbf{X}\|_{\mathsf{F}}$. We denote a bipartite graph as, for example, $\mathcal{G} = (\mathcal{V}_1, \mathcal{V}_2, \mathcal{E})$, where $\mathcal{V}_1$ and $\mathcal{V}_2$ are the two color classes and $\mathcal{E}$ is the edge set. For a vertex $x$, the degree of $x$ refers to the number of edges in $\mathcal{E}$ incident on $x$ and is denoted by $\deg_{\mathcal{G}}(x)$. The neighbors of $x$, $\mathcal{N}_{\mathcal{G}}(x)$, are those vertices that are connected to $x$.

## 11.2 System Model

Consider a BS equipped with a uniform linear array (ULA) with $M$ antennas, operating in FDD mode: UL transmission from a user to the BS takes place over a frequency band

$$\mathcal{F}_{\mathrm{ul}} = [f_{\mathrm{ul}} - \frac{\mathrm{BW}_{\mathrm{ul}}}{2}, f_{\mathrm{ul}} + \frac{\mathrm{BW}_{\mathrm{ul}}}{2}]$$

with carrier frequency $f_{\mathrm{ul}}$ and a bandwidth $\mathrm{BW}_{\mathrm{ul}}$, and downlink (DL) transmission from the BS to the user takes place over a band

$$\mathcal{F}_{\mathrm{dl}} = [f_{\mathrm{dl}} - \frac{\mathrm{BW}_{\mathrm{dl}}}{2}, f_{\mathrm{dl}} + \frac{\mathrm{BW}_{\mathrm{dl}}}{2}]$$

with carrier frequency $f_{\mathrm{dl}}$ and a bandwidth $\mathrm{BW}_{\mathrm{dl}}$. For example, one mode of operation in the 3GPP standard uses the $\mathcal{F}_{\mathrm{ul}} = [1920, 1980]$ MHz band for UL and the $\mathcal{F}_{\mathrm{dl}} = [2110, 2170]$ MHz band for DL transmission, so that $f_{\mathrm{ul}} = 1950$ MHz, $f_{\mathrm{dl}} = 2140$ MHz, and $\mathrm{BW}_{\mathrm{ul}} = \mathrm{BW}_{\mathrm{dl}} = 60$ MHz [53]. The gap between UL and DL bands is larger than the channel coherence bandwidth. For example, the gap between UL and DL bands in the example above is equal to 190 MHz, while the coherence bandwidth in a macrocell is in the order of 1 MHz [48]. Therefore, for an FDD system, *channel reciprocity* does not hold, which means that UL and DL

instantaneous channels are not the same. As a result, to transmit data, the BS has to first obtain the DL CSI. This is done by sending a number of $T_{\mathrm{dl}}$ pilot symbols to the user and receiving measurement feedback from the user, which enables the BS to estimate the DL CSI and design the beamformer. If there exist multiple users, this is done simultaneously for each one of them, and the beamformer is designed depending on all estimated channels, using one of the various methods such as zero-forcing beamforming [59].

In a massive MIMO system, this proves to be a challenge, due to the high channel dimension ($M \gg 1$). In order to train $M$ antennas, a conventional scheme requires a minimum pilot dimension of $T_{\mathrm{dl}} = M$. Hence, with such a scheme, the number of BS antennas cannot be made arbitrarily large, since the pilot dimension is limited to the dimension of the time–frequency coherence block. For example, consider a typical scenario in LTE, where groups of users are scheduled over resource blocks spanning 14 OFDM symbols $\times$ 12 subcarriers, for a total dimension of $T = 168$ symbols in the time–frequency plane [53], and a typical massive MIMO configuration serving $K \sim 20$ users with $M \geq 200$ antennas (see, e.g., [37]). In this case, since $M \geq T$, the entire resource block dimension would be consumed by the DL pilot, leaving no room for data communication. Designing a massive MIMO system that operates in FDD mode requires developing methods that overcome these dimensionality issues.

### *11.2.1 Related Work*

Several works have proposed to reduce both the DL training and UL feedback overheads by exploiting the sparse structure of the massive MIMO channel. In particular, these works assume that propagation between the BS array and the user antenna occurs through a limited number of scattering clusters, with limited support in the angle-of-arrival/angle-of-departure (AoA–AoD) domain.[4] A common model is to represent the channel as a superposition of the array response to the electromagnetic wave incoming from a small number of AoAs ($p \ll M$), i.e.,

$$\mathbf{h} = \sum_{i=1}^{p} c_p \mathbf{a}(\theta_p) \in \mathbb{C}^M,$$

where $c_p$ are complex coefficients and $\mathbf{a}(\theta_p) \in \mathbb{C}^M$ is the vector containing array element responses to a wave coming from the AoA $\theta_p$. Hence, by decomposing the angle domain into discrete "virtual beam" directions, the $M$-dimensional channel $\mathbf{h}$ admits a sparse representation in the beam domain [3, 52]. Building on this idea,

[4] From the BS perspective, AoD for the DL and AoA for the UL indicate the same domain. Hence, we shall simply refer to this as the "angle domain," while the meaning of departure (DL) or arrival (UL) is clear from the context.

a large number of works (see, e.g., [11, 15, 19, 47, 54, 57]) have proposed to use "compressed pilots," i.e., a reduced DL pilot dimension $T_{\mathrm{dl}} < M$, in order to estimate the channel vectors using compressed sensing (CS) techniques [8, 16]. For example, in [3], the sparse representation of channel multipath components in angle, delay, and Doppler domains was exploited to propose CS methods for channel estimation using far fewer measurements than required by conventional least-square (LS) methods. In [19], the authors note that the angles of the multipath channel components are common among all the subcarriers in the OFDM signaling. Then they propose to exploit the common sparsity pattern of the channel coefficients to further reduce the number of required pilot measurements. This gives rise to a so-called multiple measurement vector (MMV) setup, which is typically applied when multiple snapshots of a random vector with common sparse support can be acquired and jointly processed [10, 17]. This was adapted to FDD in the massive MIMO regime, where the frequent idea is to probe the channel using compressed DL pilots, receiving the measurements at the BS via feedback and performing channel estimation there. A recent work based on this approach was presented in [47], starting with the observation that, as shown in many experimental studies [18, 24, 28, 33], the propagation between the BS antenna array and the users occurs along scattering clusters that may be common to multiple users, since they all belong to the same scattering environment. In turn, this yields that the channel sparse representations (in the angle/beam domain) share a common part of their support. Then, [47] considers a scheme where the users feed back their noisy DL pilot measurements to the BS, and the latter runs a joint recovery algorithm, coined as joint orthogonal matching pursuit (J-OMP), able to take advantage of the common sparsity. It follows that in the presence of common sparsity, J-OMP improves upon the basic CS schemes that estimate each user channel separately.

More recent CS-based methods, in addition, make use of the *angular reciprocity* between the UL and DL channels in FDD systems to improve channel estimation. Namely, this refers to the fact that the directions (angles) of propagation for the UL and DL channel are invariant over the frequency range spanning the UL and DL bands, which is generally very small with respect to the carrier frequency (e.g., UL/DL separation of the order of 100 MHz, for carrier frequencies ranging between 2 and 6 GHz) [2, 25, 58]. In [57], the sparse set of AoAs is estimated from a preamble transmission phase in the UL, and this information is used for user grouping and channel estimation in the DL according to the well-known *joint spatial division and multiplexing* (JSDM) paradigm [1, 44]. In [15], the authors proposed a dictionary learning-based approach for training DL channels. First, in a preliminary learning phase, the BS "learns" a pair of UL–DL dictionaries that are able to sparsely represent the channel. Then, these dictionaries are used for a joint sparse estimation of instantaneous UL–DL channels. An issue with this method is that the dictionary learning phase requires off-line training and must be re-run if the propagation environment around the BS changes (e.g., due to large moving objects such as truck and buses, or a new building). In addition, the computation involved in the instantaneous channel estimation is prohibitively demanding for real-time operations with a large number of antennas ($M > 100$). In [11], the authors propose estimating the DL channel using a sparse Bayesian learning framework,

aiming at joint-maximum a posteriori (MAP) estimation of the off-grid AoAs and multipath power coefficients by observing instantaneous UL channel measurements. This method has the drawback that it fundamentally assumes discrete and separable (in the AoA domain) multipath components and that the number of signal paths (the number of channel AoAs) is known a priori. Hence, the method simply cannot be applied in scenarios with diffuse (continuous) scattering, where the scattering power is distributed over an interval of the angular domain with non-negligible width. Such scattering is observed and modeled for various types of communication channels, and they do not necessarily admit a sparse angular representation [45, 49, 50, 55].

### *11.2.2 Contribution*

The main problem shared among all the methods above is that they are not robust with respect to the channel sparsity assumption and, as we will show in our simulation results, can lead to poor channel estimates when this assumption is violated. Throughout this chapter, we develop the idea of active channel sparsification (ACS), which aims at DL data multiplexing with any pilot dimension that is available at the BS for channel training. This is done by designing a *joint precoder* that projects the user channels onto a lower-dimensional subspace such that the sparsity order (the $\ell_0$ pseudo-norm) of each projected channel is less than the pilot dimension. We argue that such a projection comprises a necessary step for interference-free DL data transmission. Among all precoders satisfying this condition, we select one that, in a certain probabilistic sense, maximizes the projected channel matrix rank. As is well-known, the channel matrix rank is equivalent to the channel multiplexing gain, i.e., the number of independent data streams that can be simultaneously multiplexed on the communication link [56]. Then the BS estimates the projected channel matrix (and not the full-dimensional channel matrix), and it communicates with the users through it. Figure 11.1 graphically sketches the idea. The projection enables stable estimation of the "effective channels," and its rank maximization capability results in maximization of the multiplexing gain and (implicitly) the DL sum rate. We emphasize that our scheme does *not* rely on channel sparsity and in fact is specifically designed to induce it in a clever way despite the channels being of arbitrary sparsity orders. In this sense, it is a solution to the FDD massive MIMO problem in its generality.

## 11.3 Channel Model

Consider a BS equipped with a uniform linear array (ULA) of $M$ antennas, serving $K$ users in a cell.[5] The channel of user $k$ is assumed to be a zero-mean, complex Gaussian vector $\mathbf{h}_k \in \mathbb{C}^M$ with covariance $\mathbf{\Sigma}_k = \mathbb{E}\left[\mathbf{h}_k \mathbf{h}_k^{\mathsf{H}}\right]$. There are a number

[5] An extension of the idea to general arrays will follow later in this chapter.

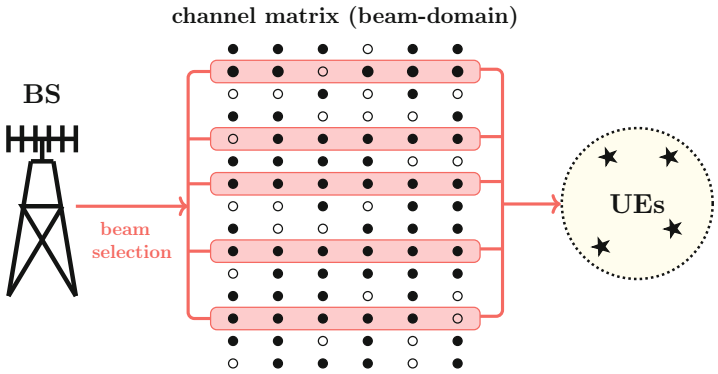



**Fig. 11.1** Schematic of the idea of active channel sparsification. The dotted shape in the middle represents the beam-domain channel matrix, in which the columns associate with user channels and rows with (virtual) beams, and filled dots represent non-zero channel coefficients. The precoder is designed to select a set of beams (and users) over which the BS transmits data in the DL

of ways to obtain the DL channel covariance of a user in MIMO FDD systems [11, 13, 43, 51], including one proposed by some of the authors of this chapter [22]. For example, the BS can first estimate the UL covariance from UL pilots that are naturally received from the users. Then it can estimate the DL covariance by "transforming" the UL covariance. We do not discuss the details of DL covariance estimation, and in order to isolate the problem of channel training from the effects of an covariance estimation, here we assume that the covariances are known to the BS.

Denoting the channel of a generic user with $\mathbf{h} \in \mathbb{C}^M$, it can be expressed as

$$\mathbf{h} = \int_{-\theta_{\max}}^{\theta_{\max}} dW(\theta)\mathbf{a}(\theta), \tag{11.1}$$

where $\theta \in [-\theta_{\max}, \theta_{\max}]$ stands for the AoA, $\theta_{\max}$ is the maximum array angular aperture (e.g., $\theta_{\max} = \pi/3$), $W$ is a zero-mean, complex Gaussian stochastic process that represents the random angular gains, and thereby the right-hand side (RHS) of (11.1) is understood as an stochastic integral [31]. In addition, $\mathbf{a}(\theta) \in \mathbb{C}^M$ is the *far-field* array response to a wave impinging from the AoA $\theta$, whose $m$-th element is given as

$$[\mathbf{a}(\theta)]_m = \exp\left(j\frac{2\pi d}{\lambda} m \sin\theta\right), \tag{11.2}$$

where $\lambda$ is the carrier wavelength and $d$ denotes the antenna spacing. We consider the latter to be taking the standard value $d = \frac{\lambda}{2\sin\theta_{\max}}$. To simplify notation, we can introduce the *normalized* AoA $\xi$ by making the change of variables $\xi = \sin\theta/\sin\theta_{\max}$. This results in the element response (11.2) to turn into $[\mathbf{a}(\xi)]_m =$

$\exp(j\pi m\xi)$ for $m \in [M]$, and the channel vector in (11.1) to be expressed as

$$\mathbf{h} = \int_{-1}^{1} dW(\xi)\mathbf{a}(\xi). \tag{11.3}$$

Assuming uncorrelated scattering, the autocorrelation of $W$ is given as

$$\mathbb{E}[dW(\xi)dW(\xi')] = \delta(\xi - \xi')\gamma(\xi)d\xi, \tag{11.4}$$

in which $\gamma$ is non-negative and known as the angular scattering function (ASF). The channel covariance is then given as

$$\mathbf{\Sigma} = \mathbb{E}[\mathbf{h}\mathbf{h}^{\mathsf{H}}] = \int_{-1}^{1} \gamma(\xi)\mathbf{a}(\xi)\mathbf{a}(\xi)^{\mathsf{H}}d\xi. \tag{11.5}$$

It is easy to show that for a ULA, the channel covariance is Toeplitz Hermitian. This results in a nice property that will be outlined below.

**Approximate Common Eigenspace of ULA Channels**
Let $\gamma$ be a function over $[-1, 1]$ bounded to $[\gamma_{\min}, \gamma_{\max}]$ with $0 \leq \gamma_{\min} \leq \gamma_{\max}$, whose Fourier transform samples over the integer set $[M]$ are equal to the sequence $[\boldsymbol{\sigma}]_m = [\mathbf{\Sigma}]_{n,n-m}$, i.e.,

$$[\boldsymbol{\sigma}]_m = \int_{-1}^{1} \gamma(\xi)e^{j\pi m\xi}d\xi. \tag{11.6}$$

According to the Szegö theorem, the Toeplitz covariance $\mathbf{\Sigma}$ can be approximated by a circulant matrix $\mathring{\mathbf{\Sigma}}$ whose first column is given as [20]

$$[\mathring{\boldsymbol{\sigma}}]_m = \begin{cases} [\boldsymbol{\sigma}]_0, & m = 0, \\ [\boldsymbol{\sigma}]_m + [\boldsymbol{\sigma}]_{m-M}, & m = 1, \ldots, M-1, \end{cases} \tag{11.7}$$

where for a negative index $i < 0$, we define $[\boldsymbol{\sigma}]_i = [\boldsymbol{\sigma}]^*_{-i}$. The approximation of the Toeplitz matrix with the circulant matrix is understood in the following senses [1]:

1. The set of eigenvalues of $\mathbf{\Sigma}$ and $\mathring{\mathbf{\Sigma}}$ denoted as $\{\lambda_m\}$ and $\{\mathring{\lambda}_m\}$ are asymptotically equally distributed. This means that for any continuous function $f$ defined over $[\gamma_{\min}, \gamma_{\max}]$, we have

$$\lim_{M\to\infty} \frac{1}{M} \sum_{m=0}^{M-1} f(\lambda_m) = \lim_{M\to\infty} \frac{1}{M} \sum_{m=0}^{M-1} f(\mathring{\lambda}_m). \tag{11.8}$$

2. The eigenvectors of $\mathbf{\Sigma}$ are approximated by those of $\mathring{\mathbf{\Sigma}}$ in the following sense. Define the asymptotic cumulative distribution function (CDF) of the eigenvalues of $\mathbf{\Sigma}$ as $F(\lambda) = \int_{-1}^{1} \mathbf{1}_{\{\gamma(\xi)\leq\lambda\}} d\xi$. Define $\mathbf{U} = [\mathbf{u}_0, \ldots, \mathbf{u}_{M-1}]$ and $\mathring{\mathbf{U}} = [\mathring{\mathbf{u}}_0, \ldots, \mathring{\mathbf{u}}_{M-1}]$ as the eigenvector matrices of $\mathbf{\Sigma}$ and $\mathring{\mathbf{\Sigma}}$ corresponding to the descendingly ordered eigenvalues $\{\lambda_m\}$ and $\{\mathring{\lambda}_m\}$, respectively. For any interval $[a, b] \subseteq [\gamma_{\min}, \gamma_{\max}]$ such that $F$ is continuous on $[a, b]$, define two sets of indices as $\mathcal{I}_{[a,b]} = \{m : \lambda_m \in [a, b]\}$ and $\mathring{\mathcal{I}}_{[a,b]} = \{m : \mathring{\lambda}_m \in [a, b]\}$, corresponding to those eigenvalues that lie in $[a, b]$. Also define the column-wise submatrices of $\mathbf{U}$ and $\mathring{\mathbf{U}}$ associated with these indices as $\mathbf{U}_{[a,b]} = \left(\mathbf{u}_m : m \in \mathcal{I}_{[a,b]}\right)$ and $\mathring{\mathbf{U}}_{[a,b]} = \left(\mathring{\mathbf{u}}_m : m \in \mathring{\mathcal{I}}_{[a,b]}\right)$. Then

$$\lim_{M\to\infty} \|\mathbf{U}_{[a,b]}\mathbf{U}_{[a,b]}^{\mathsf{H}} - \mathring{\mathbf{U}}_{[a,b]}\mathring{\mathbf{U}}_{[a,b]}^{\mathsf{H}}\|_{\mathsf{F}}^2 = 0. \tag{11.9}$$

Besides the points above, we know that the eigenvectors of a circulant matrix are given by the DFT basis $\mathbf{F} \in \mathbb{C}^{M\times M}$ of the same dimension, whose $(m, n)$-th entry is given by $[\mathbf{F}]_{m,n} = \frac{1}{\sqrt{M}} e^{-j2\pi \frac{mn}{M}}$, $m, n \in [M]$. This highly simplifies the precoder design, since now the user channels can all be (asymptotically) expressed in a shared eigenspace that is given by the DFT columns.

For a user $k$, let $\mathbf{\Sigma}_k = \mathbf{U}_k \mathbf{\Lambda}_k \mathbf{U}_k^{\mathsf{H}}$ be its channel covariance eigendecomposition and $\mathbf{\Lambda}_k = \mathrm{diag}(\boldsymbol{\lambda}_k)$ its diagonal matrix of ordered eigenvalues and $\mathbf{U}_k$ its unitary matrix of eigenvectors. The Karhunen–Loéve (KL) expansion of a random channel realization is given by Grimmett et al. [21]

$$\mathbf{h}_k = \mathbf{U}_k \mathbf{g}_k', \tag{11.10}$$

where $\mathbf{g}_k' \sim \mathcal{CN}(\mathbf{0}, \mathbf{\Lambda}_k)$ is a complex Gaussian-distributed vector. As a consequence of Szegö's theorem, we have that asymptotically as $M \to \infty$, this channel realization is equal to

$$\mathbf{h}_k = \mathbf{F}\, \mathbf{g}_k, \tag{11.11}$$

where $\mathbf{g}_k \sim \mathcal{CN}(\mathbf{0}, \widetilde{\mathbf{\Lambda}}_k)$ and where $\widetilde{\mathbf{\Lambda}}_k = \mathrm{diag}(\mathbf{\Pi}_k \boldsymbol{\lambda}_k)$ for some permutation matrix $\mathbf{\Pi}_k \in \mathbb{C}^{M\times M}$. In this decomposition, the columns of $\mathbf{F}$ give the eigenvectors (KL basis vectors), and unlike the eigenvectors matrix $\mathbf{U}_k$ in (11.10), they do *not* depend on the user index. The product $\mathbf{\Pi}_k \boldsymbol{\lambda}_k$ simply reorders the eigenvalues in $\boldsymbol{\lambda}_k$ so that they match with their eigenvectors ordered as in $\mathbf{F}$. For any user $k$, the exact eigenvectors converge to the DFT columns $\mathbf{f}_0, \ldots, \mathbf{f}_{M-1}$, that is to say in the sense of *Point 2* above, we have

$$\{\mathbf{u}_0^{(k)}, \ldots, \mathbf{u}_{M-1}^{(k)}\} \to \{\mathbf{f}_0, \ldots, \mathbf{f}_{M-1}\},$$

for all $k \in [K]$. Later we exploit this property in designing the sparsifying precoder.

Furthermore, note that the columns of $\mathbf{F}$ are similar to array response vectors, and in fact, each column with index $m \in [M]$ of the DFT matrix can be seen as the array response to an angular direction $\theta = \sin^{-1}(\frac{m}{M}\sin\theta_{\max})$, where $[\boldsymbol{\lambda}_k]_m$ can be seen as the power of the channel vector associated with user $k$ along that direction. Due to the presumably limited number of local scatterers as seen at the BS and the large number of antennas of the array, one can *hypothesize* that only a few entries of $\boldsymbol{\lambda}_k$ are significantly large, implying that the DL channel vector $\mathbf{h}_k$ is sparse in the Fourier basis. This sparsity in the beam-space domain is precisely what has been exploited in the CS-based works discussed in Sect. 11.2.1, in order to reduce the DL pilot dimension $T_{\mathrm{dl}}$. As seen in the next section, our proposed approach does *not* rely on any intrinsic channel sparsity assumption but adopts a novel artificial sparsification technique that smartly reduces the effective channel dimension to enable channel estimation regardless of its sparsity.

## 11.4 Active Channel Sparsification and DL Channel Training

In order to perform multiuser communication, the BS needs to train *instantaneous* DL channels of the users. This is done by transmitting a pilot matrix $\boldsymbol{\Phi} \in \mathbb{C}^{T_{\mathrm{dl}} \times M}$, where $T_{\mathrm{dl}}$ represents the pilot dimension. To obtain a *good* configuration for $\boldsymbol{\Phi}$, we first decompose it as the product

$$\boldsymbol{\Phi} = \boldsymbol{\Psi}\mathbf{B}, \tag{11.12}$$

where $\mathbf{B} \in \mathbb{C}^{M' \times M}$ is the to-be-designed sparsifying precoder with $M' \leq M$ being an intermediate dimension that will be determined within the precoder design, and $\boldsymbol{\Psi} \in \mathbb{C}^{T_{\mathrm{dl}} \times M'}$ is a matrix that can be chosen from a random ensemble, e.g., i.i.d. Gaussian or random unitary. Note that the design of $\mathbf{B}$ and $\boldsymbol{\Psi}$ does not depend on instantaneous channel realizations, which in fact must be estimated via the closed-loop DL probing and channel-state feedback mechanism.

The precoded DL training length (in time–frequency symbols) spans $T_{\mathrm{dl}}$ dimensions, and the DL training phase is repeated at each fading block of dimension $T$. Collecting the $T_{\mathrm{dl}}$ training symbols in a column vector, the corresponding observation at user $k$ receiver is given by

$$\begin{aligned} \mathbf{y}_k &= \boldsymbol{\Phi}\mathbf{h}_k + \mathbf{z}_k \\ &= \boldsymbol{\Psi}\mathbf{B}\mathbf{h}_k + \mathbf{z}_k = \boldsymbol{\Psi}\widetilde{\mathbf{h}}_k + \mathbf{z}_k, \end{aligned} \tag{11.13}$$

where $\mathbf{B}$ is the precoding matrix, $\mathbf{h}_k$ is the channel vector of user $k$, and we have defined $\widetilde{\mathbf{h}}_k := \mathbf{B}\mathbf{h}_k$ as the *effective channel* vector, formed by the concatenation of the actual DL channel (antenna-to-antenna) with the precoder $\mathbf{B}$. We consider additive white Gaussian noise (AWGN) with distribution $\mathbf{z}_k \sim \mathcal{CN}(\mathbf{0}, N_0\mathbf{I}_{T_{\mathrm{dl}}})$. The training and precoding matrices are normalized such that

$$\mathrm{tr}(\mathbf{\Psi B B}^{\mathsf{H}}\mathbf{\Psi}^{\mathsf{H}}) = T_{\mathrm{dl}} P_{\mathrm{dl}}, \tag{11.14}$$

where $P_{\mathrm{dl}}$ denotes the total BS transmit power, and we define the DL signal-to-noise ratio as SNR$= P_{\mathrm{dl}}/N_0$.

Most works on channel estimation focus on the estimation of the actual channels $\{\mathbf{h}_k\}$. This is recovered in our setting by letting $\mathbf{B} = \mathbf{I}_M$. However, our goal here is to design the sparsifying precoder such that each effective channel $\widetilde{\mathbf{h}}_k$ becomes sparse enough to be "stably" estimated from the measurements taken $\mathbf{\Psi}$, and yet the collection of all effective channels forms a matrix $\widetilde{\mathbf{H}} = [\widetilde{\mathbf{h}}_0, \ldots, \widetilde{\mathbf{h}}_{K-1}]$ that has a high rank. In this way, each user's channel can be partly estimated using a small pilot overhead $T_{\mathrm{dl}}$, but the BS is still able to serve many data streams using spatial multiplexing in the DL (in fact, as many as the rank of the effective channel matrix).

### *11.4.1 Necessity and Implication of Stable Channel Estimation*

Suppose that the channel representation (11.11) holds exactly and that the eigenvalue vectors $\boldsymbol{\lambda}_k$, $k \in [K]$ have support $\mathcal{S}_k = \{m : [\boldsymbol{\lambda}_k]_m \neq 0\}$ with sparsity order $s_k = |\mathcal{S}_k|$. We hasten to point out that the above are convenient *design assumptions*, made in order to obtain a tractable problem, and that the eventual precoder is applied to the actual physical channels.

**Definition 11.1 (Stable Estimation)** Consider the noisy, linear measurement model $\mathbf{y} = \mathbf{\Phi h} + \mathbf{z}$ as introduced in (11.13), where $\mathbf{z} \sim \mathcal{CN}(\mathbf{0}, N_0\mathbf{I})$. We say that an estimator $\widehat{\mathbf{h}}$ of $\mathbf{h}$ given $\mathbf{y}$ is stable if $\lim_{N_0 \downarrow 0} \mathbb{E}\left[\|\mathbf{h} - \widehat{\mathbf{h}}\|^2\right] = 0$.[6]

The following lemma yields necessary and sufficient conditions for stable estimation of the channel vectors $\mathbf{h}_k$.

**Lemma 11.1** *Consider the Gaussian vector* $\mathbf{h}_k$ *described via* (11.11) *and with support set* $\mathcal{S}_k$. *Let* $\widehat{\mathbf{h}}_k$ *denote any estimator of* $\mathbf{h}_k$ *given the observation*[7] $\mathbf{y}_k = \mathbf{\Psi h}_k + \mathbf{z}_k$. *If* $T_{dl} \geq s_k$, *there exist pilot matrices* $\mathbf{\Psi} \in \mathbb{C}^{T_{dl} \times M}$ *for which* $\lim_{N_0 \downarrow 0} \mathbb{E}\left[\|\mathbf{h} - \widehat{\mathbf{h}}\|^2\right] = 0$ *for all support sets* $\mathcal{S}_k$ *with* $|\mathcal{S}_k| = s_k$. *Conversely, for any support set* $\mathcal{S}_k : |\mathcal{S}_k| = s_k$, *any pilot matrix* $\mathbf{\Psi} \in \mathbb{C}^{T_{dl} \times M}$ *with* $T_{dl} < s_k$ *yields* $\lim_{N_0 \downarrow 0} \mathbb{E}\left[\|\mathbf{h} - \widehat{\mathbf{h}}\|^2\right] > 0$.

***Proof*** The proof follows by using the KL representation $\mathbf{h}_k = \sum_{m \in \mathcal{S}_k} g_{k,m} \sqrt{[\boldsymbol{\lambda}_k]_m} \mathbf{f}_m$, which holds exactly by assumption. Estimating $\mathbf{h}_k$ is equivalent to estimating the vector of KL Gaussian i.i.d. coefficients $\mathbf{g}_k = (g_{k,m} : m \in \mathcal{S}_k) \in \mathbb{C}^{s_k \times 1}$. Define the $M \times s_k$ DFT submatrix $\mathbf{F}_{\mathcal{S}_k} = (\mathbf{f}_m : m \in \mathcal{S}_k)$, and the corresponding diagonal $s_k \times s_k$ matrix of the non-zero eigenvalues $[\mathbf{\Lambda}_k]_{\mathcal{S}_k, \mathcal{S}_k}$. After some simple standard algebra, the MMSE estimation error covariance of $\mathbf{g}_k$ from $\mathbf{y}_k$ in (11.13) with $\mathbf{B} = \mathbf{I}_M$ can

[6] By $N_0 \downarrow 0$, we mean that $N_0$ is approaching 0 from above.

[7] Note that this coincides with (11.13) with $\mathbf{B} = \mathbf{I}_M$, i.e., without the sparsifying precoder.

be written in the form

$$\begin{aligned}\widetilde{\mathbf{R}}_e = \mathbf{I}_{s_k} - \left([\boldsymbol{\Lambda}_k]_{\mathcal{S}_k,\mathcal{S}_k}\right)^{1/2} \mathbf{F}_{\mathcal{S}_k}^{\mathsf{H}} \boldsymbol{\Psi}^{\mathsf{H}} \\ \times \left(\boldsymbol{\Psi}\mathbf{F}_{\mathcal{S}_k}[\boldsymbol{\Lambda}_k]_{\mathcal{S}_k,\mathcal{S}_k}\mathbf{F}_{\mathcal{S}_k}^{\mathsf{H}}\boldsymbol{\Psi}^{\mathsf{H}} + N_0\mathbf{I}_{T_{\mathrm{dl}}}\right)^{-1} \boldsymbol{\Psi}\mathbf{F}_{\mathcal{S}_k}\left([\boldsymbol{\Lambda}_k]_{\mathcal{S}_k,\mathcal{S}_k}\right)^{1/2}.\end{aligned} \tag{11.15}$$

Using the fact that $\mathbf{R}_e = \mathbf{F}_{\mathcal{S}_k}([\boldsymbol{\Lambda}_k]_{\mathcal{S}_k,\mathcal{S}_k})^{1/2}\widetilde{\mathbf{R}}_e([\boldsymbol{\Lambda}_k]_{\mathcal{S}_k,\mathcal{S}_k})^{1/2}\mathbf{F}_{\mathcal{S}_k}^{\mathsf{H}}$, such that $\mathrm{tr}(\mathbf{R}_e) = \mathrm{tr}([\boldsymbol{\Lambda}_k]_{\mathcal{S}_k,\mathcal{S}_k}\widetilde{\mathbf{R}}_e)$, we have that $\mathrm{tr}(\mathbf{R}_e)$ and $\mathrm{tr}(\widetilde{\mathbf{R}}_e)$ have the same vanishing order with respect to $N_0$. In particular, it is sufficient to consider the behavior of $\mathrm{tr}(\widetilde{\mathbf{R}}_e)$ as a function of $N_0$. Now, using the Sherman–Morrison–Woodbury matrix inversion lemma [23], after some algebra omitted for the sake of brevity, we arrive at

$$\mathrm{tr}(\widetilde{\mathbf{R}}_e) = s_k - \sum_{i=1}^{s_k} \frac{\mu_i}{N_0 + \mu_i}, \tag{11.16}$$

where $\mu_i$ is the $i$-th eigenvalue of the $s_k \times s_k$ matrix

$$\mathbf{A} = ([\boldsymbol{\Lambda}_k]_{\mathcal{S}_k,\mathcal{S}_k})^{1/2}\mathbf{F}_{\mathcal{S}_k}^{\mathsf{H}}\boldsymbol{\Psi}^{\mathsf{H}}\boldsymbol{\Psi}\mathbf{F}_{\mathcal{S}_k}([\boldsymbol{\Lambda}_k]_{\mathcal{S}_k,\mathcal{S}_k})^{1/2}.$$

Next, notice that

$$\mathrm{rank}(\mathbf{A}) = \mathrm{rank}(\mathbf{F}_{\mathcal{S}_k}^{\mathsf{H}}\boldsymbol{\Psi}^{\mathsf{H}}\boldsymbol{\Psi}\mathbf{F}_{\mathcal{S}_k}) = \mathrm{rank}(\mathbf{F}_{\mathcal{S}_k}\mathbf{F}_{\mathcal{S}_k}^{\mathsf{H}}\boldsymbol{\Psi}^{\mathsf{H}}) \le \min\{s_k, T_{\mathrm{dl}}\}. \tag{11.17}$$

In fact, $[\boldsymbol{\Lambda}_k]_{\mathcal{S}_k,\mathcal{S}_k}$ is diagonal with strictly positive diagonal elements, such that left and right multiplication by $([\boldsymbol{\Lambda}_k]_{\mathcal{S}_k,\mathcal{S}_k})^{1/2}$ yields rank-preserving row and column scalings, the matrix $\mathbf{F}_{\mathcal{S}_k}\mathbf{F}_{\mathcal{S}_k}^{\mathsf{H}}$ is the orthogonal projector onto the $s_k$-dimensional column space of $\mathbf{F}_{\mathcal{S}_k}$ and has rank $s_k$, while the matrix $\boldsymbol{\Psi}^{\mathsf{H}} \in \mathbb{C}^{M\times T_{\mathrm{dl}}}$ has the same rank of $\boldsymbol{\Psi}^{\mathsf{H}}\boldsymbol{\Psi}$, that is at most $T_{\mathrm{dl}}$.

For $T_{\mathrm{dl}} \ge s_k$, the existence of matrices $\boldsymbol{\Psi}$ such that the rank upper bound (11.17) holds with equality (i.e., for which $\mathrm{rank}(A) = s_k$ for any support set $\mathcal{S}_k$ of size $s_k$) is shown as follows. Generate a random $\boldsymbol{\Psi}$ with i.i.d. elements $\sim \mathcal{CN}(0, 1)$. Then, the columns of $\mathbf{F}_{\mathcal{S}_k}^{\mathsf{H}}\boldsymbol{\Psi}^{\mathsf{H}}$ form a collection of $T_{\mathrm{dl}} \ge s_k$ mutually independent $s_k$-dimensional Gaussian vectors with i.i.d. $\sim \mathcal{CN}(0, 1)$ components. The event that these vectors span a space of dimension less than $s_k$ is a null event (zero probability). Hence, such randomly generated matrix satisfies the rank equality in (11.17) with probability 1. As a consequence, for $T_{\mathrm{dl}} \ge s_k$, we have that $\mu_i > 0$ for all $i \in [s_k]$, and (11.16) vanishes as $O(N_0)$ as $N_0 \downarrow 0$. In contrast, if $T_{\mathrm{dl}} < s_k$, by (11.17) for any matrix $\boldsymbol{\Psi}$ at most $T_{\mathrm{dl}}$, eigenvalues $\mu_i$ in (11.16) are non-zero and $\lim_{N_0\downarrow 0} s_k - \sum_{i=1}^{s_k} \frac{\mu_i}{N_0+\mu_i} \ge s_k - T_{\mathrm{dl}} > 0$. □

As a direct consequence of Lemma 11.1, we have that any scheme relying on intrinsic channel sparsity cannot yield stable estimation if $T_{\mathrm{dl}} < s_k$ for some user

$k \in [K]$. Furthermore, we need to impose that the sparsity order of the projected channels $\mathbf{B}\mathbf{h}_k$, $k \in [K]$ is less than or equal to the desired pilot dimension $T_{\mathrm{dl}}$.

It is important to note that the requirement of estimation stability is *essential* in order to achieve high spectral efficiency in high SNR conditions, irrespective of the DL precoding scheme. In fact, if the estimation mean squared error (MSE) of the user channels does not vanish as $N_0 \downarrow 0$, the system self-interference due to the imperfect channel knowledge grows proportionally to the signal power, yielding a signal-to-interference plus noise ratio (SINR) that saturates to a constant when SNR becomes large. Hence, for sufficiently high SNR, the best strategy would consist of transmitting just a single data stream, since any form of multiuser precoding would inevitably lead to an interference-limited regime, where the sum rate remains bounded, while SNR $\to \infty$ [12]. Conversely, it is also well-known that when the channel estimation error vanishes as $O(N_0)$ for $N_0 \downarrow 0$, the high-SNR sum rate behaves as if the channel was perfectly known and can be achieved by very simple linear precoding [7]. A possible solution to this problem consists of serving only the users whose channel support $s_k$ is not larger than $T_{\mathrm{dl}}$. This is assumed *implicitly* in all CS-based schemes and represents a major intrinsic limitation of the CS-based approaches. In contrast, by artificially sparsifying the user channels, we manage to serve all users given a fixed DL pilot dimension $T_{\mathrm{dl}}$.

### *11.4.2 Sparsifying Precoder Design*

We now introduce a graphical model that encodes the *power profile* of each user along the common virtual beams, namely along the DFT columns. Define $\mathcal{G} = (\mathcal{V}, \mathcal{K}, \mathcal{E})$ as a bipartite graph with two color classes $\mathcal{V}$ and $\mathcal{K}$, where $\mathcal{V}$ is a node set of cardinality $M$, representing the set of virtual beams and $\mathcal{K}$ is a node set of cardinality $K$, representing the users. Also $(k, v) \in \mathcal{E}$ if and only if $[\boldsymbol{\lambda}_k]_v > \delta_0$, where $\delta_0 > 0$ is a small threshold that ensures that when the link is very weak, it does not appear in the graph. Therefore, the *biadjacency matrix* of this graph is given by an $M \times K$ binary matrix $\mathbf{A}$ for which $[\mathbf{A}]_{v,k} = 1$ if and only if $(v, k) \in \mathcal{E}$. The *weighted biadjacency matrix* is defined as the $M \times K$ matrix $\mathbf{W}$ in which $[\mathbf{W}]_{m,k} = [\boldsymbol{\lambda}_k]_m^{1/2}$.

From (11.11), the DL channel matrix $\mathbf{H} = [\mathbf{h}_0, \ldots, \mathbf{h}_{K-1}] \in \mathbb{C}^{M \times K}$ is related to the matrix of angular channel gains $\mathbf{G} = [\mathbf{g}_0, \ldots, \mathbf{g}_{K-1}] \in \mathbb{C}^{M \times K}$ as $\mathbf{H} = \mathbf{F}\mathbf{G}$. Particularly interesting is the relation between $\mathbf{H}$ and the bipartite graph $\mathcal{G}$, specifically the rank of $\mathbf{H}$ and the size of the largest subgraph in $\mathcal{G}$ that contains a perfect matching. The rigorous statement is given in Theorem 11.1, but before that, we provide the following lemma as a requirement for proving this theorem.

**Definition 11.2 (Matching)** A matching is a set of edges in a graph that do not share any endpoints. A perfect matching is a matching that connects all nodes of the graph.

**Lemma 11.2 (Rank and Perfect Matchings)** *Let* $\mathbf{Q}$ *be an* $r \times r$ *matrix with some elements identically zero and the non-identically zero elements independently drawn from a continuous distribution. Consider a bipartite graph* $\mathcal{Q}$ *with biadjacency matrix* $\mathbf{A}$ *such that* $[\mathbf{A}]_{i,j} = 1$ *if* $[\mathbf{Q}]_{i,j}$ *is not identically zero, and* $[\mathbf{A}]_{i,j} = 0$ *otherwise. Then,* $\mathbf{Q}$ *has rank* $r$ *with probability 1 if and only if* $\mathcal{Q}$ *contains a perfect matching.*

***Proof*** The determinant of $\mathbf{Q}$ is given by the expansion

$$\det(\mathbf{Q}) = \sum_{\iota \in \boldsymbol{\pi}_r} \operatorname{sgn}(\iota) \prod_i [\mathbf{Q}]_{i,\iota(i)},$$

where $\iota$ is a permutation of the set $\{1, 2, \ldots, r\}$, $\boldsymbol{\pi}_r$ is the set of all such permutations, and $\operatorname{sgn}(\iota)$ is either 1 or -1. From the construction of $\mathcal{Q}$, it is clear that the product $\prod_i [\mathbf{Q}]_{i,\iota(i)}$ is non-zero only if the edge subset $\{(i, \iota(i)),\ i = 1, \ldots, r)\}$ is a perfect matching. Hence, if $\mathcal{Q}$ contains a perfect matching, then $\det(\mathbf{Q}) \neq 0$ with probability 1 (and $\operatorname{rank}(\mathbf{Q}) = r$), since the non-identically zero entries of $\mathbf{Q}$ are drawn from a continuous distribution, such that all elements involved in the product $\prod_i [\mathbf{Q}]_{i,\iota(i)}$ are independent. If it does not contain a perfect matching, then $\det(\mathbf{Q}) = 0$ and therefore $\operatorname{rank}(\mathbf{Q}) < r$. □

**Theorem 11.1** *For a submatrix* $\mathbf{G}_{\mathcal{V}',\mathcal{K}'}$ *consisting of rows and columns with indices in* $\mathcal{V}'$ *and* $\mathcal{K}'$*, respectively, define its associated subgraph as the subgraph* $\mathcal{G}' = (\mathcal{V}', \mathcal{K}', \mathcal{E}') \subseteq \mathcal{G}$*. The rank of* $\mathbf{H}$ *is given, with probability 1, by the side length of the largest square intersection submatrix of* $\mathbf{G}$ *whose associated subgraph in* $\mathcal{G}$ *contains a perfect matching.*

***Proof*** Note that since $\mathbf{H} = \mathbf{F}\mathbf{G}$ and $\mathbf{F}$ is unitary, the rank of $\mathbf{H}$ is equal to that of $\mathbf{G}$. In addition, the rank of $\mathbf{G}$ is equivalent to the largest order of any non-zero minor in $\mathbf{G}$,[8] i.e., the side length of the largest non-singular square submatrix of $\mathbf{G}$. The elements of $\mathbf{G}$ are either identically zero or drawn from a Gaussian distribution with zero mean and a variance $[\boldsymbol{\lambda}_k]_m$ for some $(k, m) \in [K] \times [M]$. Now, according to Lemma 11.2, any such submatrix $\mathbf{Q}$ is non-singular (has rank equal to its side length) if and only if its associated subgraph $\mathcal{Q} \subseteq \mathcal{G}$ contains a perfect matching. This concludes the proof. □

Theorem 11.1 implies that the rank of the channel matrix is given, with probability 1, by the size of a certain matching in the user-virtual beam bipartite graph $\mathcal{G}$. This matching is contained in a subgraph of $\mathcal{G}' = (\mathcal{V}', \mathcal{K}', \mathcal{E}') \subseteq \mathcal{G}$ that specifies the selected users and virtual beams that must satisfy certain criteria. In particular, given a pilot dimension $T_{\mathrm{dl}}$, we want to select $\mathcal{G}'$ such that which node on either color class $\mathcal{K}'$ or $\mathcal{V}'$ has a degree of at least one, and such that:

[8] A minor of a matrix $\mathbf{G}$ is the determinant of some square submatrix of $\mathbf{G}$.

**Stability constraint:** For all $k \in \mathcal{K}'$, we should have $\deg_{\mathcal{G}'}(k) \leq T_{\mathrm{dl}}$, where $\deg_{\mathcal{G}'}$ denotes the degree of a node in the selected subgraph.

**Power constraint:** The sum of weights of the edges incident to any node $k \in \mathcal{K}'$ in the subgraph $\mathcal{G}'$ is greater than a threshold, i.e., $\sum_{m \in \mathcal{N}_{\mathcal{G}'}(k)} w_{m,k} \geq P_{\mathrm{th}}$, for some $P_{\mathrm{th}}$ and for all $k$.

**Rank objective:** The channel matrix $\mathbf{G}_{\mathcal{V}',\mathcal{K}'}$ obtained from $\mathbf{G}$ by selecting $a \in \mathcal{V}'$ ("selected beam directions") and $k \in \mathcal{K}'$ ("selected users") has large rank.

The first constraint enables stable estimation of the effective channel of any selected user with only $T_{\mathrm{dl}}$ common pilot dimensions and $T_{\mathrm{dl}}$ complex symbols of feedback per selected user. The second constraint makes sure that the effective channel strength of any selected user is greater than a desired threshold, since we do not want to spend resources on probing and serving users with weak effective channels (where "weak" is quantitatively determined by the value of $P_{\mathrm{th}}$). Therefore, $P_{\mathrm{th}}$ is a parameter that serves to obtain a trade-off between the rank of the effective matrix (which ultimately determines the number of spatially multiplexed DL data streams) and the beamforming gain (i.e., the power effectively conveyed along each selected user effective channel). The objective is motivated by the equivalence between the channel matrix rank and the system multiplexing gain. In fact, one can show that the pre-log factor in the total sum rate is given by $\mathrm{rank}(\mathbf{G}_{\mathcal{V}',\mathcal{K}'}) \times \max\{0, 1 - T_{\mathrm{dl}}/T\}$, and it is obtained by serving a number of users equal to the rank of the effective channel matrix. We can summarize these in the form of the following problem.

**Problem 11.1** Let $T_{\mathrm{dl}}$ denote the available DL pilot dimension, and let $\mathcal{M}(\mathcal{V}', \mathcal{K}')$ denote a matching of the subgraph $\mathcal{G}'(\mathcal{V}', \mathcal{K}', \mathcal{E}')$ of the bipartite graph $\mathcal{G}(\mathcal{V}, \mathcal{K}, \mathcal{E})$. Find the solution of the following optimization problem:

$$\underset{\mathcal{V}' \subseteq \mathcal{V}, \mathcal{K}' \subseteq \mathcal{K}}{\text{maximize}} \quad \left|\mathcal{M}\left(\mathcal{V}', \mathcal{K}'\right)\right| \tag{11.18a}$$

$$\text{subject to} \quad \deg_{\mathcal{G}'}(k) \leq T_{\mathrm{dl}} \ \forall k \in \mathcal{K}', \tag{11.18b}$$

$$\sum_{m \in \mathcal{N}_{\mathcal{G}'}(k)} w_{m,k} \geq P_{\mathrm{th}}, \ \forall k \in \mathcal{K}'. \tag{11.18c}$$

$\Diamond$

The theorem below shows that this problem can be cast as a mixed-integer linear program (MILP). We refer the interested reader to [29] for the proof.

**Theorem 11.2** *The optimization problem in* (11.18) *is equivalent to the MILP below:*

$$\underset{x_m, y_k, z_{m,k}}{\textit{maximize}} \quad \sum_{m \in \mathcal{V}, k \in \mathcal{K}} z_{m,k} \tag{11.19a}$$

$$\textit{subject to} \quad z_{m,k} \leq [\mathbf{A}]_{m,k} \ \ \forall m \in \mathcal{V}, k \in \mathcal{K}, \tag{11.19b}$$

$$\sum_{k\in\mathcal{K}} z_{m,k} \le x_m \quad \forall m \in \mathcal{V}, \tag{11.19c}$$

$$\sum_{m\in\mathcal{V}} z_{m,k} \le y_k \quad \forall k \in \mathcal{K}, \tag{11.19d}$$

$$\sum_{m\in\mathcal{V}} [\mathbf{A}]_{m,k} x_m \le T_{dl} y_k + M(1 - y_k) \; \forall k \in \mathcal{K} \tag{11.19e}$$

$$P_{th}\, y_k \le \sum_{m\in\mathcal{V}} [\mathbf{W}]_{m,k} x_m \quad \forall k \in \mathcal{K}, \tag{11.19f}$$

$$x_m \le \sum_{k\in\mathcal{K}} [\mathbf{A}]_{m,k}\, y_k \quad \forall m \in \mathcal{V}, \tag{11.19g}$$

$$x_m,\, y_k \in \{0, 1\} \quad \forall a \in \mathcal{V}, k \in \mathcal{K}, \tag{11.19h}$$

$$z_{m,k} \in [0, 1] \quad \forall m \in \mathcal{V}, k \in \mathcal{K}. \tag{11.19i}$$

*The solution subgraph is given by the set of nodes* $\mathcal{V}' = \{m : x_m^\star = 1\}$ *and* $\mathcal{K}' = \{k : y_k^\star = 1\}$, *with* $\{x_m^\star\}_{m=0}^{M-1}$ *and* $\{y_k^\star\}_{k=0}^{K-1}$ *being a solution of* (11.19).

The solution to this optimization, however, is not necessarily unique, i.e., there may exist several subgraphs with the same (maximum) matching size. For example, consider the miniature beam–user bipartite graph of Fig. 11.2 and suppose that we have a pilot dimension $T_{\text{dl}} = 2$. Here the matching $\mathcal{M} = \{(0, 0), (1, 1)\}$ is a matching of maximum size that is contained in two subgraphs, the first one defined by beams $\mathcal{V}_1 = \{0, 1\}$ and users $\mathcal{K}_1 = \{0, 1\}$, and the second one defined by the beams $\mathcal{V}_2 = \{0, 1, 2\}$ and users $\mathcal{K}_2 = \{0, 1\}$ (both satisfy the constraints). In such cases, we want to select the subgraph that includes the larger number of beams. In this example, this is the second subgraph. The reason is that as long as adding beams does not violate the stability constraint, we want to probe (and eventually transmit along) more beams, since this naturally increases the beamforming gain due to channel hardening. In fact, a prominent advantage of a massive MIMO system is its high beamforming gain, and in this way, the algorithm encourages solutions that result in larger beamforming gains.

In order to incorporate the preference for more selected beams in sparsification, we introduce a regularization term to the objective of (11.19) to favor solutions containing more active beams. The regularized form of (11.19) is given as

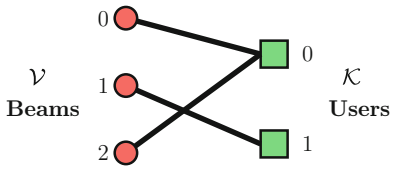



**Fig. 11.2** A toy example of the bipartite graph with $M = 3$ beams and $K = 2$ users

$$\begin{aligned} &\underset{x_m, y_k, z_{m,k}}{\text{maximize}} && \sum_{m\in\mathcal{V}}\sum_{k\in\mathcal{K}} z_{m,k} + \epsilon \sum_{m\in\mathcal{V}} x_m \\ &\text{subject to} && \{x_m, y_k, z_{m,k}\}_{m\in\mathcal{V},k\in\mathcal{K}} \in \mathcal{S}_{\text{feasible}}, \end{aligned} \qquad (\mathcal{P}_{\text{MILP}})$$

where the feasibility set $\mathcal{S}_{\text{feasible}}$ encodes the constraints (11.19a)–(11.19i). Here, the regularization factor $\epsilon$ is chosen to be a small positive value such that it does not affect the matching size of the solution subgraph. In fact choosing $\epsilon < \frac{1}{M}$ ensures this, since then $\epsilon \sum_{m\in\mathcal{V}} x_m < 1$ and a solution to $\mathcal{P}_{\text{MILP}}$ must have the same matching size as a solution to (11.19); otherwise, the objective of $\mathcal{P}_{\text{MILP}}$ can be improved by choosing a solution with a larger matching size. The introduced MILP can be efficiently solved using an off-the-shelf optimization toolbox. In the simulation results of this chapter, we have used the MATLAB intlinprog, which adopts a branch-and-bound method to find the solution to an MILP [42]. Figure 11.3 illustrates an example of the beam–user bipartite graph and how ACS acts on it with a pilot dimension of $T_{\text{dl}} = 2$. The algorithm selects a subgraph containing a matching of maximum size while not violating the estimation stability constraint (assuming for simplicity that the power constraint is satisfied). The selected beams in this graph, i.e., the nodes $\{1, 2, 3, 5, 6\} \subset \mathcal{V}$, are additionally represented by the highlighted rows of the channel matrix in Fig. 11.1.

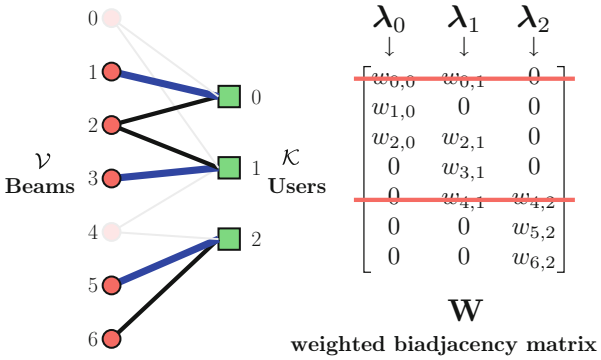



**Fig. 11.3** An example of the beam–user bipartite graph, its weighted biadjacency matrix, and the sparsification process with $M = 7$, $K = 3$ and the assumed pilot dimension $T_{\text{dl}} = 2$. The faded nodes represent the "eliminated" beams ($\{m : x_m^\star = 0\}$), schematically crossed out in the weighted biadjacency matrix. The non-faded edges (in blue and black) represent the user–beam connections that exist in the selected subgraph. The blue edges further highlight the matching of maximum size

### 11.4.3 Channel Estimation and Multiuser Precoding

For a given set of user covariance matrices, let $\{x_m^\star\}$ and $\{y_k^\star\}$ denote the MILP solutions, and denote by $\mathcal{B} = \{m : x_m^\star = 1\} = \{m_1, m_2, \ldots, m_{M'}\}$ the set of selected beam directions of cardinality $|\mathcal{B}| = M'$ and by $\mathcal{K} = \{k : y_k^\star = 1\}$ the set of selected users of cardinality $|\mathcal{K}| = K'$. The resulting sparsifying precoding matrix $\mathbf{B}$ in (11.13) is simply obtained as

$$\mathbf{B} = \mathbf{F}_{\mathcal{B}}^{\mathsf{H}}, \tag{11.20}$$

where $\mathbf{F}_{\mathcal{B}} = [\mathbf{f}_{m_1}, \ldots, \mathbf{f}_{m_{M'}}]$ and $\mathbf{f}_m$ denotes the $m$-th column of the $M \times M$ unitary DFT matrix $\mathbf{F}$. For a DFT column $\mathbf{f}_m$, we have

$$\mathbf{B}\mathbf{f}_m = \begin{cases} \mathbf{0} & \text{if } m \notin \mathcal{B} \\ \mathbf{u}_i & \text{if } m = m_i \in \mathcal{B}, \end{cases}$$

where $\mathbf{u}_i$ denotes a $M' \times 1$ vector with all zero components but a single "1" in the $i$-th position. Using the above property and (11.11), the effective DL channel vectors take on the form

$$\widetilde{\mathbf{h}}_k = \mathbf{B} \sum_{m \in \mathcal{S}_k} [\mathbf{g}_k]_m \sqrt{[\boldsymbol{\lambda}_k]_m}\, \mathbf{f}_m = \sum_{i : m_i \in \mathcal{B} \cap \mathcal{S}_k} \sqrt{[\boldsymbol{\lambda}_k]_{m_i}}\, [\mathbf{g}_k]_{m_i} \mathbf{u}_i . \tag{11.21}$$

In words, the effective channel of user $k$ is a vector with non-identically zero elements only at the positions corresponding to the intersection of the beam directions in $\mathcal{S}_k$, along which the physical channel of user $k$ carries positive energy, and in $\mathcal{B}$, selected by the sparsifying precoder. The non-identically zero elements are independent Gaussian coefficients $\sim \mathcal{CN}(0, [\boldsymbol{\lambda}_k]_{m_i})$. Notice also that, by construction, the number of non-identically zero coefficients are $|\mathcal{B} \cap \mathcal{S}_k| \leq T_{\mathrm{dl}}$ and their positions (encoded in the vectors $\mathbf{u}_i$ in (11.21)), plus an estimate of their variances $[\boldsymbol{\lambda}_k]_{m_i}$, are known to the BS. Hence, the effective channel vectors can be estimated from the $T_{\mathrm{dl}}$-dimensional DL pilot observation (11.13) with an estimation MSE that vanishes as 1/SNR. The pilot observation in the form (11.13) is obtained at the user $k$ receiver. In this chapter, we assume that each user sends its pilot observations using $T_{\mathrm{dl}}$ channel uses in the UL, using analog unquantized feedback, as analyzed for example in [7, 32]. At the BS receiver, after estimating the UL channel from the UL pilots, the BS can apply linear MMSE estimation and recovers the channel-state feedback that takes on the same form as (11.13) with some additional noise due to the noisy UL transmission.

*Remark 11.1* As an alternative, one can consider quantized feedback using $T_{\mathrm{dl}}$ channel uses in the UL. Digital quantized feedback yields generally a better end-to-end estimation MSE in the absence of feedback errors. However, the effect of decoding errors on the channel-state feedback is difficult to characterize in a

simple manner since it depends on the specific joint source-channel coding scheme employed. Hence, in this chapter, we restrict to the simple analog feedback.

With the above precoding, we have $\mathbf{B}\mathbf{B}^{\mathsf{H}} = \mathbf{I}_{M'}$. There are several options for selecting the matrix $\mathbf{\Psi}$, among which we take $\mathbf{\Psi}$ to be proportional to an arbitrary unitary matrix of dimension $T_{\mathrm{dl}} \times M'$, such that $\mathbf{\Psi}\mathbf{\Psi}^{\mathsf{H}} = P_{\mathrm{dl}}\mathbf{I}_{T_{\mathrm{dl}}}$. In this way, the DL pilot phase power constraint (11.14) is automatically satisfied. The estimation of $\widetilde{\mathbf{h}}_k$ from the DL pilot observations (11.13) (with suitably increased AWGN variance due to the noisy UL feedback) is completely straightforward and shall not be treated here in detail.

For the sake of completeness, we conclude this section with the DL precoded data phase and the corresponding sum-rate performance metric that we shall later use for numerical analysis and comparison with other schemes. Let $\widetilde{\mathbf{H}}^{\star} = [\widetilde{\mathbf{h}}_0^{\star}, \ldots, \widetilde{\mathbf{h}}_{K'-1}^{\star}]$ be the matrix of the estimated effective channels for the selected users, where we have assumed without loss of generality the order $\{0, 1, \ldots, K'-1\}$ for the selected users. We consider the ZF beamforming matrix $\mathbf{V}$ given by the column-normalized version of the Moore–Penrose pseudoinverse of the estimated channel matrix, i.e.,

$$\mathbf{V} = \left(\widetilde{\mathbf{H}}^{\star}\right)^{\dagger} \mathbf{J}^{1/2},$$

where $\left(\widetilde{\mathbf{H}}^{\star}\right)^{\dagger} = \widetilde{\mathbf{H}}^{\star}\left(\widetilde{\mathbf{H}}^{\star\mathsf{H}}\widetilde{\mathbf{H}}^{\star}\right)^{-1}$ and $\mathbf{J}$ is a diagonal matrix that makes the columns of $\mathbf{V}$ to have unit norm. A channel use of the DL precoded data transmission phase at the $k$-th user receiver takes on the form $r_k = (\mathbf{h}_k)^{\mathsf{H}}\mathbf{B}^{\mathsf{H}}\mathbf{V}\mathbf{P}^{1/2}\mathbf{d} + n_k,$, where $\mathbf{d} \in \mathbb{C}^{K'\times 1}$ is a vector of unit-energy user data symbols and $\mathbf{P}$ is a diagonal matrix defining the power allocation to the DL data streams. The transmit power constraint is given by $\mathrm{tr}(\mathbf{B}^{\mathsf{H}}\mathbf{V}\mathbf{P}\mathbf{V}^{\mathsf{H}}\mathbf{B}) = \mathrm{tr}(\mathbf{V}^{\mathsf{H}}\mathbf{V}\mathbf{P}) = \mathrm{tr}(\mathbf{P}) = P_{\mathrm{dl}}$, where we used $\mathbf{B}\mathbf{B}^{\mathsf{H}} = \mathbf{I}_{M'}$ and the fact that $\mathbf{V}^{\mathsf{H}}\mathbf{V}$ has unit diagonal elements by construction. In particular, in the simulation results section, we use the simple uniform power allocation $P_k = P_{\mathrm{dl}}/K'$ to each $k$-th user data stream. In the case of perfect ZF beamforming, i.e., for $\widetilde{\mathbf{H}}^{\star} = \widetilde{\mathbf{H}}$, we have $r_k = \sqrt{J_k P_k}d_k + n_k$, where $J_k$ is the $k$-th diagonal element of the norm normalizing matrix $\mathbf{J}$, $P_k$ is the $k$-th diagonal element of the power allocation matrix $\mathbf{P}$, and $d_k$ is the $k$-th user data symbol. Since in general $\widetilde{\mathbf{H}}^{\star} \neq \widetilde{\mathbf{H}}$, due to non-zero estimation error, the received symbol at user $k$ receiver is given by $r_k = b_{k,k}d_k + \sum_{k'\neq k} b_{k,k'}d_{k'} + n_k,$, where the coefficients $(b_{k,1}, \ldots, b_{k,K'})$ are given by the elements of the $1 \times K'$ row vector $(\mathbf{h}_k)^{\mathsf{H}}\mathbf{B}^{\mathsf{H}}\mathbf{V}\mathbf{P}^{1/2}$. Of course, in the presence of an accurate channel estimation, we expect that $b_{k,k} \approx \sqrt{J_k P_k}$ and $b_{k,k'} \approx 0$ for $k' \neq k$. For simplicity, in this chapter, we compare the performance of the proposed scheme with that of the state-of-the-art CS-based scheme in terms of ergodic sum rate, assuming that all coefficients $(b_{k,1}, \ldots, b_{k,K'})$ are known to the corresponding receiver $k$. Including the DL training overhead, this yields the rate expression (see [6])

$$R_{\text{sum}} = \left(1 - \frac{T_{\text{dl}}}{T}\right) \sum_{k \in \mathcal{K}} \mathbb{E}\left[\log\left(1 + \frac{|b_{k,k}|^2}{1 + \sum_{k' \neq k} |b_{k,k'}|^2}\right)\right]. \tag{11.22}$$

## 11.5 Simulation Results

In this section, we provide simulation results to see the empirical evidence for the performance of ACS. We also compare ACS to two of the most recent CS-based methods proposed in [47] and [15] in terms of channel estimation error and sum rate. In [47], the authors proposed a method based on common probing of the DL channel with random Gaussian pilots. The DL pilot measurements $\mathbf{y}_k$ at users $k = 1, \ldots, K$ (similar to (11.13), but with a different pilot matrix) are fed back and collected by the BS, which recovers the channel vectors using a joint orthogonal matching pursuit (J-OMP) technique able to exploit the possible common sparsity between the user channels. In [15], a method based on dictionary learning for sparse channel estimation was proposed. In this scheme, the BS jointly *learns* sparsifying dictionaries for the UL and DL channels by collecting channel measurements at different cell locations (e.g., via an offline learning phase). The actual user channel estimation is posed as a norm-minimization convex program using the trained dictionaries and with the constraint that UL and DL channels share the same support over their corresponding dictionaries. Following [15], we refer to this method as JDLCM.

### *11.5.1 Channel Estimation Error and Sum Rate vs. Pilot Dimension*

In order to have a fair comparison, we involve the UL pilot transmission step in the simulation. The JDLCM method requires UL and DL instantaneous channel samples to train its sparse-representation dictionaries. Our method (ACS) uses the UL channel samples to estimate the UL covariance and then uses that to obtain the DL covariance via a transformation (see [22] for details). Then it uses the *estimated* (and not the true) DL covariance to perform sparsification. The J-OMP method does not make any use of either the UL or DL covariance, and it is not clear how one can incorporate the covariance information in this algorithm.

For this comparison, we considered $M = 128$ antennas at the BS, $K = 13$ users, and resource blocks of size $T = 128$ symbols. For JDLCM, the sparse-representation dictionary is jointly trained for $N = 1000$ instantaneous UL and DL channels as proposed in [15]. For ACS, the BS computes the users' sample UL covariance matrices by taking $N = 1000$ UL pilot observations and then runs a non-negative least-square optimization to estimate a parametric form of the angular

scattering function $\gamma$ in (11.6). This estimate is then used to transform the UL covariance to the DL covariance. Given the obtained DL channel covariance matrix estimates, we first perform the circulant approximation and extract the vector of approximate eigenvalues $\boldsymbol{\lambda}_k,\ k \in [K]$. Then, we compute the sparsifying precoder $\mathbf{B}$ via the MILP solution. In the results presented here, we set the parameter $P_{\text{th}}$ in the MILP to a small value in order to favor a high rank of the resulting effective channel matrix over the beamforming gain.[9] After probing the effective channel of the selected users along these active beam directions via a random unitary pilot matrix $\boldsymbol{\Psi}$, we calculate their MMSE estimate using the estimated DL covariance matrices.

Eventually, for all the three methods, we compute the ZF beamforming matrix based on the obtained channel estimates. In addition, instead of considering all selected users, in both cases, we apply the Greedy ZF user selection approach of [14], which yields a significant benefit when the number of users is close to the rank of the effective channel matrix. As said before, the DL SNR is given by $\text{SNR} = P_{\text{dl}}/N_0$, and during the simulations, we consider ideal noiseless feedback for simplicity, i.e., we assume that the BS receives the measurements in (11.13) without extra feedback noise to the system.[10] The sparsity order of each channel vector is given as an input to the J-OMP method, but not to the other two methods. This represents a genie-aided advantage for J-OMP that we introduce here for simplicity. As the simulation geometry, we consider three MPC clusters with random locations within the angular range (parametrized by $\xi$ rather than $\theta$) $[-1, 1)$. We denote by $\Xi$ the $i$-th interval and set each interval size to be $|\Xi_i| = 0.2,\ i = 1, 2, 3$. The ASF for each user is obtained by selecting at random two out of three such clusters, such that the overlap of the angular components among users is large. The ASF is non-zero over the angular intervals corresponding to the chosen MPCs and zero elsewhere, i.e., $\gamma_k(\xi) = \beta \mathbf{1}_{\Xi_{i_1} \cup \Xi_{i_2}},$, where $\beta = 1/\int_{-1}^{1} \gamma_k(\xi) d\xi$ and $i_1, i_2 \in \{1, 2, 3\}$.

The described arrangement results in each generated channel vector being roughly $s_k = 0.2 \times M \approx 26$-sparse. To measure channel estimation error, we use the normalized Euclidean distance as follows. Let $\mathbf{H} \in \mathbb{C}^{M \times K'}$ define the matrix whose columns correspond to the channel vectors of the $K'$ served users, and let $\widehat{\mathbf{H}}$ denote the estimation of $\mathbf{H}$. Then the normalized Euclidean error is defined as

$$E_{\text{euc}} = \mathbb{E}\left[ \frac{\|\mathbf{H} - \widehat{\mathbf{H}}\|_{\mathsf{F}}^2}{\|\mathbf{H}\|_{\mathsf{F}}^2} \right].$$

[9] This approach is appropriate in the medium to high-SNR regime. For low SNR, it is often convenient to increase $P_{\text{th}}$ in order to serve less users with a larger beamforming energy transfer per user.

[10] Notice that by introducing noisy feedback, the relative gain with respect to J-OMP is even larger, since CS schemes are known to be more noise-sensitive than plain MMSE estimation using estimated DL covariance matrices.

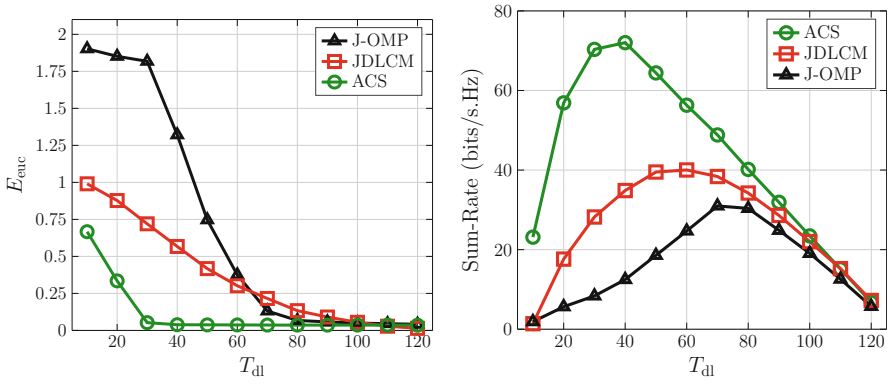



**Fig. 11.4** (**a**) Normalized channel estimation error, and (**b**) achievable sum rate as a function of DL pilot dimension with SNR $= 20$ dB, $M = 128$, and $K = 13$.

Figure 11.4a shows the normalized channel estimation error for the J-OMP, JDLCM, and our proposed active channel sparsification (ACS) method as a function of the DL pilot dimension $T_{\text{dl}}$ with SNR $= 20$ dB. Our ACS method outperforms the other two by a large margin, especially for low DL pilot dimensions. When the pilot dimension is below channel sparsity order, CS-based methods perform very poorly, since the number of channel measurements is less than the inherent channel dimension. Figure 11.4b compares the achievable sum rate for the three methods. Again our ACS method shows a much better performance compared to J-OMP and JDLCM. This figure also shows that there is an optimal DL pilot dimension that maximizes the sum rate. This optimal value is $T_{\text{dl}} \approx 40$ for our proposed method, $T_{\text{dl}} \approx 60$ for JDLCM, and $T_{\text{dl}} \approx 70$ for J-OMP.

### *11.5.2 The Effect of Channel Sparsity*

The channels can have various sparsity levels in the angular domain. While the CS-based method is in this sense at the mercy of environmental features, our active sparsification method is able to deal with different scenarios by inducing more sparsity in the channel. This section examines the effect of channel sparsity order on sum rate when the ACS method is employed. We take user ASFs to consist of two clusters, chosen at random, out of three pre-defined clusters. To have different sparsity orders, we vary the size of the angular interval, each of the clusters occupies ($|\Xi_i| = 0.2, 0.4, 0.6, 0.8$), and we see how it affects the error and sum-rate metrics. The sparsifying precoder, DL training, feedback, and data precoding are performed as before. We take $M = 128$, and as the ASF consists of two clusters, the channel sparsity order takes on the values $s_k = 26, 51, 77, 102$ for all users $k \in [K']$. Then

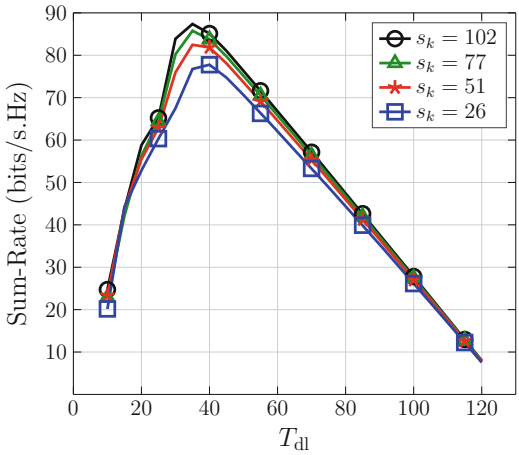



**Fig. 11.5** Sum rate vs. $T_{\mathrm{dl}}$ for various channel sparsity orders. Here SNR = 20 dB, $M = 128$, and $K = 13$.

the system sum rate is calculated empirically for each pilot dimension via Monte Carlo simulations. Notice that in these results we fix the channel coefficient power along each scattering component, such that richer (less sparse) channels convey more signal energy. This corresponds to the physical fact that the more scattered signal energy is collected at the receiving antennas the higher the received signal energy is. Figure 11.5 illustrates the results. As we can see, for a fixed $T_{\mathrm{dl}}$, when the number of non-zero channel coefficients increases, i.e., when the channel is less sparse, we generally have a larger sum rate. This is due to the fact that, with less sparse channels, the beamforming gain is larger, since more scattering components contribute to the channel. Therefore, we can generally say that with our method, for a fixed pilot dimension, less sparse channels are, in a sense, better. Of course, this is not the case for techniques based on the sparsity assumption of a small number of discrete angular components, which tend to collapse and yield poor results when such sparsity assumptions are not satisfied.

## 11.6 Beam-Space Design for Arbitrary Array Geometries

The ACS can be applied to design precoders for cases with array geometries other than the ULA. As explained in Sect. 11.3, a necessary step before performing sparsification in a tractable way is that all channels share the same (approximate) eigenspace. Earlier in this chapter, we observed that, for a massive ULA, this common eigenspace is asymptotically given by the span of the DFT basis due to an application of Szegö's theorem for large Hermitian Toeplitz matrices (see Sect. 11.3). For an arbitrary array geometry, the covariance is not necessarily

Toeplitz, and we are not aware of any work suggesting the existence of a (approximate) common eigenspace for MIMO channels of generic array geometries. Then, what is a suitable strategy to obtain an *approximate* common eigenbasis? In order to extend ACS to arrays with arbitrary design, here we propose a method for obtaining an approximate common eigenspace for channels of a multiuser system with an arbitrary array geometry. Once a common eigenbasis is obtained, sparsification can be done simply by performing the MILP on a bipartite graph that encodes the link between the users and the set of obtained virtual beams.

Consider a BS equipped with an array of arbitrary geometry consisting of $M$ antennas, communicating with $K$ users. The user channels are all assumed zero-mean, complex Gaussian with covariances $\mathbf{\Sigma}_k = \mathbb{E}\left[\mathbf{h}_k \mathbf{h}_k^{\mathsf{H}}\right]$, $k \in [K]$. Define the eigendecomposition of $\mathbf{\Sigma}_k$ as $\mathbf{\Sigma}_k = \mathbf{U}_k \mathbf{\Lambda}_k \mathbf{U}_k^{\mathsf{H}}$, where $\mathbf{U}_k$ is the unitary matrix of eigenvectors ($\mathbf{U}_k^{\mathsf{H}} \mathbf{U}_k = \mathbf{I}_M$) and $\mathbf{\Lambda}_k$ is the diagonal matrix of non-negative eigenvalues. We note that the eigenbasis of distinct covariances is generally different. This makes the joint processing of the channels and the precoding design difficult. Hence, we are interested in obtaining an approximate common eigenbasis $\mathbf{U}$ among all covariances $\{\mathbf{\Sigma}_k\}$. An ideal choice for $\mathbf{U}$ is one that "approximately" diagonalizes all the members of $\{\mathbf{\Sigma}_k\}$, such that the diagonal elements of $\mathbf{U}^{\mathsf{H}} \mathbf{\Sigma} \mathbf{U}$ "closely" follow the true eigenvalues of $\mathbf{\Sigma}_k$ for all $k$. If the covariances are in fact jointly diagonalizable, i.e., if there exists a unitary matrix $\mathbf{U}^c$ such that $\mathbf{U}_1 = \mathbf{U}_2 = \ldots = \mathbf{U}_K = \mathbf{U}^c$, then it is desirable to obtain $\mathbf{U}^c$ as the common eigenbasis. If the covariances are *not* jointly diagonalizable, then we want to obtain a unitary matrix $\mathbf{U}^\star$ that "best" diagonalizes the covariances.

Some of the present authors have studied this problem in a slightly different context in [30], where the goal was to obtain the approximate common eigenbasis $\mathbf{U}$ given a number of $N$ samples of the instantaneous channels of the $K$ users. There, they assumed a parametric form of the covariances as $\{\mathbf{\Sigma}_k = \mathbf{U} \mathbf{\Lambda}_k' \mathbf{U}^{\mathsf{H}}\}$ and performed a maximum likelihood (ML) estimation of $\mathbf{U}$ and $\{\mathbf{\Lambda}_k'\}$ given the samples. It turns out that the emergent ML problem can be cast as the following optimization:

$$\underset{\{\mathbf{u}_m\}}{\text{minimize}} \sum_{m,k} \log\left(\mathbf{u}_m^{\mathsf{H}} \widehat{\mathbf{\Sigma}}_k \mathbf{u}_m\right) \quad \text{subject to} \quad \mathbf{u}_m^{\mathsf{H}} \mathbf{u}_n = \delta_{m,n},\ m, n \in [M], \tag{11.23}$$

where $\widehat{\mathbf{\Sigma}}_k = \frac{1}{N} \sum_{n=0}^{N-1} \mathbf{h}_k[i] \mathbf{h}_k[i]^{\mathsf{H}}$ is the sample covariance of the instantaneous channels for $k \in [K]$. However, in this chapter, we have assumed that the channel covariances (or estimates thereof) are readily given. Then it seems natural to apply the same formulation by substituting the sample covariance with the available (true or estimated) covariances. This results in the following problem:

$$\begin{aligned} \underset{\{\mathbf{u}_m\}}{\text{minimize}} \quad & f(\mathbf{U}) = \sum_{m,k} \log\left(\mathbf{u}_m^{\mathsf{H}} \mathbf{\Sigma}_k \mathbf{u}_m\right) \\ \text{subject to} \quad & \mathbf{u}_m^{\mathsf{H}} \mathbf{u}_n = \delta_{m,n},\ m, n \in [M], \end{aligned} \tag{$\mathcal{P}_1$}$$

which presents an optimization over the manifold of unitary matrices $\mathcal{U} = \{\mathbf{U} \in \mathbb{C}^{M\times M} : \mathbf{U}^{\mathsf{H}}\mathbf{U} = \mathbf{I}_M\}$. To solve the ML problem $\mathcal{P}_1$, we propose a gradient projection method and show that it converges to a stationary point of the cost function $f$. But first, let us study the problem when applied to a special class of user covariances.

### 11.6.1 Jointly Diagonalizable Covariances

One can show that, if the true channel covariances are jointly diagonalizable, then the global optimum of $\mathcal{P}_1$ is given by the common eigenbasis. To see this, first note that the channel covariance of user $k$ can be decomposed as $\mathbf{\Sigma}_k = \mathbf{U}^c\mathbf{\Lambda}_k\mathbf{U}^{c\,\mathsf{H}}$, for $k \in [K]$, where $\mathbf{U}^c \in \mathbb{C}^{M\times M}$, $\mathbf{U}^{c\,\mathsf{H}}\mathbf{U}^c = \mathbf{I}_M$ denotes the common eigenbasis.

**Definition 11.3 (Majorization)** For $\mathbf{x} \in \mathbb{R}^M$, define $\mathbf{x}^{\downarrow}$ as the vector containing the elements of $\mathbf{x}$ in descending order. Let $\mathbf{y} \in \mathbb{R}^M$ be another vector such that $\sum_{i=0}^{M-1}[\mathbf{x}]_i = \sum_{i=0}^{M-1}[\mathbf{y}]_i$. We say $\mathbf{x}$ *majorizes* $\mathbf{y}$ ($\mathbf{x} \succ \mathbf{y}$) iff $\sum_{i=0}^{m}\mathbf{x}_i^{\downarrow} \geq \sum_{i=0}^{m}\mathbf{y}_i^{\downarrow}$, for all $m \in [M]$.

The following theorem shows that $\mathbf{U}^c$ is a global optimum of $\mathcal{P}_1$.

**Theorem 11.3** *Let $\mathbf{\Sigma}_k$, $k = 0, \ldots, K-1$, be a set of jointly diagonalizable covariance matrices. Then $\mathbf{U}^\star = \mathbf{U}^c$ is a global optimum of $\mathcal{P}_1$.*

***Proof*** Given a unitary matrix $\mathbf{U}$, let $\boldsymbol{\sigma}_k(\mathbf{U}) \in \mathbb{R}^M$ be a vector such that $[\boldsymbol{\sigma}_k(\mathbf{U})]_m = \mathbf{u}_m^{\mathsf{H}}\mathbf{\Sigma}_k\mathbf{u}_m$. Particularly, we can check that $\boldsymbol{\sigma}_k(\mathbf{U}^c)$ is the vector of eigenvalues of $\mathbf{\Sigma}_k$. From the properties of eigenvalue decomposition, it follows that $\boldsymbol{\sigma}_k(\mathbf{U}^c) \succ \boldsymbol{\sigma}_k(\mathbf{U})$ for all $\mathbf{U} \in \mathcal{U}$ and all $k \in [K]$. Besides, the function $h(\mathbf{x}) = \sum_i \log([\mathbf{x}]_i)$ is Schur-concave [46], and hence, $\sum_m \log([\boldsymbol{\sigma}_k(\mathbf{U}^c)]_m) \leq \sum_m \log([\boldsymbol{\sigma}_k(\mathbf{U})]_m)$ for all $k$. Therefore, we have $f(\mathbf{U}^c) \leq f(\mathbf{U})$ for all $\mathbf{U} \in \mathcal{U}$, proving $\mathbf{U}^c$ to be the global minimizer of $f$ over $\mathcal{U}$. □

This theorem shows that $\mathcal{P}_1$ satisfies a minimum requirement for finding a set of approximate common eigenvectors: at least when the channel covariances *do* share a common eigenbasis, the ML optimum coincides with it.

### 11.6.2 ML via Projected Gradient Descent

Now we turn to solving the ML problem $\mathcal{P}_1$. We use a projected gradient descent (PGD) method to minimize the objective cost function $f$. The PGD is a well-known iterative optimization algorithm [4], which starts from an initial point $\mathbf{U}^{(0)}$ and

consists of the following two steps per iteration:

$$\widetilde{\mathbf{U}}^{(t)} = \mathbf{U}^{(t)} - \alpha_t \nabla f(\mathbf{U}^{(t)}) \qquad \text{(Gradient Step)}$$

$$\mathbf{U}^{(t+1)} = \mathcal{P}_{\mathcal{U}}(\widetilde{\mathbf{U}}^{(t)}), \qquad \text{(Projection Step)}$$

where $\nabla f(\mathbf{U}^{(t)}) \in \mathbb{C}^{M\times M}$ is the gradient of $f$ at $\mathbf{U}^{(t)}$, $\mathcal{P}_{\mathcal{U}} : \mathbb{C}^{M\times M} \to \mathcal{U}$ is the orthogonal projection operator onto the set of unitary matrices, and $\alpha_t > 0$ is a step size. The following lemma, proved in [38], shows how one can compute the orthogonal projection.

**Lemma 11.3** *Let $\mathbf{V} \in \mathbb{C}^{M\times M}$ be a matrix with singular value decomposition $\mathbf{V} = \mathbf{S}\mathbf{D}\mathbf{T}^{\mathsf{H}}$, where $\mathbf{S}$ and $\mathbf{T}$ are unitary matrices of left and right eigenvectors and $\mathbf{D} = diag(\mathbf{d})$ is non-negative diagonal. Then, the orthogonal projection of $\mathbf{V}$ onto the set of unitary matrices is given by $\mathcal{P}_{\mathcal{U}}(\mathbf{V}) = \mathbf{S}\mathbf{T}^{\mathsf{H}}$.*

The following theorem presents a guarantee for the convergence of PGD when applied to solve $\mathcal{P}_1$ (see [30] for the proof).

**Theorem 11.4** *Let $\mathbf{U}^{(0)} \in \mathcal{U}$ be an initial point and consider the gradient projection update rule*

$$\mathbf{U}^{(t+1)} = \mathcal{P}_{\mathcal{U}}\left(\mathbf{U}^{(t)} - \alpha_t \nabla f(\mathbf{U}^{(t)})\right), \quad t = 0, 1, \ldots, \tag{11.24}$$

*with $\alpha_t \in (0, \frac{1}{L})$ for all $t$, where $L$ is the Lipschitz constant of $\nabla f(\mathbf{U})$. Then the sequence $\{\mathbf{U}^{(t)},\ t = 0, 1, \ldots\}$ converges to a stationary point of $f(\mathbf{U})$.*

Theorem 11.4 guarantees that the sequence generated by PGD converges to a stationary point of the likelihood function. This gives a suitable common eigenbasis that, in a sense, approximately diagonalizes all the user covariance matrices. This basis can serve as the beam-space representation of the channel.

### *11.6.3 Extension of ACS to Arbitrary Array Geometries*

We can directly extend the ACS technique for FDD massive MIMO channels with non-ULA geometries. In Sect. 11.6, we proposed a method of designing a common eigenbasis. Given user channel covariances $\{\mathbf{\Sigma}_k\}$, or their estimates, this method yields a common eigenbasis $\mathbf{U}^\star$, and the user-dependent "approximate eigenvalue" matrices $\mathbf{\Lambda}_k^\star = \text{diag}(\boldsymbol{\lambda}_k^\star)$, $k \in [K]$, where $[\boldsymbol{\lambda}_k^\star]_m = \mathbf{u}_m^{\star\mathsf{H}} \mathbf{\Sigma}_k \mathbf{u}_m$. Eventually, the covariance of user $k$ can be approximated as

$$\mathbf{\Sigma}_k \approx \mathbf{U}^\star \mathbf{\Lambda}_k^\star \mathbf{U}^{\star\mathsf{H}}. \tag{11.25}$$

The eigenbasis $\mathbf{U}^\star$ consists of the array virtual beams. Since this beam space is shared among all users, we can define the bipartite user–beam graph introduced in Sect. 11.4.2. In this case, the edge weight between a user $k$ and a beam $m$ is given by $w_{m,k} = [\boldsymbol{\lambda}_k^\star]_m$. Then we can solve the same matching-size maximization problem in (11.18) through the MILP. Let $\{x_m^\star\}$ denote the MILP solution for the binary variables representing beam nodes and $\{y_k^\star\}$ its solution for binary variables representing user nodes. Also let $\mathcal{B} = \{m : x_m^\star = 1\}$ define the set of active beams and $\mathcal{K} = \{k : y_k^\star = 1\}$ the set of active users. The sparsifying precoder in this case is given as

$$\mathbf{B} = \mathbf{U}_{\mathcal{B}}^{\star\mathsf{H}}. \tag{11.26}$$

The rest of the channel training and precoding procedure is performed just like the ULA case.

*Remark 11.2* Note that since $\mathbf{U}^\star$ is only an approximate eigenbasis of $\boldsymbol{\Sigma}_k$, we cannot guarantee the coefficients of the linear expansion of a random channel vector $\mathbf{h}_k$ in terms of the columns of $\mathbf{U}^\star$ to be independent random variables with a continuous distribution. Hence, we cannot prove that maximizing the matching size in the beam–user bipartite graph is equivalent to maximizing the channel matrix rank. The reason is that the conditions of Lemma 11.2 are violated, since we cannot assume a distribution on the coefficients. Nevertheless, we assume that the error of approximating the covariances as in (11.25) is not large, such that $\mathbf{U}^\star$ is close to $\mathbf{U}_k$ for all $k$. This would lead the coefficients of the expansion in terms of the columns of $\mathbf{U}^\star$ to be close to the Gaussian coefficients of the Karhunen–Loéve expansion. Then maximizing the matching size will maximize the channel matrix rank.

## References

1. Adhikary, A., Nam, J., Ahn, J.Y., Caire, G.: Joint spatial division and multiplexing: the large-scale array regime. IEEE Trans. Inf. Theory **59**(10), 6441–6463 (2013)
2. Ali, A., González-Prelcic, N., Heath, R.W.: Millimeter wave beam-selection using out-of-band spatial information. IEEE Trans. Wirel. Commun. **17**(2), 1038–1052 (2017)
3. Bajwa, W.U., Haupt, J., Sayeed, A.M., Nowak, R.: Compressed channel sensing: A new approach to estimating sparse multipath channels. Proc. IEEE **98**(6), 1058–1076 (2010)
4. Bertsekas, D.P., Scientific, A.: Convex optimization algorithms. In: Athena Scientific Belmont (2015)
5. Boccardi, F., Heath, R.W., Lozano, A., Marzetta, T.L., Popovski, P.: Five disruptive technology directions for 5G. IEEE Commun. Mag. **52**(2), 74–80 (2014)
6. Caire, G.: On the ergodic rate lower bounds with applications to massive MIMO. IEEE Trans. Wirel. Commun. **17**(5), 3258–3268 (2018)
7. Caire, G., Jindal, N., Kobayashi, M., Ravindran, N.: Multiuser MIMO achievable rates with downlink training and channel state feedback. IEEE Trans. Inf. Theory **56**(6), 2845–2866 (2010)

8. Candès, E.J., Wakin, M.B.: An introduction to compressive sampling. IEEE Signal Process. Mag. **25**(2), 21–30 (2008)
9. Chan, P.W., Lo, E.S., Wang, R.R., Au, E.K., Lau, V.K., Cheng, R.S., Mow, W.H., Murch, R.D., Letaief, K.B.: The evolution path of 4G networks: FDD or TDD? IEEE Commun. Mag. **44**(12), 42–50 (2006)
10. Chen, J., Huo, X.: Theoretical results on sparse representations of multiple-measurement vectors. IEEE Trans. Signal Process. **54**(12), 4634–4643 (2006)
11. Dai, J., Liu, A., Lau, V.K.: FDD massive MIMO channel estimation with arbitrary 2d-array geometry. IEEE Trans. Signal Process. **66**(10), 2584–2599 (2018)
12. Davoodi, A.G., Jafar, S.A.: Aligned image sets under channel uncertainty: Settling conjectures on the collapse of degrees of freedom under finite precision CSIT. IEEE Trans. Inf. Theory **62**(10), 5603–5618 (2016)
13. Decurninge, A., Guillaud, M., Slock, D.T.: Channel covariance estimation in massive MIMO frequency division duplex systems. In: Proceedings of the 2015 IEEE Globecom Workshops (GC Wkshps), pp. 1–6. IEEE, New York (2015)
14. Dimic, G., Sidiropoulos, N.D.: On downlink beamforming with greedy user selection: performance analysis and a simple new algorithm. IEEE Trans. Signal Process. **53**(10), 3857–3868 (2005)
15. Ding, Y., Rao, B.D.: Dictionary learning-based sparse channel representation and estimation for FDD massive MIMO systems. IEEE Trans. Wirel. Commun. **17**(8), 5437–5451 (2018)
16. Donoho, D.L.: Compressed sensing. IEEE Trans. Inf. Theory **52**(4), 1289–1306 (2006)
17. Eldar, Y.C., Rauhut, H.: Average case analysis of multichannel sparse recovery using convex relaxation. IEEE Trans. Inf. Theory **56**(1), 505–519 (2010)
18. Gao, X., Edfors, O., Rusek, F., Tufvesson, F.: Linear pre-coding performance in measured very-large MIMO channels. In: Proceedings of the 2011 IEEE Vehicular Technology Conference (VTC Fall), pp. 1–5. IEEE, New York (2011)
19. Gao, Z., Dai, L., Wang, Z., Chen, S.: Spatially common sparsity based adaptive channel estimation and feedback for FDD massive MIMO. IEEE Trans. Signal Process. **63**(23), 6169–6183 (2015)
20. Gray, R.M.: Toeplitz and circulant matrices: A review. Foundations and Trends®in Communications and Information Theory **2**(3), 155–239 (2006). Now publishers inc
21. Grimmett, G.S., et al.: Probability and random processes. Oxford University Press, Oxford (2020)
22. Haghighatshoar, S., Khalilsarai, M.B., Caire, G.: Multi-band covariance interpolation with applications in massive MIMO. In: 2018 IEEE International Symposium on Information Theory (ISIT), pp. 386–390. IEEE, New York (2018)
23. Horn, R.A., Johnson, C.R.: Matrix analysis. Cambridge University Press, Cambridge (1990)
24. Hoydis, J., Hoek, C., Wild, T., ten Brink, S.: Channel measurements for large antenna arrays. In: Proceedings of the 2012 International Symposium on Wireless Communication Systems (ISWCS), pp. 811–815. IEEE, New York (2012)
25. Hugl, K., Kalliola, K., Laurila, J.: Spatial reciprocity of uplink and downlink radio channels in FDD systems. Proc. COST 273 Technical Document TD (02) **66**, 7 (2002)
26. Jiang, Z., Molisch, A.F., Caire, G., Niu, Z.: Achievable rates of FDD massive MIMO systems with spatial channel correlation. IEEE Trans. Wirel. Commun. **14**(5), 2868–2882 (2015)
27. Jindal, N.: MIMO broadcast channels with finite-rate feedback. IEEE Trans. Inf. Theory **52**(11), 5045–5060 (2006)
28. Kaltenberger, F., Gesbert, D., Knopp, R., Kountouris, M.: Correlation and capacity of measured multi-user MIMO channels. In: IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications, 2008 (PIMRC 2008), pp. 1–5. IEEE, New York (2008)
29. Khalilsarai, M.B., Haghighatshoar, S., Yi, X., Caire, G.: FDD massive MIMO via UL/DL channel covariance extrapolation and active channel sparsification. IEEE Trans. Wirel. Commun. **18**(1), 121–135 (2018)

30. Khalilsarai, M.B., Haghighatshoar, S., Caire, G.: Joint approximate covariance diagonalization with applications in MIMO virtual beam design. In: 2020 IEEE Global Communications Conference (GLOBECOM). IEEE, New York (2020)
31. Klebaner, F.C.: Introduction to stochastic calculus with applications. World Scientific, Singapore (2005)
32. Kobayashi, M., Jindal, N., Caire, G.: Training and feedback optimization for multiuser MIMO downlink. IEEE Trans. Commun. **59**(8), 2228–2240 (2011)
33. Kyritsi, P., Cox, D.C., Valenzuela, R.A., Wolniansky, P.W.: Correlation analysis based on MIMO channel measurements in an indoor environment. IEEE J. Sel. Areas Commun. **21**(5), 713–720 (2003)
34. Larsson, E.G., Edfors, O., Tufvesson, F., Marzetta, T.L.: Massive MIMO for next generation wireless systems. IEEE Commun. Mag. **52**(2), 186–195 (2014)
35. Love, D.J., Heath, R.W., Strohmer, T.: Grassmannian beamforming for multiple-input multiple-output wireless systems. IEEE Trans. Inf. Theory **49**(10), 2735–2747 (2003)
36. Lozano, A., Heath, R.W., Andrews, J.G.: Fundamental limits of cooperation. IEEE Trans. Inf. Theory **59**(9), 5213–5226 (2013)
37. Malkowsky, S., Vieira, J., Liu, L., Harris, P., Nieman, K., Kundargi, N., Wong, I.C., Tufvesson, F., Öwall, V., Edfors, O.: The world's first real-time testbed for massive MIMO: Design, implementation, and validation. IEEE Access **5**, 9073–9088 (2017)
38. Manton, J.H.: Optimization algorithms exploiting unitary constraints. IEEE Trans. Signal Process. **50**(3), 635–650 (2002)
39. Marzetta, T.L.: How much training is required for multiuser MIMO? In: Fortieth Asilomar Conference on Signals, Systems and Computers, 2006 (ACSSC'06), pp. 359–363. IEEE, New York (2006)
40. Marzetta, T.L.: Noncooperative cellular wireless with unlimited numbers of base station antennas. IEEE Trans. Wireless Commun. **9**(11), 3590–3600 (2010)
41. Marzetta, T.L., Larsson, E.G., Yang, H., Ngo, H.Q.: Fundamentals of Massive MIMO. Cambridge University, Cambridge (2016)
42. MATLAB: version 9.9.0 (R2010b). The MathWorks Inc., Natick, Massachusetts (2020)
43. Miretti, L., Cavalcante, R.L.G., Stanczak, S.: FDD massive MIMO channel spatial covariance conversion using projection methods. In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 3609–3613. IEEE, New York (2018)
44. Nam, J., Adhikary, A., Ahn, J.Y., Caire, G.: Joint spatial division and multiplexing: Opportunistic beamforming, user grouping and simplified downlink scheduling. IEEE J. Sel. Top. Sign. Proces. (JSTSP) **8**(5), 876–890 (2014)
45. Pascual-García, J., Molina-García-Pardo, J.M., Martinez-Ingles, M.T., Rodriguez, J.V., Saurin-Serrano, N.: On the importance of diffuse scattering model parameterization in indoor wireless channels at mm-wave frequencies. IEEE Access **4**, 688–701 (2016)
46. Peajcariaac, J.E., Tong, Y.L.: Convex functions, partial orderings, and statistical applications. Academic Press, London (1992)
47. Rao, X., Lau, V.K.: Distributed compressive CSIT estimation and feedback for FDD multi-user massive MIMO systems. IEEE Trans. Signal Process. **62**(12), 3261–3271 (2014)
48. Rayal, F.: LTE in a Nutshell: The physical layer. In: Telesystem Innovations (2010)
49. Richter, A.: Estimation of Radio Channel Parameters: Models and Algorithms. ISLE, Blacksburg (2005)
50. Richter, A., Thomä, R.S.: Parametric modeling and estimation of distributed diffuse scattering components of radio channels (2003)
51. Sánchez-Fernández, M., Jamali, V., Llorca, J., Tulino, A.: Gridless multidimensional angle of arrival estimation for arbitrary 3D antenna arrays. IEEE Trans. Wirel. Commun. **20**(7), 4748–4764 (2021)
52. Sayeed, A.M.: Deconstructing multiantenna fading channels. IEEE Trans. Signal Process. **50**(10), 2563–2579 (2002)
53. Sesia, S., Toufik, I., Baker, M.: LTE-the UMTS long term evolution: from theory to practice. Wiley, New York (2011)

54. Sim, M.S., Park, J., Chae, C.B., Heath, R.W.: Compressed channel feedback for correlated massive MIMO systems. J. Commun. Networks **18**(1), 95–104 (2016)
55. Thomä, R., Landmann, M., Richter, A., Trautwein, U.: Multidimensional high-resolution channel sounding. In: Smart Antennas in Europe–State-of-the-Art, vol. 3. Hindawi Publishing Corporation, London (2005)
56. Tse, D., Viswanath, P.: Fundamentals of wireless communication. Cambridge University, Cambridge (2005)
57. Xie, H., Gao, F., Zhang, S., Jin, S.: A unified transmission strategy for TDD/FDD massive MIMO systems with spatial basis expansion model. IEEE Trans. Veh. Technol. **66**(4), 3170–3184 (2017)
58. Xie, H., Gao, F., Jin, S., Fang, J., Liang, Y.C.: Channel estimation for TDD/FDD massive MIMO systems with channel covariance computing. IEEE Trans. Wirel. Commun. **17**(6), 4206–4218 (2018)
59. Yang, H., Marzetta, T.L.: Performance of conjugate and zero-forcing beamforming in large-scale antenna systems. IEEE J. Sel. Areas Commun. **31**(2), 172–179 (2013)
60. Yin, H., Gesbert, D., Filippou, M., Liu, Y.: A coordinated approach to channel estimation in large-scale multiple-antenna systems. IEEE J. Sel. Areas Commun. **31**(2), 264–273 (2013)
61. Zheng, L., Tse, D.N.C.: Communication on the Grassmann manifold: A geometric approach to the noncoherent multiple-antenna channel. IEEE Trans. Inf. Theory **48**(2), 359–383 (2002)

# Chapter 12
# Atmospheric Radar Imaging Improvements Using Compressed Sensing and MIMO


**Jorge Luis Chau, Juan Miguel Urco, Tobias Weber, and Jeremy Olaore Aweda**

## 12.1 Introduction

The focus of this chapter is the signal processing in atmospheric radar imaging (ARI) and in particular its application to the study of polar mesospheric summer echoes (PMSEs). Atmospheric received signals are stochastic and result from the scattering of non-homogeneous atmospheric irregularities. In the case of PMSE, the received signals could be considered quasi-stationary in time scales of a few seconds with correlation times of milliseconds to a few seconds, organized in patches with horizontal sizes of 1–5 kms separated between a few kilometers to tens or hundreds of kilometers, organized in various narrow vertical layers of $\sim 150-600$ m thickness, and a typical signal-to-noise ratio (SNR) after matched filtering varying from $-10$ to 40 dB, e.g., [34]. Moreover, PMSE patches drift horizontally with the background horizontal wind, e.g., [1]. Climatologically, a westward wind of $\sim 30-50$ m/s is observed during the summer polar mesosphere at around 85 km, e.g., [6].

Atmospheric radar systems used in ARI operate at very high frequencies (VHFs) due to the frequency-dependent scattering properties of atmospheric irregularities.

J. L. Chau
Leibniz Institute of Atmospheric Physics at the University of Rostock, Kühlungsborn, Germany
e-mail: jchau@iap-kborn.de

J. M. Urco
Leibniz Institute of Atmospheric Physics at the University of Rostock, Kühlungsborn, Germany

Department of Electrical and Computer Engineering and Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, Champaign, IL, USA
e-mail: urco@iap-kborn.de

T. Weber · J. O. Aweda (✉)
Institut für Nachrichtentechnik, University of Rostock, Rostock, Germany
e-mail: tobias.weber@uni-rostock.de; jeremy.aweda@uni-rostock.de



G. Kutyniok et al. (eds.), *Compressed Sensing in Information Processing*, Applied and Numerical Harmonic Analysis, https://doi.org/10.1007/978-3-031-09745-4_12

This results in huge antenna arrays, and it is crucial to make the best use of the still rather limited number of antennas. Promising approaches are (a) the use of non-uniform antenna arrays, i.e., a pseudo-random sampling of the electromagnetic wave field, (b) the use of multiple-input multiple-output (MIMO) antenna configurations, and (c) the use of tracking techniques to exploit the time dynamics of the PMSE brightness.

We first start with the presentation of the system model for ARI, followed by usual ARI techniques. The application of MIMO and compressed sensing to different atmospheric radar fields is presented in Sect. 12.2.5.

## 12.2 System Model and Inversion Methods for Atmospheric Radar Imaging

This section describes the system model for ARI. The system model for the single-input multiple-output (SIMO) antenna configuration is presented followed by the MIMO case. This section is complemented with a description of inversion methods used in atmospheric radar imaging.

### *12.2.1 System Model for SIMO Atmospheric Radar Imaging*

Figure 12.1 shows the system model with a reference point and one antenna element $k$ at a known position in space with the position vector $\vec{r}_k$. A narrowband signal with the complex amplitude $\underline{s}_{\mathrm{RP},n}$ being backscattered from a single target $n$ in the far-field would be received at the reference point. The direction of arrival is characterized by the wavenumber vector $\vec{\beta}_n$, which is described in terms of the unit vector $\vec{u}_n$ into the direction of the target $n$ and the wavelength $\lambda$ as

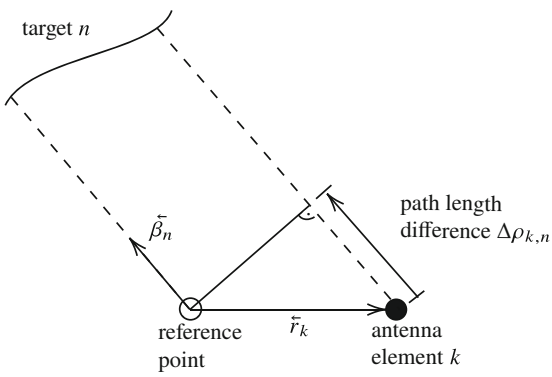



**Fig. 12.1** Model for SIMO atmospheric radar imaging

$$\vec{\beta}_n = \frac{2\pi}{\lambda}\vec{u}_n. \tag{12.1}$$

Due to the position $\vec{r}_k$ of the antenna element $k$ in space, there is a path length difference $\Delta\rho_{k,n}$ between the antenna element $k$ and the reference point. This path length difference can be computed as

$$\Delta\rho_{k,n} = -\vec{u}_n \cdot \vec{r}_k. \tag{12.2}$$

Since the reference point-related signal $\underline{s}_{\mathrm{RP},n}$ is a narrowband signal, the signal $\underline{s}_{k,n}$ received at antenna element $k$ is a phase shifted version of the signal $\underline{s}_{\mathrm{RP},n}$ [22]. This phase shift is

$$\Delta\phi_{k,n} = -2\pi\frac{\Delta\rho_{k,n}}{\lambda} = 2\pi\frac{\vec{u}_n \cdot \vec{r}_k}{\lambda} = \vec{\beta}_n \cdot \vec{r}_k. \tag{12.3}$$

The signal $\underline{s}_{k,n}$ received at antenna element $k$ expressed in terms of the signal $\underline{s}_{\mathrm{RP},n}$ that would be received at the reference point is

$$\underline{s}_{k,n} = \underline{s}_{\mathrm{RP},n}\mathrm{e}^{\mathrm{j}\Delta\phi_{k,n}} = \underline{s}_{\mathrm{RP},n}\mathrm{e}^{\mathrm{j}\vec{\beta}_n\cdot\vec{r}_k}. \tag{12.4}$$

The exponentional term $\mathrm{e}^{\mathrm{j}\vec{\beta}_n\cdot\vec{r}_k}$ is called the steering factor.

In the case where there are $N$ different targets $n = 1, \ldots, N$, the signal $\underline{s}_k$ being received at the $k$-th antenna element is a superposition of the phase shifted versions of all $N$ signals $\underline{s}_{\mathrm{RP},n}$, $n = 1, \ldots, N$. Furthermore, the signal $\underline{s}_k$ received at antenna element $k$ are corrupted by noise $\underline{n}_k$ with zero mean. The signal $\underline{s}_k$ received at the antenna element $k$ can thus be expressed with a summation as

$$\underline{s}_k = \sum_{n=1}^{N} \underline{s}_{\mathrm{RP},n}\mathrm{e}^{\mathrm{j}\vec{\beta}_n\cdot\vec{r}_k} + \underline{n}_k. \tag{12.5}$$

Since the targets in ARI are fluctuating, it is common practice to look at the statistics of the received signals rather than their particular realizations [46]. The spatial correlation of the signals $\underline{s}_k$ and $\underline{s}_l$ received at antenna elements $k$ and $l$ also known as the visibility in radio astronomy [46] is

$$\underline{v}_{k,l} = \mathrm{E}\{\underline{s}_k\underline{s}_l^*\} = \sum_{n=1}^{N}\sum_{m=1}^{N}\mathrm{E}\{\underline{s}_{\mathrm{RP},n}\underline{s}_{\mathrm{RP},m}^*\}\mathrm{e}^{\mathrm{j}(\vec{\beta}_n\cdot\vec{r}_k - \vec{\beta}_m\cdot\vec{r}_l)} + \mathrm{E}\{\underline{n}_k\underline{n}_l^*\}. \tag{12.6}$$

The signals $\underline{s}_{\mathrm{RP},n}$ and $\underline{s}_{\mathrm{RP},m}$ from different targets $n \neq m$ are assumed to be uncorrelated. Therefore, the term $\mathrm{E}\{\underline{s}_{\mathrm{RP},n}\underline{s}_{\mathrm{RP},m}^*\}$ is zero for $n \neq m$. Furthermore,

$$\mathrm{E}\{\underline{s}_{\mathrm{RP},n}\underline{s}_{\mathrm{RP},n}^*\} = \mathrm{E}\{|\underline{s}_{\mathrm{RP},n}|^2\} = b_n \tag{12.7}$$

is commonly referred to as the brightness in radio astronomy [46].

The noise $\underline{n}_k$ and $\underline{n}_l$ at different antenna elements $k \neq l$ is assumed to be uncorrelated. Therefore, the term $\mathrm{E}\{\underline{n}_k \underline{n}_l^*\}$ is zero for $k \neq l$. For $k = l$, one obtains the variance

$$\mathrm{E}\{\underline{n}_k \underline{n}_k^*\} = \mathrm{E}\{ \mid \underline{n}_k \mid^2 \} = \sigma^2. \tag{12.8}$$

Using

$$\Delta\vec{r}_{k,l} = \vec{r}_k - \vec{r}_l \tag{12.9}$$

and the brightnesses $b_n$, the visibility $\underline{v}_{k,l}$ can be rewritten as

$$\underline{v}_{k,l} = \begin{cases} \sum_{n=1}^{N} b_n \mathrm{e}^{\mathrm{j}\vec{\beta}_n \cdot \Delta\vec{r}_{k,l}}, & k \neq l \\ \sum_{n=1}^{N} b_n \mathrm{e}^{\mathrm{j}\vec{\beta}_n \cdot \Delta\vec{r}_{k,k}} + \sigma^2 = \sum_{n=1}^{N} b_n + \sigma^2, & k = l. \end{cases} \tag{12.10}$$

The co-array is defined as the auto-correlation of the antenna array [18]. For the co-array as well as Eq. (12.10), only the displacements $\Delta\vec{r}_{k,l}$ between antenna elements matter. However, depending on the positions of the antenna elements in space, several pairs of antenna elements may result in the same displacements thereby creating a redundancy. Figure 12.2(left) shows an array of antenna elements arranged in an asymmetrical cross used for meteor studies in [17]. The corresponding co-array is shown in Fig. 12.2(right). The large circle at the center of the array shows the redundancy in the co-array where five pairs of antenna elements resulted in the same displacement.

The visibilities obtained from the $K$ antenna elements can be combined in a visibility vector.

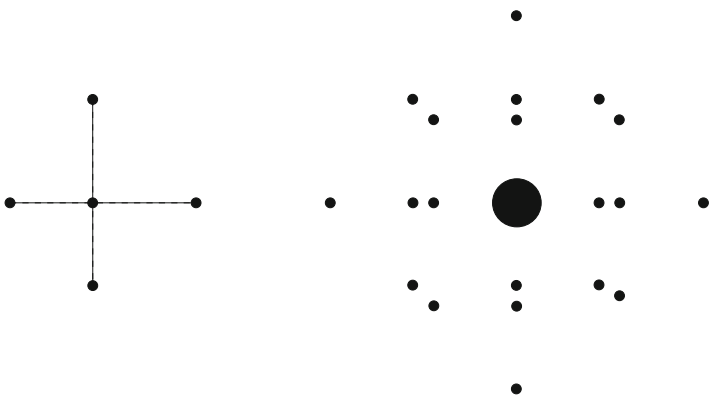


**Fig. 12.2** An asymmetrical cross array with $K = 5$ antenna elements (left) [17] and its corresponding co-array (right)

Representing the summation as a matrix vector product yields the linear system of equations

$$\begin{pmatrix} \underline{v}_{1,1} \\ \vdots \\ \underline{v}_{K,1} \\ \vdots \\ \underline{v}_{1,K} \\ \vdots \\ \underline{v}_{K,K} \end{pmatrix} = \begin{pmatrix} 1 & \cdots & 1 \\ \vdots & & \vdots \\ \mathrm{e}^{\mathrm{j}\vec{\beta}_1 \cdot \Delta\vec{r}_{K,1}} & \cdots & \mathrm{e}^{\mathrm{j}\vec{\beta}_N \cdot \Delta\vec{r}_{K,1}} \\ \vdots & & \vdots \\ \mathrm{e}^{\mathrm{j}\vec{\beta}_1 \cdot \Delta\vec{r}_{1,K}} & \cdots & \mathrm{e}^{\mathrm{j}\vec{\beta}_N \cdot \Delta\vec{r}_{1,K}} \\ \vdots & & \vdots \\ 1 & \cdots & 1 \end{pmatrix} \cdot \begin{pmatrix} b_1 \\ \vdots \\ b_N \end{pmatrix} + \begin{pmatrix} \sigma^2 \\ \vdots \\ 0 \\ \vdots \\ 0 \\ \vdots \\ \sigma^2 \end{pmatrix} \tag{12.11}$$

with $K^2$ equations for $N$ brightnesses $b_n$, $n = 1, \ldots, N$. The matrix with the exponentionals is called the sensing matrix.

In reality, $I$ measurements of the received signals are taken at the antenna elements $k$ and $l$. The $i$-th measured received signals at antenna elements $k$ and $l$ are denoted by $\underline{s}_k^{(i)}$ and $\underline{s}_l^{(i)}$, respectively. From these $I$ measurements, an estimate of the visibility is computed as

$$\hat{\underline{v}}_{k,l} = \frac{1}{I} \sum_{i=1}^{I} \underline{s}_k^{(i)} \underline{s}_l^{*(i)}. \tag{12.12}$$

The brightnesses $b_n$ to be estimated are real-valued, see Eq. (12.7). To ensure that the inversion algorithm used for estimating the brightnesses always returns a real-valued solution, it might be useful to reformulate the system model of (12.11) into an equivalent purely real-valued system model. This can be achieved by taking linear combinations of the visibility as in

$$\frac{\underline{v}_{k,l} + \underline{v}_{l,k}}{2} = \begin{cases} \sum_{n=1}^{N} b_n \cos\left(\vec{\beta}_n \cdot \Delta\vec{r}_{k,l}\right), & k \neq l \\ \sum_{n=1}^{N} b_n + \sigma^2, & k = l \end{cases} \tag{12.13}$$

and

$$\frac{\underline{v}_{k,l} - \underline{v}_{l,k}}{2\mathrm{j}} = \sum_{n=1}^{N} b_n \sin\left(\vec{\beta}_n \cdot \Delta\vec{r}_{k,l}\right). \tag{12.14}$$

Using the linear combinations of the visibilities, one can define a vector $\mathbf{v}$ with $K^2$ elements. The $m$-th element of the vector $\mathbf{v}$ is

$$[\mathbf{v}]_m = \begin{cases} \frac{\underline{v}_{k,l}+\underline{v}_{l,k}}{2} = \sum_{n=1}^{N} b_n \cos\left(\vec{\beta}_n \cdot \Delta\vec{r}_{k,l}\right), & m = 2k-1+(l-1)^2,\ k<l \\ \frac{\underline{v}_{k,l}-\underline{v}_{l,k}}{2\mathrm{j}} = \sum_{n=1}^{N} b_n \sin\left(\vec{\beta}_n \cdot \Delta\vec{r}_{k,l}\right), & m = 2k+(l-1)^2,\ k<l \\ \underline{v}_{k,k} - \sigma^2 = \sum_{n=1}^{N} b_n, & m = k^2. \end{cases} \tag{12.15}$$

Furthermore, defining a $K^2 \times N$ matrix $\mathbf{A}$ with the elements

$$[\mathbf{A}]_{m,n} = \begin{cases} \cos\left(\vec{\beta}_n \cdot \Delta\vec{r}_{k,l}\right), & m = 2k-1+(l-1)^2,\ k<l \\ \sin\left(\vec{\beta}_n \cdot \Delta\vec{r}_{k,l}\right), & m = 2k+(l-1)^2,\ k<l \\ 1, & m = k^2 \end{cases} \tag{12.16}$$

and a brightness vector $\mathbf{b}$ with $N$ elements

$$[\mathbf{b}]_n = b_n, \tag{12.17}$$

one obtains a purely real-valued linear system of equations

$$\mathbf{v} = \mathbf{A} \cdot \mathbf{b}. \tag{12.18}$$

Furthermore, from the definition of (12.7), it is clear that the brightnesses $b_n$ are nonnegative. This non-negativity has to be considered when designing an appropriate inversion algorithm.

### 12.2.2 System Model for MIMO Atmospheric Radar Imaging

Figure 12.3 shows the system model with transmitter and receiver side reference points as well as a transmit antenna element $p$ with position vector $\vec{r}_p$ and a receive antenna element $k$ with position vector $\vec{r}_k$. When transmitting a signal from a transmit antenna element at the transmitter side reference point, a narrowband signal with the complex amplitude $\underline{s}_{\mathrm{RP},n}$ being backscattered from a target $n$ in the far-field would be received at the receiver side reference point. The direction of departure of the signal transmitted from the transmitter side reference point would in general be different from the direction of arrival of the signal received at the receiver side reference point due to the transmit antenna array and the receive antenna array being at different places. The wavenumber vectors $\vec{\beta}_{\mathrm{TX},n}$ and $\vec{\beta}_{\mathrm{RX},n}$ characterizing the direction of departure and the direction of arrival, respectively, are as described in Eq. (12.1).

Since the reference point-related signal $\underline{s}_{\mathrm{RP},n}$ is a narrowband signal, the signal $\underline{s}_{p,k,n}$ received at antenna element $k$ due to the transmission from antenna element $p$ is a phase shifted version of the signal $\underline{s}_{\mathrm{RP},n}$. This phase shift is

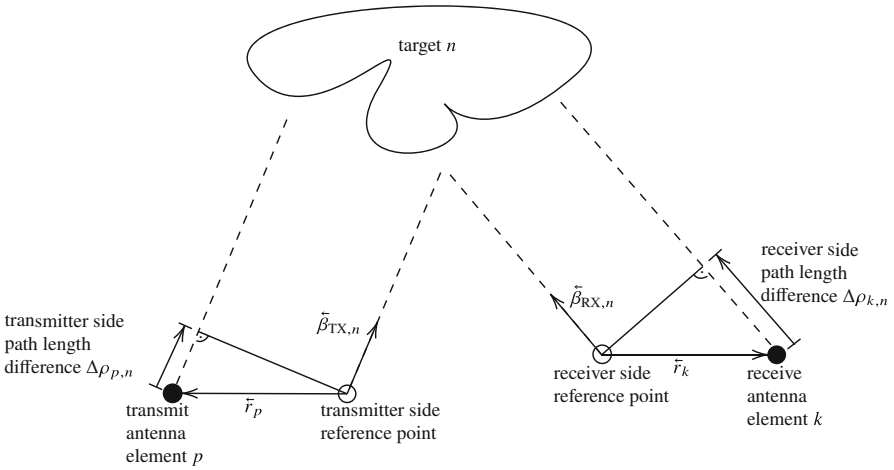



**Fig. 12.3** Model for MIMO atmospheric radar imaging

$$\Delta\phi_{p,k,n} = \vec{\beta}_{\mathrm{TX},n} \cdot \vec{r}_p + \vec{\beta}_{\mathrm{RX},n} \cdot \vec{r}_k, \tag{12.19}$$

see (12.3).

Therefore, the signal $\underline{s}_{p,k,n}$ received at the $k$-th antenna element due to the $p$-th antenna element illuminating the target $n$ is

$$\underline{s}_{p,k,n} = \underline{s}_{\mathrm{RP},n} \mathrm{e}^{\mathrm{j}\Delta\phi_{p,k,n}} = \underline{s}_{\mathrm{RP},n} \mathrm{e}^{\mathrm{j}(\vec{\beta}_{\mathrm{TX},n}\cdot\vec{r}_p + \vec{\beta}_{\mathrm{RX},n}\cdot\vec{r}_k)}. \tag{12.20}$$

In the case that $N$ different targets $n = 1, \ldots, N$ are present, the signal $\underline{s}_{p,k}$ received at antenna element $k$ when transmitting from antenna element $p$ taking the zero mean noise $\underline{n}_k$ into consideration is

$$\underline{s}_{p,k} = \sum_{n=1}^{N} \underline{s}_{\mathrm{RP},n} \mathrm{e}^{\mathrm{j}(\vec{\beta}_{\mathrm{TX},n}\cdot\vec{r}_p + \vec{\beta}_{\mathrm{RX},n}\cdot\vec{r}_k)} + \underline{n}_k. \tag{12.21}$$

Taking the spatial correlation of the signal $\underline{s}_{p,k}$ received at antenna element $k$ when transmitting from antenna element $p$ and the signal $\underline{s}_{q,l}$ received at antenna element $l$ when transmitting from antenna element $q$ yields the visibility

$$\begin{aligned}
\underline{v}_{p,q,k,l} &= \mathrm{E}\left\{\underline{s}_{p,k}\underline{s}^*_{q,l}\right\} \\
&= \sum_{n=1}^{N}\sum_{m=1}^{N} \mathrm{E}\left\{\underline{s}_{\mathrm{RP},n}\underline{s}^*_{\mathrm{RP},m}\right\} \mathrm{e}^{\mathrm{j}(\vec{\beta}_{\mathrm{TX},n}\cdot\vec{r}_p + \vec{\beta}_{\mathrm{RX},n}\cdot\vec{r}_k - \vec{\beta}_{\mathrm{TX},m}\cdot\vec{r}_q - \vec{\beta}_{\mathrm{RX},m}\cdot\vec{r}_l)} \\
&\quad + \mathrm{E}\left\{\underline{n}_k\underline{n}^*_l\right\}.
\end{aligned} \tag{12.22}$$

Using the same assumptions and definitions as in the SIMO case, one can write

$$\underline{v}_{p,q,k,l} = \begin{cases} \sum_{n=1}^{N} b_n \mathrm{e}^{\mathrm{j}(\vec{\beta}_{\mathrm{TX},n}\cdot\Delta\vec{r}_{p,q}+\vec{\beta}_{\mathrm{RX},n}\cdot\Delta\vec{r}_{k,l})}, & k \neq l \\ \sum_{n=1}^{N} b_n \mathrm{e}^{\mathrm{j}(\vec{\beta}_{\mathrm{TX},n}\cdot\Delta\vec{r}_{p,q}+\vec{\beta}_{\mathrm{RX},n}\cdot\Delta\vec{r}_{k,k})} + \sigma^2, & p \neq q, k = l \\ \sum_{n=1}^{N} b_n + \sigma^2, & p = q, k = l. \end{cases} \tag{12.23}$$

In the following, the special case that the transmit and receive antenna elements are collocated such that the direction of departure and arrival is the same as depicted in Fig. 12.4 will be considered. For such a collocated MIMO radar, the transmit and receive signals are characterized by the same wavenumber vector $\vec{\beta}_{\mathrm{TX},n} = \vec{\beta}_{\mathrm{RX},n} = \vec{\beta}_n$. The signal received at antenna element $k$ due to the transmission from antenna element $p$ in the case of collocated MIMO radar is

$$\underline{s}_{p,k} = \sum_{n=1}^{N} \underline{s}_{\mathrm{RP},n} \mathrm{e}^{\mathrm{j}\vec{\beta}_n\cdot(\vec{r}_p+\vec{r}_k)} + \underline{n}_k. \tag{12.24}$$

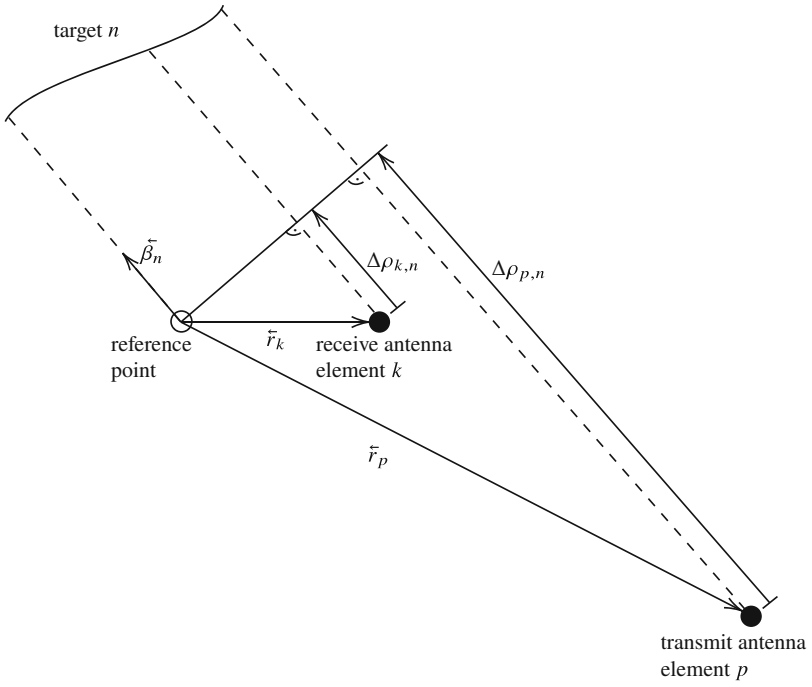



**Fig. 12.4** Model for Collocated MIMO atmospheric radar imaging

The use of additional transmit antenna elements results in signals being received from several transmit–receive paths. When $P$ transmit and $K$ receive antenna elements are used, the resulting number of transmit–receive paths is $PK$. The $PK$ transmit–receive paths are equivalent to using $PK$ virtual receive antenna elements called a virtual array. The virtual array is the convolution of the transmit antenna array and the receive antenna array where only the sums of the transmit and receive antenna elements' positions $\vec{r}_p+\vec{r}_k$ matter. Depending on the position of the transmit and receive antenna elements, some of the measurements by the virtual array result in the same displacement thereby leading to a redundancy. The steering matrix of the virtual array is basically the Kronecker product of the transmitter side and receiver side steering matrices [27]. The elements of the steering matrix are composed of the steering factors. Figure 12.5 depicts the concept of the virtual array. It shows a MIMO configuration with two transmit antenna elements and three receive antenna elements. The first row shows the MIMO radar with two Tx and three Rx antennas. The second and third rows show the layout where the received signal is due to the transmission from only one Tx. The resulting virtual array on the fourth row with six receive antenna elements is due to the three receive antenna elements receiving signals as a result of the transmission from the two transmit antenna elements, i.e., the bottom row is a combination of the two layouts above it.

The visibility $\underline{v}_{p,q,k,l}$ for the collocated MIMO radar case simplifies to

$$\underline{v}_{p,q,k,l} = \begin{cases} \sum_{n=1}^{N} b_n \mathrm{e}^{\mathrm{j}\vec{\beta}_n\cdot(\Delta\vec{r}_{p,q}+\Delta\vec{r}_{k,l})}, & k \neq l \\ \sum_{n=1}^{N} b_n \mathrm{e}^{\mathrm{j}\vec{\beta}_n\cdot(\Delta\vec{r}_{p,q}+\Delta\vec{r}_{k,k})} + \sigma^2, & p \neq q, k = l \\ \sum_{n=1}^{N} b_n + \sigma^2, & p = q, k = l. \end{cases} \tag{12.25}$$

Defining

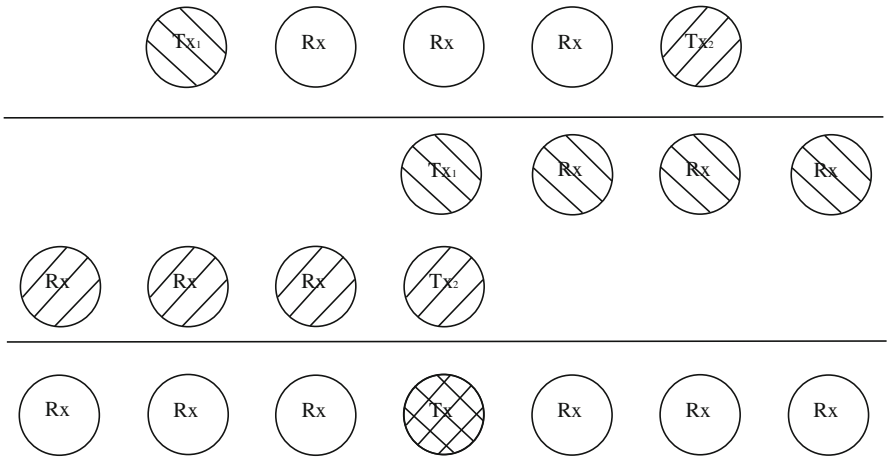



**Fig. 12.5** A MIMO radar (top) and its resulting virtual array (bottom)

$$\Delta\vec{r}_{p,q,k,l} = \Delta\vec{r}_{p,q} + \Delta\vec{r}_{k,l}, \tag{12.26}$$

the visibility $\underline{v}_{p,q,k,l}$ can be rewritten as

$$\underline{v}_{p,q,k,l} = \begin{cases} \sum_{n=1}^{N} b_n \mathrm{e}^{\mathrm{j}\vec{\beta}_n \cdot \Delta\vec{r}_{p,q,k,l}}, & k \neq l \\ \sum_{n=1}^{N} b_n \mathrm{e}^{\mathrm{j}\vec{\beta}_n \cdot \Delta\vec{r}_{p,q,k,k}} + \sigma^2, & p \neq q, k = l \\ \sum_{n=1}^{N} b_n + \sigma^2, & p = q, k = l. \end{cases} \tag{12.27}$$

Similar to the co-arrays in the SIMO case, the correlation of the virtual array depends only on the sum of the displacements between the transmit antenna elements and the displacements between the receive antenna elements $\Delta\vec{r}_{p,q,k,l} = \Delta\vec{r}_{p,q} + \Delta\vec{r}_{k,l}$.

It is recommended to rewrite the system model as a purely real-valued system model as it was done in the SIMO case. This can be achieved by taking linear combinations of the visibilities $\underline{v}_{p,q,k,l}$ as in

$$\frac{\underline{v}_{p,q,k,l} + \underline{v}_{q,p,l,k}}{2} = \begin{cases} \sum_{n=1}^{N} b_n \cos\left(\vec{\beta}_n \cdot \Delta\vec{r}_{p,q,k,l}\right), & k \neq l \\ \sum_{n=1}^{N} b_n \cos\left(\vec{\beta}_n \cdot \Delta\vec{r}_{p,q,k,k}\right) + \sigma^2, & p \neq q, k = l \\ \sum_{n=1}^{N} b_n + \sigma^2, & p = q, k = l \end{cases} \tag{12.28}$$

and

$$\frac{\underline{v}_{p,q,k,l} - \underline{v}_{q,p,l,k}}{2j} = \sum_{n=1}^{N} b_n \sin\left(\vec{\beta}_n \cdot \Delta\vec{r}_{p,q,k,l}\right). \tag{12.29}$$

Using the linear combinations of the visibilities, one can define a block vector

$$\mathbf{v} = \begin{pmatrix} \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_r \\ \vdots \\ \mathbf{v}_R \end{pmatrix} \tag{12.30}$$

with

$$R = \frac{P(P+1)}{2} \tag{12.31}$$

blocks $\mathbf{v}_r$, where $P$ is the number of transmit antenna elements and

$$r = p + \frac{q(q-1)}{2},\ p \le q. \tag{12.32}$$

The $m$-th element of the $r$-th block is

$$[\mathbf{v}_r]_m = \begin{cases} \frac{\underline{v}_{p,p,k,l} + \underline{v}_{p,p,l,k}}{2}, & m = 2k - 1 + (l-1)^2,\ p = q,\ k < l \\ \frac{\underline{v}_{p,p,k,l} - \underline{v}_{p,p,l,k}}{2\mathrm{j}}, & m = 2k + (l-1)^2,\ p = q,\ k < l \\ \underline{v}_{p,p,k,k} - \sigma^2, & m = k^2,\ p = q \\ \frac{\underline{v}_{p,q,k,l} + \underline{v}_{q,p,l,k}}{2}, & m = 2k - 1 + 2K(l-1),\ p < q,\ k \neq l \\ \frac{\underline{v}_{p,q,k,l} - \underline{v}_{q,p,l,k}}{2j}, & m = 2k + 2K(l-1),\ p < q,\ k \neq l \\ \frac{\underline{v}_{p,q,k,k} + \underline{v}_{q,p,k,k}}{2} - \sigma^2, & m = 2k - 1 + 2K(l-1),\ p < q,\ k = l \\ \frac{\underline{v}_{p,q,k,k} - \underline{v}_{q,p,k,k}}{2j}, & m = 2k + 2K(l-1),\ p < q,\ k = l, \end{cases} \tag{12.33}$$

where $K$ is the number of receive antenna elements. Similarly, a block matrix

$$\mathbf{A} = \begin{pmatrix} \mathbf{A}_1 \\ \vdots \\ \mathbf{A}_r \\ \vdots \\ \mathbf{A}_R \end{pmatrix} \tag{12.34}$$

with $R$ blocks can be defined. The $m$-th row and $n$-th column element of the $r$-th block is

$$[\mathbf{A}_r]_{m,n} = \begin{cases} \cos\left(\vec{\beta}_n \cdot \Delta\vec{r}_{p,p,k,l}\right), & m = 2k - 1 + (l-1)^2,\ p = q,\ k < l \\ \sin\left(\vec{\beta}_n \cdot \Delta\vec{r}_{p,p,k,l}\right), & m = 2k + (l-1)^2,\ p = q,\ k < l \\ 1, & m = k^2,\ p = q \\ \cos\left(\vec{\beta}_n \cdot \Delta\vec{r}_{p,q,k,l}\right), & m = 2k - 1 + 2K(l-1),\ p < q,\ k \neq l \\ \sin\left(\vec{\beta}_n \cdot \Delta\vec{r}_{p,q,k,l}\right), & m = 2k + 2K(l-1),\ p < q,\ k \neq l \\ \cos\left(\vec{\beta}_n \cdot \Delta\vec{r}_{p,q,k,k}\right), & m = 2k - 1 + 2K(l-1),\ p < q,\ k = l \\ \sin\left(\vec{\beta}_n \cdot \Delta\vec{r}_{p,q,k,k}\right), & m = 2k + 2K(l-1),\ p < q,\ k = l. \end{cases} \tag{12.35}$$

Finally, a brightness vector $\mathbf{b}$ with $N$ elements

$$[\mathbf{b}]_n = b_n \tag{12.36}$$

can be defined. One obtains a purely real-valued linear system of equations

$$\mathbf{v} = \mathbf{A} \cdot \mathbf{b} \tag{12.37}$$

as in the SIMO case.

ARI is a spectral analysis problem as the visibility and the brightness are a Fourier pair.

### 12.2.3 SIMO vs MIMO Arrays

Figure 12.6a depicts a uniform linear antenna array consisting of two transmit antenna elements and six receive antenna elements. As mentioned in the previous section, the visibility $\underline{v}_{p,q,k,l}$ depends only on the displacements $\Delta\vec{r}_{p,q,k,l} = \Delta\vec{r}_{p,q} + \Delta\vec{r}_{k,l}$ for the MIMO case and $\Delta\vec{r}_{k,l}$ for the SIMO case. These displacements are also known as baselines. The baselines for the SIMO case considering only one transmit antenna element can be seen in Fig. 12.6b. Since the array is uniform, some of the baselines are identical and are therefore redundant. The number of redundant baseline is color coded in Fig. 12.6b and e.

The point spread or instrument function that is the response of an antenna array to a punctual target $b_0$ at $\vec{\beta}_n = \vec{\beta}_0$ is used to characterize the performance of the antenna array. The visibility for this condition reduces to

$$\underline{v}_m = b_0 \mathrm{e}^{\mathrm{j}\vec{\beta}_0 \cdot \Delta\vec{r}_m}. \tag{12.38}$$

Taking the inverse Fourier transform of the visibility $\underline{v}_m$ yields the point spread function

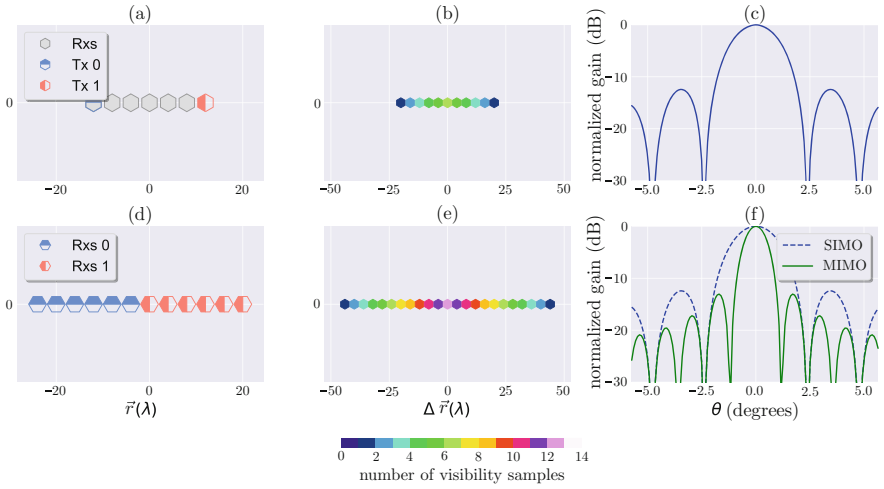



**Fig. 12.6** Antenna positions and array patterns. (**a**) SIMO: antenna array. (**b**) SIMO: visibility (1 Tx). (**c**) SIMO: instrument function. (**d**) MIMO: virtual antenna array. (**e**) MIMO: visibility (2 Txs). (**f**) MIMO: instrument function

$$\hat{b}(\vec{\beta}-\vec{\beta}_0)=\sum_{m=1}^{PK}\underline{v}_m e^{-j\vec{\beta}\cdot\Delta\vec{r}_m}=b_0\sum_{m=1}^{PK}e^{j\vec{\beta}_0\cdot\Delta\vec{r}_m}e^{-j\vec{\beta}\cdot\Delta\vec{r}_m}=b_0\sum_{m=1}^{PK}e^{-j(\vec{\beta}-\vec{\beta}_0)\cdot\Delta\vec{r}_m}. \tag{12.39}$$

The point spread function $\hat{b}(\vec{\beta}-\vec{\beta}_0)$ characterizes the angular resolution of an antenna array. The ideal instrument function of a Dirac delta function is desired for an antenna array. However, the limited number of visibilities for conventional arrays results in its finite angular resolution. The instrument function for the SIMO case considering $\vec{\beta}_0 = 0$ is depicted in Fig. 12.6c, which is not close to a Dirac delta function. The mainlobe has a half-power beam width (HPBW) of 2° and the sidelobes have a normalized gain of −14 dB.

The angular resolution $\Delta\theta$ and the maximum unambiguous angle $\theta_{\max}$ for a uniform linear array with $\vec{\beta}=\frac{2\pi}{\lambda}\sin(\theta)$ are computed as

$$\sin(\Delta\theta)=\frac{\lambda}{(\Delta\vec{r}_{\max})} \tag{12.40}$$

and

$$\sin(\theta_{\max})=\frac{\lambda}{2(\Delta\vec{r}_{\min})}, \tag{12.41}$$

where $\theta$ is the elevation angle in the direction of the target and $\Delta\vec{r}_{\max}$ and $\Delta\vec{r}_{\min}$ are the maximum and minimum separation between two baselines, respectively. Figure 12.6d shows the MIMO configuration with 12 virtual receive antenna elements forming the virtual array which is two times larger than in the SIMO case with six receive antenna elements. The resulting angular resolution is also two times better as shown in Fig. 12.6f. However, the strongest sidelobes still have a normalized gain of $\approx -14$ dB as in the SIMO case.

### *12.2.4 Inversion Methods*

The systems of linear Eqs. (12.37) and (12.18) are underdetermined systems with infinitely many solutions matching to the measured visibilities $\mathbf{v}$. Therefore, the radar imaging task is to select the solution that represents the most probable brightnesses $\hat{\mathbf{b}}$ from the set of possible solutions.

#### 12.2.4.1 The Capon Method

The Capon method is an adaptive technique proposed by Palmer et al. [29] for solving the radar imaging problem based on the work of [4]. The method minimizes the sidelobe interference by choosing the weights at each direction of arrival

adaptively. It can be seen as an extension of the beam steering approach. In order to use the Capon method, the visibilities $\mathbf{v}$ have to be rearranged in a matrix form $\mathbf{V}$. The element $[\mathbf{V}]_{k,l}$ on the $k$-th row and $l$-th column is $[\mathbf{V}]_{k,l} = \langle \underline{s}_k \underline{s}_l^* \rangle$, for $k = 1, \ldots, K$ and $l = 1, \ldots, K$, where $K$ is the total number of receive antenna elements. The estimate of the brightness for the $n$-th target $[\hat{\mathbf{b}}]_n$ is obtained as

$$[\hat{\mathbf{b}}]_n = \frac{1}{\mathbf{a}_n^{\mathrm{H}} \cdot \mathbf{V}^{-1} \cdot \mathbf{a}_n}, \tag{12.42}$$

where $\mathbf{a}_n$ is the $n$-th steering vector and $\mathbf{a}_n^{\mathrm{H}}$ being the conjugate transpose of $\mathbf{a}_n$. Equation (12.42) is solved for all $N$ targets to get the complete estimate of the brightness vector $\hat{\mathbf{b}}$.

#### 12.2.4.2 Maximum Entropy Method

The maximum entropy (MaxEnt) method chooses the brightness $\hat{\mathbf{b}}$ which maximizes the entropy while being consistent with the measured visibilities $\mathbf{v}$ [45]. The entropy as a measure of the probability is defined as

$$\mathrm{H}(b_1, \ldots, b_N) = -\sum_{n=1}^{N} b_n \ln b_n. \tag{12.43}$$

The brightness $\hat{\mathbf{b}}$ that maximizes the entropy and is consistent with the measured visibilities is computed as

$$\hat{\mathbf{b}} = \underset{b_1,\ldots,b_N}{\arg\max}\{\mathrm{H}(b_1, \ldots, b_N)\}, \quad \text{s.t.} \begin{cases} B = \sum_{n=1}^{N} b_n, \\ \|\mathbf{v} - \mathbf{A} \cdot \mathbf{b}\|_2^2 = 0. \end{cases} \tag{12.44}$$

The maximization of the entropy results in a nonlinear problem which can be solved numerically using the hybrid method described in [31]. Additionally, the error covariance matrix can be used as a constraint in the optimization task to obtain a more detailed radar image as done in [20].

#### 12.2.4.3 Compressed Sensing

Compressed sensing (CS) formalizes the long known knowledge that fewer measurements than required by the famous Shannon–Nyquist sampling theorem are needed for the exact recovery of signals that do not completely occupy the spectrum. It formalizes this knowledge by requiring that the signal be sparse in some known domain. While images are not sparse in their original domain, they have been shown to be sparse in the Fourier and wavelet domain by the work of [36, 38, 47]. Curvelets

[33], bandlets [26], and adaptive dictionaries [30] have been proposed in an attempt to improve the sparsity of complicated images. An arbitrary sparsity basis for the brightness $\mathbf{b}$ can be exploited by

$$\mathbf{b} = \boldsymbol{\Psi} \cdot \mathbf{f} \tag{12.45}$$

and

$$\mathbf{v} = \mathbf{A} \cdot \boldsymbol{\Psi} \cdot \mathbf{f}, \tag{12.46}$$

where $\boldsymbol{\Psi}$ is an $N \times N$ matrix defining the sparsity basis of the brightness $\mathbf{b}$ and $\mathbf{f}$ is the corresponding sparse vector of the brightness $\mathbf{b}$ in the arbitrary basis $\boldsymbol{\Psi}$.

Although Eq. (12.46) is still an underdetermined system of linear equations, the work of [2] shows that exact recovery of $\mathbf{f}$ is possible under two conditions. The first condition is that the sparsity vector $\mathbf{f}$ is $F$-sparse, i.e., it has at most $F$ nonzero elements with $F < N$, where $N$ is the number of elements in the brightness $\mathbf{b}$. Secondly, it is required that the sensing matrix $\mathbf{H} = \mathbf{A} \cdot \boldsymbol{\Psi}$ satisfies the restricted isometric property (RIP), which requires that any $F$ columns of the sensing matrix $\mathbf{H}$ be approximately orthogonal. A naive estimate of the sparsity vector $\hat{\mathbf{f}}$ can be computed from noisy measured visibilities $\mathbf{v}$ as

$$\hat{\mathbf{f}} = \arg\min_{\mathbf{f}} \{\|\mathbf{f}\|_0\}, \quad \text{s.t. } \|\mathbf{v} - \mathbf{A} \cdot \boldsymbol{\Psi} \cdot \mathbf{f}\|_2^2 < \sigma^2, \tag{12.47}$$

where the "L0-norm" $\|\mathbf{f}\|_0$ is the number of nonzero elements in the sparsity vector $\mathbf{f}$. Although the "L0-norm" minimization yields the sparsest version of $\mathbf{f}$ that agrees with the measured data, it is unfortunately non-convex and difficult to solve for most problems.

For a sensing matrix $\mathbf{H}$ that satisfies the RIP condition, results from [3] and [13] show that the "L0-norm" minimization is equivalent to the more computationally attractive "L1-norm" minimization such that

$$\hat{\mathbf{f}} = \arg\min_{\mathbf{f}} \{\|\mathbf{f}\|_1\}, \quad \text{s.t. } \|\mathbf{v} - \mathbf{A} \cdot \boldsymbol{\Psi} \cdot \mathbf{f}\|_2^2 < \sigma^2. \tag{12.48}$$

The "L1-norm" minimization also known as basis pursuit can be solved using linear programming. The estimated brightness $\hat{\mathbf{b}}$ can be computed from $\hat{\mathbf{f}}$ using Eq. (12.45).

### *12.2.5 MIMO Implementations*

It is necessary in a MIMO radar to separate the signals from different transmit antenna elements. When designing a transmit diversity scheme to generate transmit signals, the range and velocity of the target $n$ have to be considered so that the backscattered signals are still orthogonal upon reception. It should be noted that

designing a transmit diversity scheme where the orthogonality of the backscattered signals is not destroyed by delay or Doppler shift is rather a dream than a technical solution. In a more comprehensive sense, the correlation of the backscattered signals depends on the thickness of the target as well as the target's Doppler bandwidth. Consider a scenario where two signals are transmitted from two spatially separated transmit antenna elements and allowing a frequency separation of 1 MHz between the transmitted signals. While the two signals are orthogonal to each other considering ideal band-limited signals, their backscattered forms might not be orthogonal to each other depending on the nature of the target $n$. If the target $n$ is a hard target with a narrow Doppler bandwidth, for example, a specular meteor having a Doppler frequency of 100 Hz, the backscattered signals will still be orthogonal to each other. On the other hand, if the target is a volume with multiple scatters, each scatter with its own Doppler frequency greater than 500 kHz, the backscattered signals might not be orthogonal to each other.

It should be noted that no transmit diversity scheme ensures perfectly orthogonal receive signals. However, a transmit diversity scheme that ensures a low correlation among the backscattered signals is sufficient. Transmit diversity can be achieved by transmitting signals with different frequencies, at different times, or with different polarizations. Unfortunately, frequency and polarization transmit diversities are not suitable for ARI due to the frequency and polarization-dependent scattering nature of atmospheric targets. In an attempt to use frequency diversity, the frequency separation between the transmitted signals from two transmit antenna elements must be at least the bandwidth of the target. This bandwidth could be a few megahertz for atmospheric targets, and such large frequency separations are not suitable for most atmospheric targets due to their frequency-dependent scattering properties.

Time diversity, waveform diversity, and a suboptimal diversity scheme are presented in the following. The limitations as well as the suitability for specific radar targets are also described.

#### 12.2.5.1 Time Diversity

Time diversity involves transmitting the same waveform from all transmit antenna elements where each signal is transmitted after a time delay. For one transmitting antenna element and a desired maximum unambiguous range $d_{max}$, the pulse repetition interval (PRI) $T$ is computed as

$$T = \tau + \frac{2d_{max}}{c}, \tag{12.49}$$

where $\tau$ is the transmitted pulse width and $c$ being the speed of light. In the MIMO case where $P$ transmit antenna elements are present, every $p$-th pulse is used for a certain transmit antenna element. Therefore, the MIMO PRI $T_{\mathrm{MIMO}}$ is

$$T_{\mathrm{MIMO}} = PT. \tag{12.50}$$

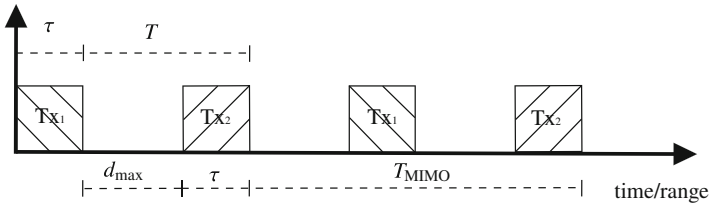



**Fig. 12.7** Time diagram of a pulsed MIMO radar with two transmitters using time diversity

Such a time diversity scheme with two transmitters $P = 2$ is depicted in Fig. 12.7.

As seen in Eq. (12.50), the larger the total number of transmit antenna elements $P$, the longer the MIMO PRI $T_{\mathrm{MIMO}}$, resulting in a reduced transmit energy per antenna element. Therefore, time diversity is only suitable for radar targets with high signal-to-noise ratios and long correlation times.

Separating the Doppler and range processing in a rather suboptimal trivial way as against a joint Doppler and range processing considering the MIMO ambiguity function as done in, e.g., [32] and [10], the maximum unambiguous Doppler frequency $w_{\max}$ is computed as

$$w_{\max} = \frac{\pi}{T_{\mathrm{MIMO}}} = \frac{\pi}{PT}. \tag{12.51}$$

Therefore, the Doppler bandwidth of a MIMO radar with $P$ transmit antenna elements is $P$ times smaller than the Doppler bandwidth of a radar with $P = 1$ transmit antenna element.

In order to keep the loss in Doppler bandwidth and average energy low, time diversity should be used only for targets with short ranges.

#### 12.2.5.2 Waveform Diversity

Waveform diversity uses codes that are almost uncorrelated with their shifted versions to generate quasi-orthogonal transmit signals. Waveform diversity can be applied to pulsed radar, which can be considered as a special case of continuous wave radar with most of the code bits being zero. However, short codes exhibit a high cross-correlation; thus, using only waveform diversity is not recommended for pulsed radars. For long pulses, there are several known codes, some of which include Gold codes [15], Walsh–Hadamard codes [16], polyphase codes [11, 14], and pseudo-random binary codes [28]. The choice of one of the codes depends on the hardware capabilities. Figure 12.8 shows an example of such a diversity scheme with each transmit antenna element simultaneously transmitting a coded waveform for the entire transmission duration. All transmit antenna elements simultaneously transmitting their waveforms for the entire duration mitigate against the loss of average energy per antenna element experienced in time diversity. The signal for

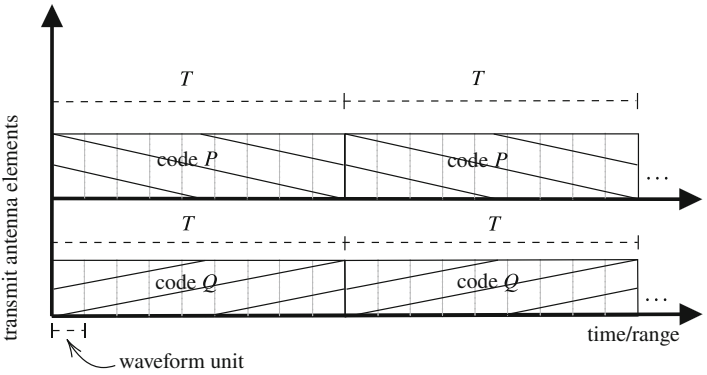



**Fig. 12.8** Time diagram of a continuous wave MIMO radar with two transmitters using waveform diversity

each transmit–receive link can be recovered with inverse methods as long as the code sequence is known. A thorough analysis of the signals using waveform diversity is done in [39].

An analysis of the auto- and cross-correlation properties of two coded waveforms $w_P$ and $w_Q$ is carried out in the following. The auto-correlation $C_{P,P}(d_i, d_j)$ of waveform $w_P$ at different range lags $d_n - d_i$ is defined as

$$C_{P,P}(d_i, d_j) = \sum_{n=1}^{N} w_P(d_n - d_i) w_P^*(d_n - d_j). \tag{12.52}$$

Similarly, the cross-correlation of two waveforms $w_P$ and $w_Q$ is defined as

$$C_{P,Q}(d_i, d_j) = \sum_{n=1}^{N} w_P(d_n - d_i) w_Q^*(d_n - d_j). \tag{12.53}$$

Equation (12.53) shows the interference between received signals from different ranges, due to a high range sidelobe. Figure 12.9 depicts the normalized auto- and cross-correlation functions for two pseudo-random binary codes $P$ and $Q$ of length 50. The correlation values are normalized to the code length. The auto-correlation for $d_i = d_j$ (mainlobe) is 1 as desired, while the correlation for the sidelobes $d_i \neq d_j$ is 0.12. The range peak-to-sidelobe ratio (PSLR) is thus $\approx 10$. The range PSLR can be improved by using longer codes.

To efficiently use the available bandwidth and time, the number of orthogonal codes should be approximately equal to the time–bandwidth product. Unfortunately, there is a limited number of known codes for many types of codes.

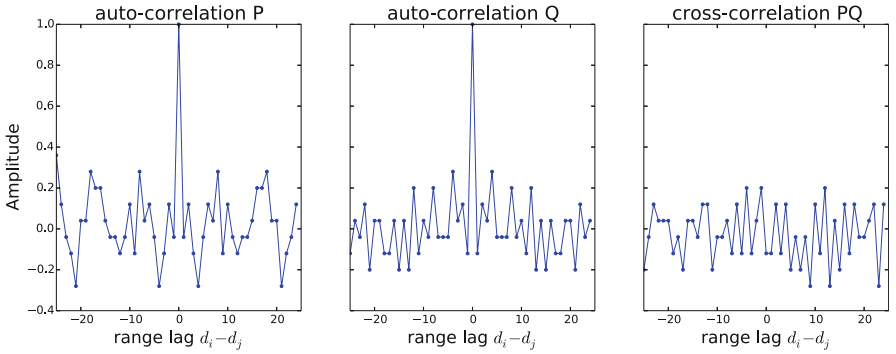



**Fig. 12.9** Auto- and cross-correlation functions of two waveforms $P$ and $Q$ of length 50

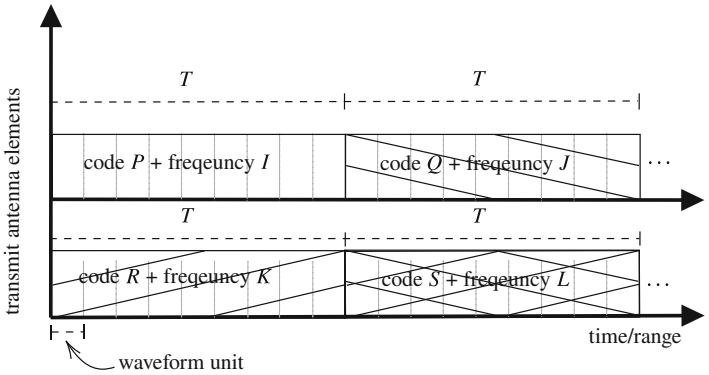



**Fig. 12.10** Time diagram of a continuous wave MIMO radar using suboptimal diversity

### 12.2.5.3 Suboptimal Diversity

A suboptimal diversity scheme is illustrated in Fig. 12.10 where code and frequency diversity are combined to generate the transmit signals. An optimal diversity scheme would combine the advantages of all transmit diversity schemes and minimize their disadvantages. Theoretically, an optimal transmit diversity scheme would provide the highest number of transmit signals that ensure orthogonal receive signals for the same radio spectrum and time resolution. Unfortunately, such an optimal diversity scheme does not exist. The suboptimal diversity shown in Fig. 12.10 can be implemented with a few hardware modifications. The combination of code and frequency diversity reduces the large frequency separation that would otherwise be necessary for a frequency diversity scheme. This suboptimal diversity basically increases the number of codes without increasing the code length. Although increasing the code length might produce the same result as this suboptimal diversity scheme, it would require an increased sampling rate, i.e., increased bandwidth on transmission and reception.

**Table 12.1** Advantages and disadvantages of transmit diversities

| Transmit diversity | Advantages | Disadvantages |
|---|---|---|
| Time | -Easy to implement<br>-No additional signal processing | -Poor time resolution<br>-Range ambiguity<br>-Reduced Doppler bandwidth<br>-Less average Tx power |
| Waveform | -One single operating frequency<br>- Scalable | -High range sidelobe<br>-Coupling between transmit signals might be problematic<br>-Requires specialized software |
| Optimal | -One single operating frequency<br>-Scalable | -Requires specialized hardware and software<br>-Coupling between transmit signals is minimized |

Table 12.1 summarizes the advantages and disadvantages of the diversity schemes discussed.

## 12.3 Applications of MIMO in Atmospheric Radar Imaging

In this section, we deal with the applications of MIMO to atmospheric radar imaging. Although MIMO has been used previously in communications and hard-target radar applications, our efforts below are pioneering in their respective fields, i.e., ionospheric irregularities, polar mesospheric summer echoes, and mesospheric wind measurements from meteor echoes. All of these three targets are stochastic in nature. The implementation and results of these applications are presented below.

### *12.3.1 MIMO in Atmospheric Radar Imaging for Ionospheric Studies*

Atmospheric radar imaging was first introduced and implemented in the early 1990s at the Jicamarca Radio Observatory (JRO) to study equatorial electrojet (EEJ) plasma instabilities [24]. Since then, the technique has been applied using different algorithms (see Sect. 12.2.4 in both 2D and 3D applications at different locations [21]). All of these previous studies were implemented with a single transmitter and many receivers, i.e., SIMO.

We proposed and implemented the first MIMO application to atmospheric radar imaging [39]. As in the case of the first SIMO imaging, our MIMO approach was implemented at the JRO to study EEJ instabilities. This time though, two

spatially separated antennas were used, allowing the virtual array with a larger aperture and more visibility samples than previously obtained with SIMO. This first implementation was tested with three transmit diversity schemes: time, polarization, and code. Frequency diversity was not implemented, since the echoes from EEJ instabilities are frequency dependent. In addition, Capon and MaxEnt imaging techniques were implemented. Clearly, resulting EEJ images were significantly improved when MIMO with MaxEnt was utilized.

Although polarization diversity was tested, it was not recommended, since more hardware was required, i.e., twice as many receivers, and it could not be scaled up if more transmitter diversity were needed. Time and code diversity provided similar results. However, time diversity has the disadvantage that all the available transmitter power is not used.

Based on the successful implementation at JRO, MIMO has also been theoretically evaluated to study fine ionospheric structures at high latitudes. In particular, different configurations of antennas as well as inversion techniques have been tested for the soon-to-be-finished EISCAT 3D radar. As in the case of JRO implementation, the MIMO study using the EISCAT 3D configuration consisted on dividing the transmitting array into multiple independent transmitters, as expected the resulting virtual arrays was much larger and with more samples, allowing measurements with unprecedented angular resolution [35].

### *12.3.2 Polar Mesospheric Summer Echoes Imaging*

Having demonstrated the utility of MIMO to improve radar imaging studies of ionospheric targets, MIMO has also been applied to 3D radar imaging studies of polar mesospheric summer echoes (PMSEs). Urco et al. [40] implemented SIMO and MIMO using the Middle Atmosphere Alomar Radar System (MAARSY) located in Northern Norway. MAARSY is an active and modular phased antenna array operating at 53.5 MHz and at a maximum peak power of 800 kW located in Andoya, Norway (69.30° N, 16.04° E). The antennas can be grouped into fifty-five (55) symmetric "hexagons" with seven (7) antennas each, and seven (7) adjacent "hexagons" are further grouped into an "anemone." An "anemone" can be used as one transmit or receive antenna element. The whole array has a directive gain of 33.5 dBi, a half-power beam width of 3.6°, and a maximum sidelobe suppression of 17 dB with respect to the mainlobe. A complete technical description of MAARSY can be found in [25].

Coherent MIMO was implemented at MAARSY as depicted in Fig. 12.11a. Three (3) anemones B, D, and F were used as transmit antenna elements, while fifteen (15) hexagons were used as receive antenna elements. Time diversity was employed in order to ensure that the transmit signals were orthogonal to each other. Each transmit antenna element was interleaved every 2 ms. The resulting virtual array has forty-five (45) virtual receive antenna elements as depicted in Fig. 12.11d and an angular resolution of $\approx 0.6°$. The resolution achieved is equivalent to an

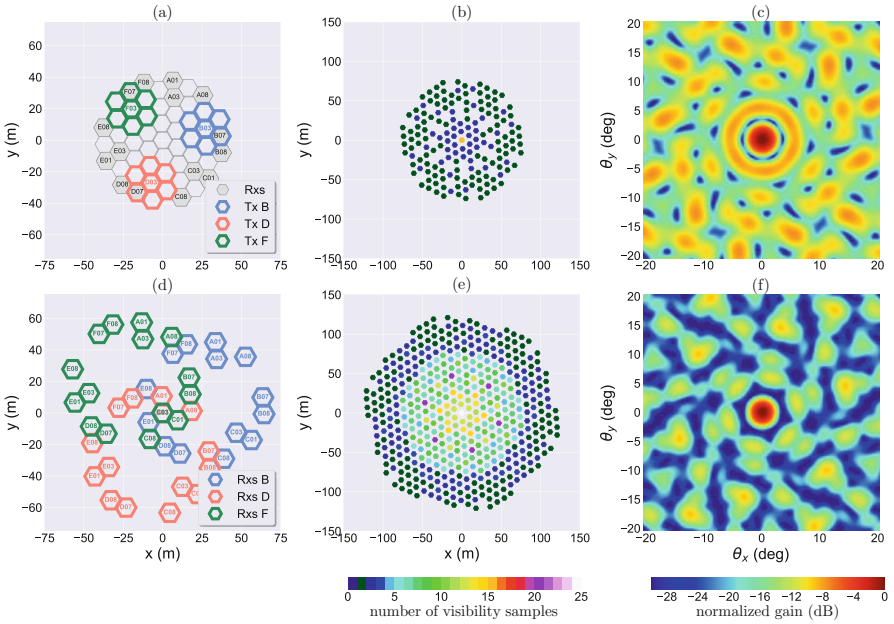



**Fig. 12.11** Antenna positions, visibility samples, and array point spread function (after [40]). (**a**) SIMO: antenna array. (**b**) SIMO: visibility (1 Tx). (**c**) SIMO: point spread function. (**d**) MIMO: virtual antenna array. (**e**) MIMO: visibility (3 Txs). (**f**) MIMO: point spread function

antenna area of 450 m which is more than five times the nominal diameter of MAARSY. Figures 12.11b and c show the visibility and the point spread function, respectively, for the SIMO case when only one transmit antenna element is used for transmission, while Figs. 12.11e and f show the same for the collocated MIMO configuration. The color coded redundancy is seen in Fig. 12.11b and e. The complete MAARSY MIMO configuration used to generate the result in Fig. 12.12 is summarized in Table 12.2.

Figure 12.12 shows an example of the East-West (EW)–North-South (NS) 2D image at 00:56:55 UT on July 17, 2017 at 85.8 km above ground using MAARSY. The intensity, Doppler, and spectral width are represented as lightness, hue, and saturation, respectively. From the results in Fig. 12.12, it can be seen that MaxEnt method has a better image quality than the Capon method for both SIMO and MIMO configurations. The Capon method tries to reduce the sidelobes adaptively by steering them to echo-free zones which unfortunately do not exist in this application as the entire area is filled with PMSE scattering. However, it should be noted that MaxEnt method is computationally more demanding than Capon. Furthermore and as expected, MIMO configuration reproduces a cleaner and more defined image than SIMO configuration for both MaxEnt and Capon method due to the improved angular resolution of MIMO configuration. It can be concluded based on the similar image quality in Fig. 12.12b and c that image reproduction using

**Table 12.2** Radar parameters used for PMSE observations

| Radar parameters | SIMO | MIMO |
|---|---|---|
| Frequency | 53.5 MHz | 53.5 MHz |
| Pulse coding | Complementary 16 | Complementary 16 |
| Pulse repetition frequency (PRF) | 1000 Hz | 1000 Hz |
| Range resolution | 450m | 450m |
| Number of coherent integrations | 8 | 8 |
| Effective PRF (after integration) | 12.5 Hz | 12.5 Hz |
| Number of FFT points | 16 | 16 |
| Number of incoherent integrations | 128 | 128 |
| Equivalent integration time | 81.92s | 81.92s |
| Number of transmitters (beams) | 1 | 5 (3 processed) |
| Transmit diversity | – | Time |
| Tx interleaving | – | 2 ms |

MIMO configuration and Capon (MIMO–Capon) has a comparable performance to SIMO configuration using MaxEnt (SIMO-MaxEnt).

To quantify the performance of the different implementations, we use a meteor echo which can be considered as a point target in range and angle with known scattering properties. Figure 12.13 shows the angular resolution achieved by the different imaging implementations considered in this experiment.

The middle and rightmost plots in Fig. 12.13 show the normalized angular power distributions in the East-West direction $\theta_x$ (middle plot) and North-South direction $\theta_y$ (rightmost plot), respectively. The sample points for a given angle were fitted to a Gaussian function, and the HPBW for each implementation was estimated.

The angular resolution achieved is summarized in Table 12.3. An improvement factor that serves as a reference to the theoretical angular resolution of the full MAARSY array is included in the last column. As expected, the improvement of using MIMO over SIMO configuration was roughly 50%. Surprisingly, an angular resolution of approximately 0.6° was achieved when MIMO was combined with the MaxEnt.

These unprecedented spatial–temporal observations of PMSE have allowed us to study a Kelvin–Helmholtz Instability (KHI) event in four dimensions for the first time [8]. By characterizing the spatial and temporal dimensions of the event, and using turbulent scaling analysis, we are able to qualify the flow conditions, i.e., it was turbulent with relatively high Reynolds numbers and weakly stratified with a Froude number close to 1.

Since the computational complexity of MaxEnt is high, our approach is to employ MIMO–Capon for a quick rough overview of a scenario and employ MIMO–MaxEnt for a more detailed evaluation for particular scenarios of interest.

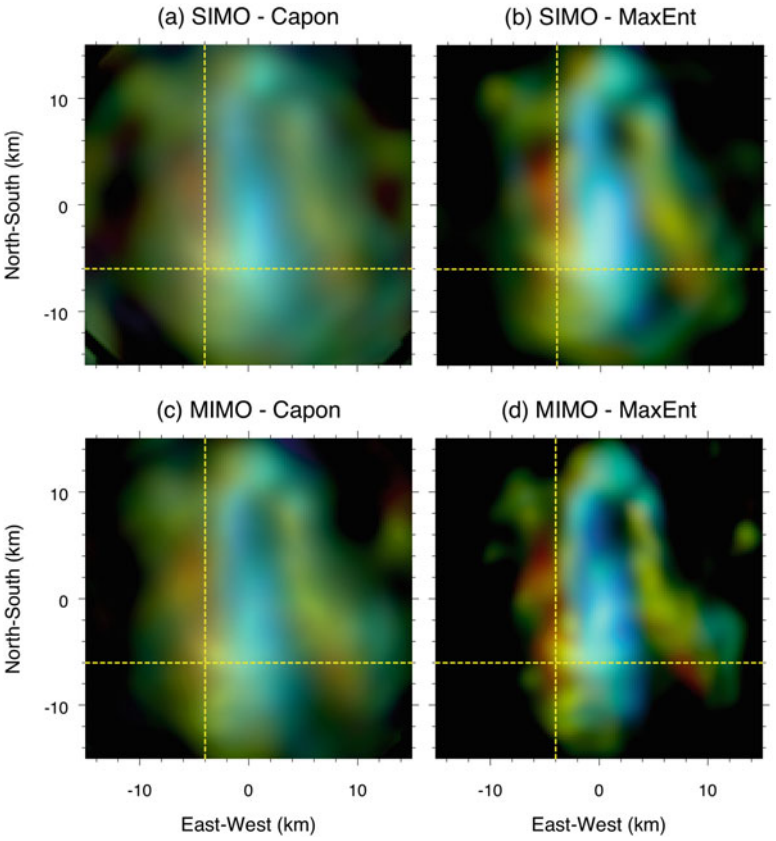



**Fig. 12.12** 2D PMSE images for a range of 85.8 km. (**a**) SIMO configuration with Capon's method, (**b**) SIMO configuration with MaxEnt method, (**c**) MIMO configuration with Capon's method, and (**d**) MIMO configuration with MaxEnt method (after [40])

### *12.3.3 MIMO in Specular Meteor Radars to Measure Mesospheric Winds*

Coherent MIMO has been applied also to specular meteor radars (SMRs), where a single target needs to be imaged at a given range and time. In standard SMRs, the transmitter and receiver are collocated, and the target localization is done by a receiver station consisting of at least five closely separated antennas (5-antenna interferometer) [17, 19]. Radar interferometry is a special case of radar imaging, where a single dominant target is at a given range, time, and frequency. Recently, the standard SMR system has been extended to a multistatic approach (incoherent MIMO), where either multiple interferometer receivers are added between 60–150 km from an existing transmitter [37] or the detections of two or more closely located monostatic systems working at different frequencies are combined [6]. In

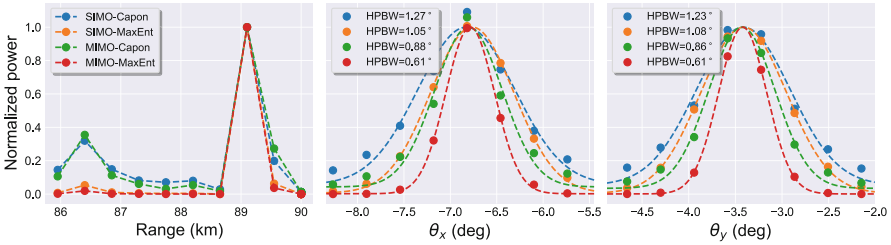



**Fig. 12.13** Normalized angular power distribution of specular meteor echo as a function of range (leftmost plot), East-West direction $\theta_x$ (middle plot), and North-South direction $\theta_y$ (rightmost plot) (after [40])

**Table 12.3** Performance of imaging techniques

| Technique | Angular resolution | Spatial resolution at 85 km | Equivalent antenna aperture | Improvement factor |
|---|---|---|---|---|
| MAARSY | 3.60° | 5.33 km | 76 m | – |
| SIMO–Capon | 1.27° | 1.88 km | 216 m | 2.83 |
| MIMO–Capon | 0.88° | 1.30 km | 312 m | 4.09 |
| SIMO–MaxEnt | 1.05° | 1.55 km | 261 m | 3.42 |
| MIMO–MaxEnt | 0.615° | 0.90 km | 450 m | 5.90 |

both cases, the angle of arrival with respect to the receiver is measured. By using coherent MIMO, the implementation of the multistatic approach is more reliable, cheaper, and easier to scale than previously thought. Figure 12.14 shows a sketch of coherent and non-coherent MIMO.

Again motivated by the successful implementations of MIMO to studies of EEJ and PMSE instabilities, MIMO has been implemented for SMRs. In this case, the interferometry was also done in transmission. To make use of the maximum available power and given that meteors do not last long, code diversity was implemented using coded continuous wave signal (i.e., spread spectrum transmission). Each transmit antenna used a different code, and all codes were quasi-orthogonal to each other. This implementation is now called SIMONe (Spread Spectrum Interferometric Multistatic meteor radar Observing Network). Details of the implementation and first results can be found in [7]. Figure 12.15 shows sketches of the different configurations possible with SIMONe.

Besides MIMO and spread spectrum, SIMONe makes use of a simple and computationally efficient compressed sensing approach to decode the received signals [41]. The approach allows us to go after strong targets as well as weak targets, taking advantage of the sparseness nature of the meteors. A block diagram of our compressed sensing implementation in SIMONe is shown in Fig. 12.16.

After the prototype test, a 10-day campaign called SIMONe 2018 was conducted in November 2018. During this campaign, SIMONe links were added to existing MMARIA links in northern Germany. 2 MIMO and 4 SIMO links were added in a

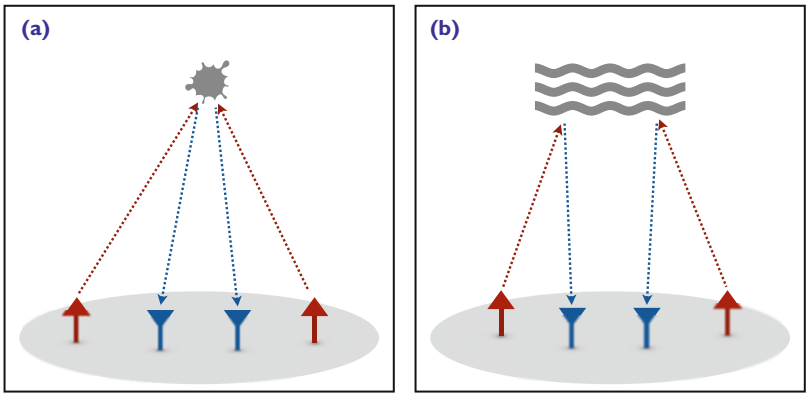



**Fig. 12.14** Coherent versus non-coherent. (**a**) RCS diversity. (**b**) Spatial diversity

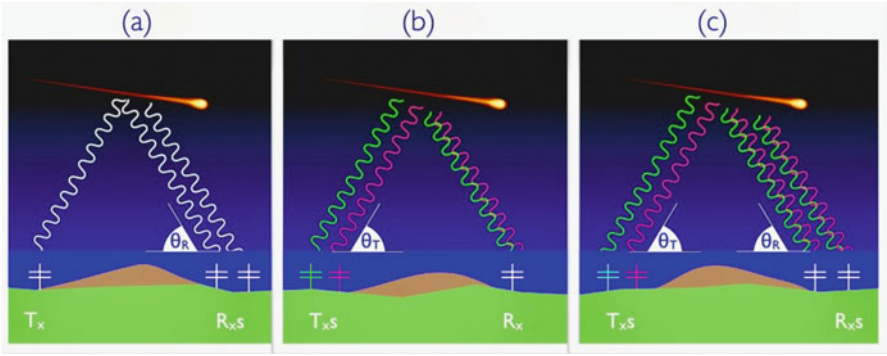



**Fig. 12.15** Sketch of SIMONe configurations: (**a**) SIMO, (**b**) MISO, and (**c**) MIMO (after [7])

couple of days. During this campaign, more than 200 thousand meteors per day were detected, approximately 20 times more than in a traditional system. Figure 12.17 shows SIMONe 2018 detections on November 5, 2018. Given the high number of good quality detections, a second-order statistics approach to investigate second-order wind statistics was developed [44]. SIMONe 2018 data have also been used to study the mesospheric frequency spectra of horizontal winds during the campaign [5], as well as primary and secondary gravity waves [42]. Specific details of the campaign can be found in [5].

The SIMONe concept is now matured and currently in operation in Peru [9] and Argentina [12] to study the mesosphere dynamics under different background geophysics. New operational deployments are being planned in Northern Norway and Northern Germany. The details of operational SIMONe can be found in [9].

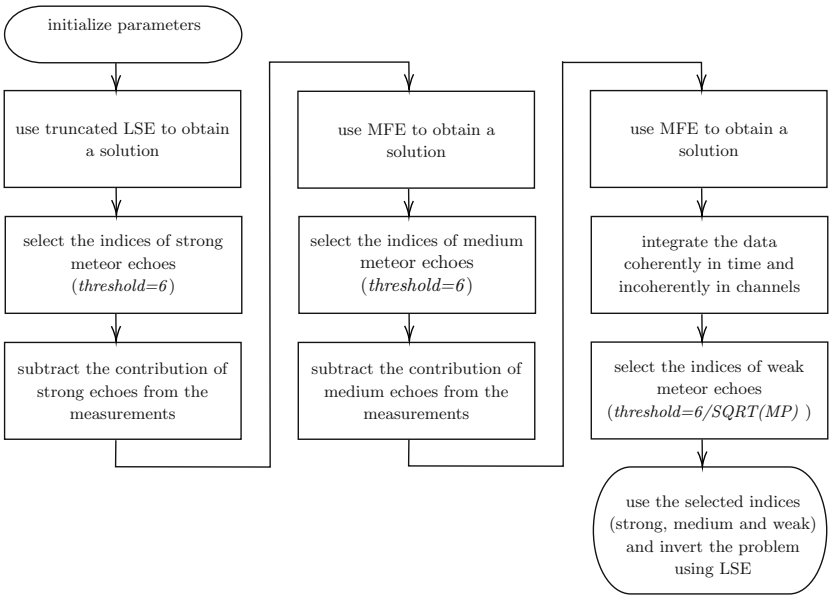



**Fig. 12.16** Block diagram of SIMONe compressing sensing decoding (after [41])

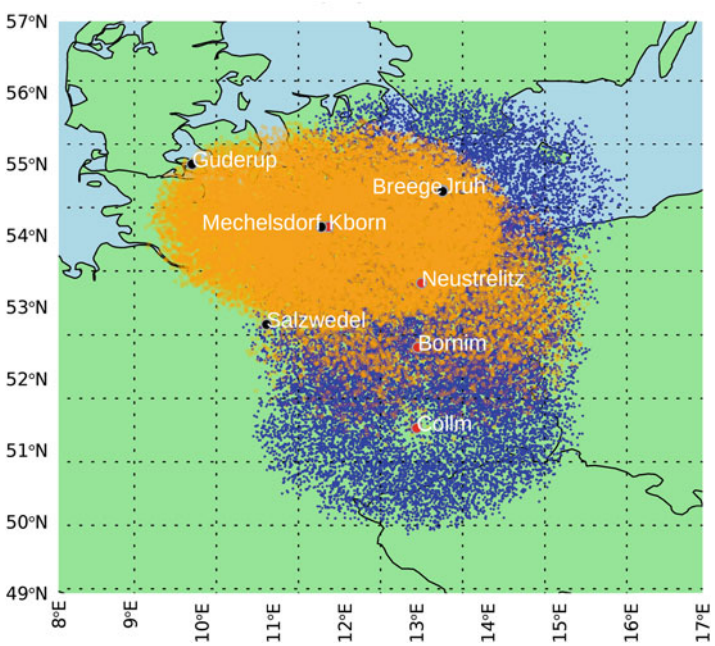



**Fig. 12.17** Meteor detections on November 5, 2018 during the SIMONe 2018 campaign. More than 200 thousand meteors were detected (after [44])

## 12.4 Summary and Future Work

We have presented applications of MIMO that improve current atmospheric radars. Specifically, one can achieve a better angular resolution by using MIMO on existing transmitting arrays, thereby synthesizing a larger virtual array. Another important improvement is the development of the SIMONe concept, which revolutionizes the implementation and performance of multistatic specular meteor radars.

We have done some preliminary work on using machine learning to solve the inverse problem in ARI. We intend to publish the results in a future work.

Future work will also focus on employing more sophisticated CS techniques for ARI and then to combine tracking techniques with CS. A special feature of ARI is the fact that the brightness, i.e., the image, is nonnegative, which was exploited in the signal recovery, e.g., [23]. Furthermore, the rather small number of antennas, i.e., samples, enables novel otherwise too complex signal recovery techniques. The application of MIMO configurations results in a highly structured sensing matrix. It basically results from the Kronecker product of the transmitter side and the receiver side steering matrix. The suitability of MIMO antenna configurations for radar imaging was analyzed. MIMO is also useful to validate methods that do not use MIMO and heavily rely on the exploitation of a priori knowledge. This task is especially important to study weak PMSE signals given that in the case of MIMO the available transmitter power has to be shared. Systems without MIMO capability would have to solely rely on CS ARI. A first approach toward combining CS and tracking techniques to study PMSE is to look at the whole measurement sequence at once and to apply conventional signal recovery algorithms. Unfortunately, this brute force approach has high computational complexity and is limited to time dynamics which can be described by a linear model. Thus, it seems attractive to separate the tracking, which is also justified by theory. If the time dynamics can be modeled by a Markov model, a single forward or a forward and a backward iteration shall be used depending on whether only the past measurements (online) or past and future measurements (offline) are exploited, respectively. When separating the tracking signal recovery algorithms, we might use algorithms exploiting statistical a priori knowledge, like Bayesian approximate message passing (BAMP). For the tracking, Kalman filters could be used in the case of Gaussian models. Grid-based methods, which can be combined with any discretized stochastic model, are attractive and will be the main focus of our project as long as the size of the scenario is not too large. If the computational complexity of grid based methods is too large, one could resort to particle filtering, e.g., [43].

The work presented in this chapter was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under the project number 403837627, called “Compressed sensing radar imaging of polar mesospheric summer echoes using tracking and MIMO approaches (CP-PMSE-MIMO)” as part of the priority program called “Compressed Sensing in Information Processing (CoSIP).”

## Table of Mathematical Symbols

| Symbol | Definition |
|---|---|
| $\mathbf{a}$ | Steering vector |
| $\mathbf{A}$ | Sensing matrix |
| $b$ | Brightness |
| $\mathbf{b}$ | Brightness vector |
| $\hat{\mathbf{b}}$ | Estimated brightness vector |
| $C$ | Cross-correlation of two waveforms |
| $d$ | Range |
| $c$ | Speed of light |
| $\mathbf{f}$ | Brightness vector in a given sparsity basis |
| $k$ | Receive antenna element |
| $K$ | Total number of receive antenna elements |
| $n$ | Target |
| $N$ | Total number of targets |
| $\underline{n}$ | Noise |
| $p$ | Transmit antenna element |
| $P$ | Total number of transmit antenna elements |
| $\vec{r}$ | Position vector |
| $\underline{s}$ | Received signal |
| $T$ | Pulse repetition interval |
| $\vec{u}$ | Unit vector |
| $\underline{v}$ | Visibility |
| $\mathbf{v}$ | Visibility vector |
| $w$ | Coded waveform |
| $w_{\max}$ | Maximum unambiguous Doppler frequency |
| $\vec{\beta}$ | Wavenumber vector |
| $\Delta\vec{r}$ | Displacement |
| $\Delta\theta$ | Angular resolution |
| $\Delta\rho$ | Path length difference |
| $\theta$ | Elevation angle |
| $\theta_{\max}$ | Maximum unambiguous elevation angle |
| $\lambda$ | Wavelength |
| $\sigma^2$ | Noise power (variance) |
| $\tau$ | Transmitted pulse width |
| $\mathbf{\Psi}$ | Sparsity basis |

## References


1. Baumgarten, G., Fritts, D.: Quantifying Kelvin-Helmholtz instability dynamics observed in noctilucent clouds: 1. methods and observations. J. Geophys. Res. Atmos. **119**(15), 9324–9337 (2014)
2. Candès, E.: Compressive sampling. In: Proceedings of the international congress of mathematicians, vol. 3, pp. 1433–1452. Madrid, Spain (2006)
3. Candès, E., Romberg, J., Tao, T.: Stable signal recovery from incomplete and inaccurate measurements. Commun. Pure Appl. Math. **59**(8), 1207–1223 (2006)
4. Capon, J.: High-resolution frequency-wavenumber spectrum analysis. Proc. IEEE **57**(8), 1408–1418 (1969)
5. Charuvil Asokan, H., Chau, J., Marino, R., Vierinen, J., Vargas, F., Urco, J., Clahsen, M., Jacobi, C.: Study of second-order wind statistics in the mesosphere and lower thermosphere region from multistatic specular meteor radar observations during the SIMONe 2018 campaign. Atmos. Chem. Phys. Discuss. **2020**, 1–27 (2020)
6. Chau, J., Stober, G., Hall, C., Tsutsumi, M., Laskar, F., Hoffmann, P.: Polar mesospheric horizontal divergence and relative vorticity measurements using multiple specular meteor radars. Radio Sci. **52**(7), 811–828 (2017)
7. Chau, J., Urco, J., Vierinen, J., Volz, R., Clahsen, M., Pfeffer, N., Trautner, J.: Novel specular meteor radar systems using coherent MIMO techniques to study the mesosphere and lower thermosphere. Atmos. Meas. Tech. **12**, 2113–2127 (2019). https://doi.org/10.5194/amt-12-2113-2019
8. Chau, J., Urco, J., Avsarkisov, V., Vierinen, J., Latteck, R., Hall, C., Tsutsumi, M.: Four-dimensional quantification of Kelvin-Helmholtz instabilities in the polar summer mesosphere using volumetric radar imaging. Geophys. Res. Lett. **47**(1), e2019GL086081 (2020). https://doi.org/10.1029/2019GL086081
9. Chau, J., Urco, J., Vierinen, J., Harding, B., Clahsen, M., Pfeffer, N., Kuyeng, K., Milla, M., Erickson, P.: Multistatic specular meteor radar network in Peru: System description and initial results. Earth and Space Science **8**(1), e2020EA001293 (2021). https://doi.org/10.1029/2020EA001293
10. Chen, C., Vaidyanathan, P.: Properties of the MIMO radar ambiguity function. In: 2008 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 2309–2312. IEEE, New York (2008)
11. Chu, D.: Polyphase codes with good periodic correlation properties. IEEE Trans. Inf. Theory **18**(4), 531–532 (1972)
12. Conte, J., Chau, J., Urco, J., Latteck, R., Vierinen, J., Salvador, J.: First studies of mesosphere and lower thermosphere dynamics using a multistatic specular meteor radar network over southern Patagonia. Earth and Space Science **8**(2), e2020EA001356 (2021). https://doi.org/10.1029/2020EA001356
13. Donoho, D.: For most large underdetermined systems of linear equations the minimal L1-norm solution is also the sparsest solution. Commun. Pure Appl. Math. **59**(6), 797–829 (2006). https://doi.org/10.1002/cpa.20132
14. Frank, R.: Polyphase codes with good nonperiodic correlation properties. IEEE Trans. Inf. Theory **9**(1), 43–45 (2006)
15. Gold, R.: Optimal binary sequences for spread spectrum multiplexing. IEEE Trans. Inf. Theory **13**(4), 619–621 (1967)
16. Harmuth, H.: Transmission of Information by Orthogonal Functions, 2nd edn. Springer, Heidelberg (1972). https://doi.org/10.1007/978-3-642-61974-8
17. Hocking, W., Fuller, B., Vandepeer, B.: Real-time determination of meteor-related parameters utilizing modern digital technology. J. Atmos. Sol. Terr. Phys. **63**(2), 155–169 (2001)
18. Hoctor, R., Kassam, S.: The unifying role of the coarray in aperture synthesis for coherent and incoherent imaging. Proc. IEEE **78**(4), 735–752 (1990). https://doi.org/10.1109/5.54811

19. Holdsworth, D., Reid, I., Cervera, M.: Buckland Park all-sky interferometric meteor radar. Radio Sci. **39**(5), 1–12 (2004). https://doi.org/10.1029/2003RS003014
20. Hysell, D., Chau, J.: Optimal aperture synthesis radar imaging. Radio Sci. **41**(02), 1–12 (2006). https://doi.org/10.1029/2005RS003383
21. Hysell, D.L., Aveiro, H.C., Chau, J.L.: Ionospheric Irregularities, chap. 18, pp. 217–240. American Geophysical Union (AGU), Washington, D.C. (2014). https://doi.org/10.1002/9781118704417.ch18
22. Johnson, D., Dudgeon, D.: Array signal processing: concepts and techniques. Prentice Hall, Englewood Cliffs, NJ (1993)
23. Khajehnejad, M., Dimakis, A., Xu, W., Hassibi, B.: Sparse recovery of nonnegative signals with minimal expansion. IEEE Trans. Signal Process. **59**(1), 196–208 (2010)
24. Kudeki, E., Sürücü, F.: Radar interferometric imaging of field-aligned plasma irregularities in the equatorial electrojet. Geophys. Res. Lett. **18**(1), 41–44 (1991)
25. Latteck, R., Singer, W., Rapp, M., Vandepeer, B., Renkwitz, T., Zecha, M., Stober, G.: MAARSY: The new MST radar on Andøya-system description and first results. Radio Sci. **47**(1), 1–18 (2012)
26. Le Pennec, E., Mallat, S.: Bandelet image approximation and compression. Multiscale Model. Simul. **4**(3), 992–1039 (2005)
27. Li, J., Xu, L., Stoica, P., Forsythe, K., Bliss, D.: Range compression and waveform optimization for MIMO radar: a Cramer-Rao bound based study. IEEE Trans. Signal Process. **56**(1), 218–232 (2007)
28. MacWilliams, F., Sloane, N.: Pseudo-random sequences and arrays. Proc. IEEE **64**(12), 1715–1729 (1976). https://doi.org/10.1109/PROC.1976.10411
29. Palmer, R., Gopalam, S., Yu, T., Fukao, S.: Coherent radar imaging using Capon's method. Radio Sci. **33**(6), 1585–1598 (1998)
30. Peyre, G.: Best basis compressed sensing. IEEE Trans. Signal Process. **58**(5), 2613–2622 (2010)
31. Powell, M.: A hybrid method for nonlinear equations. In: Rabinowitz, P. (ed.) Numerical Methods for Nonlinear Algebraic Equations. Gordon and Breach (1970)
32. San Antonio, G., Fuhrmann, D., Robey, F.: MIMO radar ambiguity functions. IEEE J. Sel. Top. Sign. Proces. **1**(1), 167–177 (2007)
33. Smith, D., Arlinghaus, L., Yankeelov, T., Welch, E.: Curvelets as a sparse basis for compressed sensing magnetic resonance imaging. In: Ourselin, S., Haynor, D. (eds.) Medical Imaging 2013: Image Processing, vol. 8669, pp. 621–627. SPIE, New York (2013)
34. Sommer, S., Chau, J.: Patches of polar mesospheric summer echoes characterized from radar imaging observations with MAARSY. In: Annales Geophysicae, vol. 34, pp. 1231–1241. Copernicus GmbH, Germany (2016)
35. Stamm, J., Vierinen, J., Urco, J., Gustavsson, B., Chau, J.: Radar imaging with EISCAT 3D. Ann. Geophys. **39**(1), 119–134 (2021)
36. Stéphane, M.: Sparse representations. In: Stéphane, M. (ed.) A Wavelet Tour of Signal Processing, 3rd edn., pp. 1–31. Elsevier, Amsterdam (2009)
37. Stober, G., Chau, J.: A multistatic and multifrequency novel approach for specular meteor radars to improve wind measurements in the MLT region. Radio Sci. **50**(5), 431–442 (2015). https://doi.org/10.1002/2014RS005591. 2014RS005591
38. Taubman, D., Marcellin, M.: Image transforms. In: JPEG2000 Image compression fundamentals, standards and practice, pp. 143–207. Springer, Berlin (2002)
39. Urco, J., Chau, J., Milla, M., Vierinen, J., Weber, T.: Coherent MIMO to improve aperture synthesis radar imaging of field-aligned irregularities: First results at Jicamarca. IEEE Trans. Geosci. Remote Sens. **56**(5), 2980–2990 (2018). https://doi.org/10.1109/TGRS.2017.2788425
40. Urco, J., Chau, J., Weber, T., Latteck, R.: Enhancing the spatio-temporal features of polar mesosphere summer echoes using coherent MIMO and radar imaging at MAARSY. Atmos. Meas. Tech. **12**, 955–969 (2019). https://doi.org/10.5194/amt-12-955-2019

41. Urco, J., Chau, J., Weber, T., Vierinen, J., Volz, R.: Sparse signal recovery in MIMO specular meteor radars with waveform diversity. IEEE Trans. Geosci. Remote Sens. **57**(12), 10088–10098 (2019). https://doi.org/10.1109/TGRS.2019.2931375
42. Vargas, F., Chau, J., Charuvil Asokan, H., Gerding, M.: Mesospheric gravity wave activity estimated via airglow imagery, multistatic meteor radar, and saber data taken during the SIMONe 2018 campaign. Atmos. Chem. Phys. Discuss. **2020**, 1–27 (2020)
43. Vaswani, N., Zhan, J.: Recursive recovery of sparse signal sequences from compressive measurements: A review. IEEE Trans. Signal Process. **64**(13), 3523–3549 (2016)
44. Vierinen, J., Chau, J., Asokan, H.C., Urco, J., Clahsen, M., Avsarkisov, V., Marino, R., Volz, R.: Observing mesospheric turbulence with specular meteor radars: A novel method for estimating second order statistics of wind velocity. Earth and Space Sciences **6**(7), 1171–1195 (2019). https://doi.org/10.1029/2019EA000570
45. Wilczek, R., Drapatz, S.: A high accuracy algorithm for maximum entropy image restoration in the case of small data sets. Astron. Astrophys. **142**, 9–12 (1985)
46. Woodman, R.: Coherent radar imaging: Signal processing and statistical properties. Radio Sci. **32**(6), 2373–2391 (1997). https://doi.org/10.1029/97RS02017
47. Yuan, X., Haimi-Cohen, R.: Image compression based on compressive sensing: end-to-end comparison with JPEG. IEEE Trans. Multimedia **22**(11), 2889–2904 (2020)

# Chapter 13
# Over-the-Air Computation for Distributed Machine Learning and Consensus in Large Wireless Networks


**Matthias Frey, Igor Bjelaković, and Sławomir Stańczak**

## 13.1 Introduction

Wireless communications is becoming increasingly pervasive both in everyday life and in business and industry. Its importance is increasing in vehicular communications, smart manufacturing, mobile health care, environmental monitoring, and smart agriculture, to name a few examples. Due to the ubiquity of wireless devices, efficient use of the electromagnetic spectrum is a more pressing issue than ever: Despite the availability of additional spectrum, e.g., in the millimeter wave range, the scarcity of communication resources is expected to continue to be an increasing problem as the size of wireless networks increases. Therefore, if the traditional paradigm of source-channel separation is followed, then the number of wireless devices that can be deployed in any given geographical area will be too small to realize many of the aforementioned applications. A very promising approach to accommodate more devices is to relax the separation between the source and channel to a certain extent and design communication schemes from the ground up targeted to specific technical applications. A key observation in this context is that often, the data available at the transmitters is either redundant or it is only of

M. Frey (✉)
Technische Universität Berlin, Network Information Theory Group, Berlin, Germany
e-mail: matthias.frey@tu-berlin.de

I. Bjelaković
Fraunhofer Heinrich Hertz Institute, Berlin, Germany
e-mail: igor.bjelakovic@hhi.fraunhofer.de

S. Stańczak
Technische Universität Berlin, Network Information Theory Group, Berlin, Germany

Fraunhofer Heinrich Hertz Institute, Berlin, Germany
e-mail: slawomir.stanczak@tu-berlin.de



G. Kutyniok et al. (eds.), *Compressed Sensing in Information Processing*, Applied and Numerical Harmonic Analysis, https://doi.org/10.1007/978-3-031-09745-4_13

interest for the receiver in a combined form. If the application-specific combining of information is performed exclusively or predominantly in receiver-side processing, this alone can imply orders of magnitude worse resource utilization than is required for the application at hand, especially if the wireless network is very large.

This can be immediately concluded from the data processing inequality which implies that no receiver-side processing of a signal can increase the information contained in the signal [27, Section 2.3]. Therefore, the entropy or the amount of information contained in $f(s_1, \ldots, s_K)$, where $s_1, \ldots, s_K$ are random variables, is smaller than or equal to the amount of information contained in the random vector $(s_1, \ldots, s_K)$. Consequently, combining information to evaluate a function $f$ can incur a loss of information at the receiver side. In many cases of practical interest, the information loss is significant, and communication resources can be used much more efficiently if the combining is carried out in the channel and not in post-processing, as we illustrate in the following example.

*Example 13.1* Suppose that $K$ transmitters send their data $s_1, \ldots, s_n$ to a single receiver through a multiple-access channel. For simplicity, we assume that each $s_k$ is an independent random variable uniformly distributed over $\mathcal{S} = \{0, 1\}$. Now if the receiver reconstructs each of these variables, then the entropy or the amount of information available at the receiver is $\sum_{k=1}^{K} H(s_k) = K$ bits where $H : \mathcal{S} \to \mathbb{R}_{\geq 0} : s \mapsto \sum_{s \in \mathcal{S}} p(s) \log_2(1/p(s))$ is the Shannon entropy[1] and $p : \mathcal{S} \mapsto [0, 1]$ is the probability mass function. This means that the transmitters have to transmit $K$ bits to the receiver. Therefore, if the capacity of the communication channel is 1 bit per channel use, then $K$ channel uses are necessary to convey the full information to the receiver.[2] Now we assume that the receiver is only interested in $f(s_1, \ldots, s_K) = \sum_{k=1}^{K} s_k$ which can be easily computed from $s_1, \ldots, s_K$. By the data processing inequality, this operation cannot increase the amount of information. In fact, the entropy of the function is $H(\sum_k s_k) = K - \sum_{k=0}^{K} \binom{K}{k} 2^{-K} \log_2 \binom{K}{k}$ which is strictly smaller than $K$ for all $K \geq 2$. This means that instead of transmitting $K$ bits that are necessary to reconstruct each $s_k$, the transmitters can send significantly less information to the receiver if its objective is to compute the sum function $f(s_1, \ldots, s_K)$.

The class of communication schemes in which the signals are transmitted concurrently and the processing at the transmitter and receiver sides are designed so that the receiver directly reconstructs the combined information that is necessary for a certain application is called Computation over Multiple-Access Channel (CoMAC), AirComp or Over-the-Air (OTA) computation. The goal of these schemes is to obtain a scaling behavior of the communication cost in the number of transmitters

[1] We use the convention $0 \cdot \log(1/0) := 0$ in the definition.

[2] In the case of orthogonal channel access, it is necessary to establish $K$ independent (interference-free) communication channels, where each of these has the capacity of 1 bit per channel use.

that is better than the linear growth[3] that would ensue from a separation of source and channel coding. Therefore, such schemes exhibit the inherent property that the receiver is unable to fully reconstruct all of the transmitted information.

This paradigm shift away from source-channel separation has great potential to solve a fundamental scaling law issue that could otherwise hinder the development of the envisioned massively sized wireless networks. However, it also comes with a major downside, as it renders virtually all existing security schemes inapplicable. For example, since OTA computation usually involves the transmission of analog instead of digital signals, neither standard methods of cryptography nor of Physical Layer Security can be used. On the other hand, secure communication is becoming increasingly important in an interconnected world. For some applications such as e-health and smart manufacturing, a lack of security guarantees could be enough to completely prevent the development of a communication scheme in the first place. It is, therefore, of essential importance in the development of OTA computation schemes to design them from the ground up with security concerns in mind. While existing methods of Physical Layer Security cannot be directly applied, there is a variety of tools that can guide the development of OTA computation schemes which can guarantee security on the Physical Layer.

In the remainder of this chapter, we first give an overview of the current state of the art in the area of OTA computation, as well as two of its currently most prominent applications: Distributed Machine Learning and distributed consensus. We then survey our own contributions, which include the development of an OTA computation scheme for fast-fading channels, applications to consensus algorithms and Distributed Machine Learning as well as the first steps towards the design of OTA computation schemes which feature inherent protection against eavesdropping attacks. The integration of security guarantees into the design of OTA computation schemes has, to the best of our knowledge, not appeared in the literature before. We conclude the chapter with a summary of open research problems.

## 13.2 Over-the-Air Computation

The idea of a scheme that allows a receiver to reconstruct directly a combined form of two messages, but not the original messages themselves, can be traced back to [48] where a source coding problem is formulated in which it is the receiver's task to reconstruct a sequence of modulo-2 sums of encoded bits. An uncoded analog scheme for obtaining a noisy estimate of a function of transmitted values with an application to wireless sensor networks has appeared in [34] and is, to the best of our knowledge, the first work that proposes a joint source-channel approach to OTA computation.

[3] If the expense necessary for coordination and scheduling is also considered, this growth can even be superlinear.

The authors in [34] take an analog approach in which a certain amount of noise is tolerated in the received value and the function is computed only once.[4] This is in contrast with a class of digital schemes that are closer to [48] in the sense that they also consider functions with finite domains and typically give error guarantees for a large number of repeated function computations.

### *13.2.1 Digital Over-the-Air Computation*

In digital OTA computation, the function that is to be computed maps between discrete sets. The computation is carried out repeatedly, and the objective of the corresponding coding scheme is that the probability of a decoding error approaches zero as the number of repetitions tends to infinity.

More formally, [57] introduces the problem of digital computation coding in the following way:

**Definition 13.1** A *digital computation coding problem* consists of the following:

- A multiple-access channel $W$ which maps channel inputs $X_1, \ldots, X_K$ ranging over the input alphabets $\mathcal{X}_1, \ldots, \mathcal{X}_K$ to a channel output $Y$ which ranges over the channel output alphabet $\mathcal{Y}$.
- An *objective function*

$$f : \mathcal{S}_1 \times \cdots \times \mathcal{S}_K \to \mathcal{S}, \tag{13.1}$$

  where $\mathcal{S}_1, \ldots, \mathcal{S}_K, \mathcal{S}$ are finite sets.
- A probability distribution on $\mathcal{S}_1 \times \cdots \times \mathcal{S}_K$.

The idea is that, given this problem, the transmitters encode their messages $S_1, \ldots, S_K$ as sequences of channel inputs in such a way that the receiver can, with high probability of success, reconstruct $f(S_1, \ldots, S_K)$ without necessarily being able to draw any further information about $S_1, \ldots, S_K$.

**Definition 13.2** An $(m, M, \varepsilon)$-*code* for a given digital computation coding problem consists of:

- for each $k \in \{1, \ldots, K\}$, an *encoder*

$$F_k : \mathcal{S}_k^m \to \mathcal{X}_k^M \tag{13.2}$$

- a *decoder*

$$D : \mathcal{Y}^M \to \mathcal{S}^m \tag{13.3}$$

[4] The function can be computed multiple times since the scheme can simply be repeated, however, the individual instances do not take advantage of the repeated computation.

such that if the sequence of channel inputs is determined by $X_k^M := F_k(S_k^M)$, the error probability at the receiver satisfies

$$\mathbb{P}\left(D(Y^M) \neq (f(S_1^{(1)}, \ldots, S_K^{(1)}), \ldots, f(S_1^{(m)}, \ldots, S_K^{(m)}))\right) \leq \varepsilon. \tag{13.4}$$

These notions can then be used to define the analog of rate and capacity in classical source or channel coding problems.

**Definition 13.3** The *computation rate* of an $(m, M, \varepsilon)$-code is defined as the ratio $m/M$. A computation rate $R$ is called *achievable* if there is a sequence of $(m, M, \varepsilon)$-codes of computation rate $R$ where $M \to \infty$ and $\varepsilon \to 0$. The *computation capacity* is the supremum of all achievable computation rates.

This framework is extended by allowing the alphabets $\mathcal{S}_1, \ldots, \mathcal{S}_K, \mathcal{S}$ to be infinite and then characterizing the rate-distortion trade-off. In any case, the computation coding problem combines source and channel coding because the encoders simultaneously remove redundancy from the sources and protect the transmission against channel noise. The authors of [57] note examples where the rate that separate source and channel coding can achieve is strictly less than the computation capacity.

In the setting with finite alphabets, the typical objective function considered is addition in a finite field, and the main application noted by the authors is physical layer network coding. This idea was seminal to a lot of follow-up research (e.g., [38, 58, 59, 62, 80]) which has expanded upon and refined the idea of using Over-the-Air computation as a means for increasing the efficiency of network coding. Notably, there is also a work [37] which proposes schemes that use digital computation codes in conjunction with a quantizer to compute functions that are of interest in other applications, such as the arithmetic mean, the geometric mean, and the Euclidean norm.

### *13.2.2 Analog Over-the-Air Computation*

The framework of digital computation codes is promising and its applications to network coding are highly relevant as they can realize impressive performance gains in wireless networks. However, it also has downsides in the context of other applications:

- The notion of computation capacity is an asymptotic one valid only for block lengths tending to infinity. While finite-blocklength results are certainly conceivable, it is nonetheless an inherent property of any approach involving digital coding that a certain number of repeated function computations is necessary in order to guarantee a reasonably low probability of decoding error. This can

be problematic in applications where only a few computations are necessary or where protocols are used in which the roles of transmitters and receivers change frequently with only very few computations being done between these changes.
- To the best of our knowledge, the only known digital coding schemes which can deal with channel fading compute sums over finite fields for the application of network coding. Examples of functions that existing digital schemes cannot compute over fading channels include weighted sums which have a high relevance in the context of OTA ML, as well as maxima and various kinds of averages which are important in the context of consensus algorithms and control systems.
- The digital coding schemes can only deal with discrete messages. If real (or floating point) numbers are processed in a certain application, a quantizer needs to be added to the system. Since quantization is a form of source coding, this is somewhat in contrast with the observation that joint source-channel approaches are necessary to achieve optimum system performance.

A way to make OTA computation applicable where these disadvantages hinder the use of digital schemes is to process analog input values directly into an electromagnetic signal without first going through a sequence of bits (or other discrete values) as an intermediary step. A striking observation in this context is that a standard wireless channel actually performs a summation of the transmitted signals (which, through their IQ representations, can be seen as points in Euclidean space). This opens the door to the computation both of weighted sums and (as a special case) arithmetic averages, which we have noted above are very relevant functions both for OTA ML and consensus algorithms. There are two important research questions that these observations directly raise:

- If we were able to compute real function values in an analog system without error, this would in the point-to-point case degrade to a possibility to losslessly transmit a real number through the wireless channel which would imply infinite Shannon capacity of the channel. Since this is known to be unrealistic for any real-world channel, we can immediately conclude that a certain amount of noise in the computed function values is unavoidable in any kind of analog OTA computation scheme. But is it possible to control the strength of the noise, for instance, by providing tail bounds for its magnitude?
- We can expect from the structure of the wireless channel that it can compute sums in Euclidean space, but can we, with the use of suitable pre- and post-processing schemes, compute a larger class of functions OTA?

With respect to the latter question, it is clear that since the wireless channel performs an addition of its input signals, the class of functions that we can compute OTA are in a certain sense functions that can be reduced to a summation. In order to make this statement more precise, we use an already existing mathematical notion, called nomographic functions, that defines this kind of functions formally. This connection was first observed in [35] and has been discussed and analyzed in [36] in more detail than we can in the following summary.

**Definition 13.4** A *nomographic representation* of a function $f : \mathbb{R}^K \to \mathbb{R}$ consists of functions $f_1, \ldots, f_K, F : \mathbb{R} \to \mathbb{R}$ such that

$$\forall x_1, \ldots, x_K \in \mathbb{R} : \quad f(x_1, \ldots, x_K) = F\left(\sum_{k=1}^{K} f_k(x_k)\right). \tag{13.5}$$

A function $f : \mathbb{R}^K \to \mathbb{R}$ which has a nomographic representation is called a *nomographic function*.

It has been noted in [18, Theorem 8] that every function is nomographic according to this definition. We state a version of this result that fits with Definition 13.4. Since it illustrates the arguments below very well, we also give a full proof, based on the same idea as in [18].

**Theorem 13.1** (adapted from [18, Theorem 8]). *Every function $f : \mathbb{R}^K \to \mathbb{R}$ is nomographic.*

***Proof*** We first fix an arbitrary bijection $\phi : \mathbb{R} \to (0, 1)$. An example of a possible choice is

$$\phi : x \mapsto \begin{cases} \frac{1}{2} \cdot \frac{1}{x+1}, & x \in (0, \infty) \\ \frac{1}{2} \cdot \left(1 + \frac{1}{|x|+1}\right), & x \in (-\infty, 0) \\ \frac{1}{2}, & x = 0. \end{cases} \tag{13.6}$$

Next, we define for every $x \in (0, 1)$ the decimal[5] representation of $x$ as a sequence of digits $a_{x,1}, a_{x,2}, \cdots \in \{0, \ldots, 9\}$ such that

$$x = 0.\overbrace{a_{x,1}a_{x,2}\ldots}^{\text{dec}}, \tag{13.7}$$

with the definition

$$0.\overbrace{a_{x,1}a_{x,2}\ldots}^{\text{dec}} := \sum_{i=1}^{\infty} a_{x,i} \cdot 10^{-i}. \tag{13.8}$$

We make the choice for the sequence $a_{x,1}, a_{x,2}, \ldots$ unique by requiring that it has to contain infinitely many non-zero elements. Let, for all $k \in \{1, \ldots, K\}$,

[5] Of course, there is nothing special about base 10 here, and in fact, [18] uses dyadic representations. We have chosen the base 10 here so that our representation coincides with the usual decimal notation of numbers.

$$f_k(x) := 0.\overbrace{\underbrace{0\ldots0}_{k-1}a_{\phi(x),1}\underbrace{0\ldots0}_{K-1}a_{\phi(x),2}\underbrace{0\ldots0}_{K-1}a_{\phi(x),3}\underbrace{0\ldots0}_{K-1}\ldots}^{\text{dec}} \tag{13.9}$$

Define $\psi_1, \ldots, \psi_K, F : (0, 1) \to \mathbb{R}$ by

$$\psi_k : \quad 0.\overbrace{b_1 b_2 \ldots}^{\text{dec}} \mapsto \phi^{-1}\left(0.\overbrace{b_k b_{k+K} b_{k+2K} \ldots}^{\text{dec}}\right) \tag{13.10}$$

$$F : \quad x \mapsto f\left(\psi_1(x), \ldots, \psi_K(x)\right). \tag{13.11}$$

With the definitions (13.8) and (13.9), we can see that

$$\sum_{k=1}^{K} f_k(x_k) = 0.\overbrace{a_{\phi(x_1),1}\ldots a_{\phi(x_K),1}a_{\phi(x_1),2}\ldots a_{\phi(x_K),2}\ldots}^{\text{dec}}. \tag{13.12}$$

Clearly, the decimal representation of $\sum_{k=1}^{K} f_k(x_k)$ contains the full decimal representations of $x_1, \ldots, x_K$ and, therefore, allows for their full reconstruction. More specifically, the maps

$$(x_1, \ldots, x_K) \mapsto \sum_{k=1}^{K} f_k(x_k) \tag{13.13}$$

$$x \mapsto (\psi_1(x), \ldots, \psi_K(x)) \tag{13.14}$$

are inverses of each other and, therefore, (13.5) is satisfied, concluding the proof that $f$ is nomographic. □

In order to use the nomographic representation of a function in a wireless communication system, the inner functions $f_1, \ldots, f_K$ should be computed at the transmitter before the actual transmission, while the outer function $F$ should be implemented and evaluated at the receiver. Therefore, $f_1, \ldots, f_K$ are sometimes referred to as the pre-processing functions while $F$ is called a post-processing function. The summation is performed by the wireless channel due to its superposition property. If the receiver has access to $f_1(x_1) + \cdots + f_K(x_K)$, then from (13.13) and (13.14), it is clear that a full reconstruction of $x_1, \ldots, x_K$ is possible and in fact, this full reconstruction is used as an intermediate step in post-processing. On the other hand, we know from Example 13.1 that by the data processing inequality, we cannot hope for such a strategy to be effective if the channel under consideration has finite capacity. Indeed, in (13.9) we can see that arbitrarily significant digits of the transmitted values can be hidden in digits of arbitrarily low significance in the real number that is transmitted over the channel and, therefore, even a channel noise of extremely low power can cause arbitrarily strong disruptions.

It appears that in order to apply a nomographic representation to an OTA computation problem, Definition 13.4 is not strong enough and we need to impose additional constraints on the functions $f_1, \ldots, f_K, F$. Indeed, a famous result [10, 46] states that every continuous function $f : \mathbb{R}^K \to \mathbb{R}$ can be written as a sum of $2K + 1$ functions with continuous nomographic representations,[6] giving a positive answer in part to the question posed by Hilbert as the thirteenth problem in his list of unresolved mathematical problems of the twentieth century [41]. If there was a result implying that for every algebraic function, there is a nomographic representation consisting only of algebraic functions, this would give a positive answer to the as-of-yet unresolved part of Hilbert's thirteenth problem. We can, therefore, expect that proving such a result would be very hard.[7] Another result worth noting in this context is that the set of functions with a continuous nomographic representation is nowhere dense in the space of continuous functions [19]. This provides another piece of evidence that generic nomographic representations suitable for OTA computation may not exist.

A pragmatic way to proceed in light of these difficulties is to attempt to find a subclass of functions that is small enough to permit nomographic representations which are suitable for use with noisy communication systems and at the same time large enough to contain most functions of interest in practical OTA computation problems.

We conclude this section with a brief summary of papers that propose approaches to the OTA computation problem for functions particularly relevant to applications for consensus problems in wireless networks and ML over wireless channels. Goldenbaum and Stanczak [35] presents a scheme that is able to deal with imperfect synchronization and the presence of fading in OTA computation; extensive theoretical analyses for the asymptotic case is provided for the arithmetic and geometric mean functions. In [65], under the assumption of known fading coefficients at the transmitter, a similar scheme is used for computing the sign of a weighted sum which is the decision function of a linear support vector machine used for classification. As a result, the authors obtain a distributed binary classification scheme that is highly efficient in massively sized wireless networks. In the more recent work [49], under the assumption that the sources are independent and the channel state is known at both the receiver and the transmitter, the authors derive analog OTA computation schemes for sums that are optimal in terms of mean square error. In the case of i.i.d. Gaussian sources the authors of [25] show how to OTA compute sums over fading channels where the channel state information is available neither at the transmitter nor the receiver.

[6] A continuous nomographic representation is a nomographic representation that consists only of continuous functions $f_1, \ldots, f_K, F$.

[7] Hilbert even hypothesized that the correct answer to the question would be negative [41, 42], which was, however, partly disproven in [10, 46].

### 13.2.3 Analog Over-the-Air Computation as a Compressed Sensing Problem

While an important motivation for studying the problem of OTA computation is the prospective application to distributed Compressed Sensing (see Sect. 13.3.3), the problem of analog OTA computation itself can also be viewed as a variation of a compressed sensing problem.

Consider

$$Y = E \cdot X + N, \tag{13.15}$$

where $E \in \mathbb{C}^{M\times K}$, $X \in \mathbb{C}^{K}$, $Y \in \mathbb{C}^{M}$ and $N$ is a random vector ranging over $\mathbb{C}^{M}$. A classical Compressed Sensing problem (cf., e.g., [28]) would be to recover a sparse signal vector $X$ from the measurement $Y$ under the knowledge of $E$ and the statistics of $N$ in the regime $K \gg M$.

Considering the structure of the pre-processors that we use in the achievability results of Theorems 13.2 and 13.3, the signal at the receiver can be represented as $Y$ in (13.15), where $K$ is the number of transmitters, $M$ is the number of channel uses, $X$ is a shifted and rescaled version of the distributed data at the transmitters, $N$ is the additive channel noise and

$$E = \begin{pmatrix} U_1(1)H_1(1) & \cdots & U_K(1)H_K(1) \\ \vdots & & \vdots \\ U_1(M)H_1(M) & \cdots & U_K(M)H_K(M) \end{pmatrix}, \tag{13.16}$$

where $H_k(m), k = 1, \ldots, K, m = 1, \ldots, M$ denote the channel fading coefficients and $U_k(m), k = 1, \ldots, K, m = 1, \ldots, M$ represent an additional randomization which is artificially introduced by the transmitters in pre-processing. In the case of our proposed schemes, these randomization coefficients are chosen uniformly from the set $\{-1, 1\}$.

This formulation, while it is quite close to the Compressed Sensing problem, also exhibits a few important differences:

- The signal $X$ is not sparse in general. However, the goal of the receiver is not to reconstruct $X$ in its entirety, but rather some function of $X$, which needs not be linear. The function value that needs to be recovered by the receiver is a linear scalar and, therefore, only one-dimensional, which is a condition that comes very close to the usual sparseness condition in Compressed Sensing. Studying a problem in Compressed Sensing where the objective is not to recover the full signal vector from the measurements but rather some function of it (e.g., a classification result or a parameter estimation) is something that has also been done before (e.g., [44, 77]).
- The channel coefficients are random with only statistical information available at the receivers. The values $U_k(m), k = 1, \ldots, K, m = 1, \ldots, M$ can, however, be

freely chosen at the transmitters. Therefore, the matrix $E$ in this scenario consists of two parts, one of which is unknown and the other part can be designed. This is in contrast to the usual Compressed Sensing scenario where the whole sensing matrix can usually be designed or is at least known for the purpose of the sparse recovery procedure.

Because of these differences, the proofs of our results surveyed in Sect. 13.4 call for the use of mathematical tools that are different from those normally applied in Compressed Sensing.

## 13.3 Applications of Over-the-Air Computation

OTA computation has potential applications in every setting in which such a large number of wireless devices share constrained wireless resources that it becomes inefficient or even infeasible to exclusively use traditional scheduling and separate decoding of all transmitted information before it is post-processed at the receiver. Furthermore, even if the available resources are tremendous, but the number of participating devices is so large that traditional scheduling becomes prohibitively expensive, OTA computation can be a useful tool to solve the problem. On the other hand, it inherently fuses concepts that have traditionally been separate in communication systems. We have already discussed the point that from an information-theoretic perspective, it is a joint source-channel approach that breaks with the traditional separation paradigm. But also from the perspective of network architecture, it means using schemes on the physical layer that are at least in part tailored to specific applications, and traditional methods of scheduling and routing have to be adapted to be compatible. Therefore, OTA computation can be seen as a cross-layer approach that encompasses the entire network stack from the application layer all the way down to the physical layer. While the pre- and post-processing schemes can be proposed in such a generic manner that they can in principle be used for a large variety of potential applications, they still need to be carefully adapted to each one. There are two main fields of application that have recently motivated the development of OTA computation schemes, namely distributed OTA Machine Learning and consensus algorithms. In this section, we give a brief overview of these two applications.

### *13.3.1 Distributed Machine Learning*

In this subsection, we take a look at distributed ML, in particular, Federated Learning (FL), describe how this field branches into Vertical FL (VFL) and Horizontal FL (HFL) and cite a few examples from the literature that approach FL

problems with OTA computation methods. First, we need to define what ML is for the sake of this chapter, and we follow the formalism in [69].

**Definition 13.5** A *statistical learning problem* is a tuple $(\mathcal{X}, \mathcal{Y}, \mathcal{P}, L)$, where

- the *feature alphabet* $\mathcal{X}$ is a Polish space (usually a high-dimensional Euclidean space),
- $\mathcal{Y} \subseteq \mathbb{R}$ is called the *label alphabet*,
- $\mathcal{P}$ is a probability measure on $\mathcal{X} \times \mathcal{Y}$,
- $L : \mathcal{X} \times \mathcal{Y} \times \mathbb{R} \to [0, \infty)$ is called the *loss function*.

In the usual application setting, only the feature and label alphabets and the loss function are known about the statistical learning problem, while information about $\mathcal{P}$ is only known indirectly through a training sample.

**Definition 13.6** Given a statistical learning problem $(\mathcal{X}, \mathcal{Y}, \mathcal{P}, L)$, a training sample of length $N$ is a sequence $(x_n, y_n)_{n=1}^N \in \mathcal{X}^N \times \mathcal{Y}^N$ where each $(x_n, y_n)$ is drawn i.i.d. according to $\mathcal{P}$.

The objective in solving a statistical learning problem is to find an ML model which can make predictions about the labels of newly drawn samples of $\mathcal{P}$, given only the features. An ML model is a mathematical object which provides, given a set of parameters, a labeling function. Examples of ML models are neural networks, support vector machines, and decision trees.

**Definition 13.7** Given a statistical learning problem $(\mathcal{X}, \mathcal{Y}, \mathcal{P}, L)$, a *labeling function* is a function $f : \mathcal{X} \to \mathbb{R}$. A labeling function induces a *risk* (sometimes also called *loss*) $\mathcal{R}_{L,\mathcal{P}} := \mathbb{E}_{\mathcal{P}} L(X, Y, f(X))$, where $(X, Y)$ is the pair of random variables ranging over $\mathcal{X} \times \mathcal{Y}$ and distributed according to $\mathcal{P}$.

Typically, the objective is to exploit the indirect knowledge that we have about $\mathcal{P}$ through the training sample to obtain a labeling function with low risk, which is usually the measure for how well we have solved the statistical learning problem. To this end, a training procedure for a given ML model takes a training sample as its input and outputs parameters for the ML model. Therefore, in conjunction with the model, it maps training samples to labeling functions.

Distributed ML studies cases of statistical learning problems where some of the information about the statistical learning problem or the training sample are only known at certain locations in a network. Although there are possibilities for communication between the agents in the network, there are application-specific reasons for not transmitting the entire information to a central point. One particular instance of Distributed ML is called Federated Learning (FL) [47]. In FL, the initial main reason for not transmitting all the available information to a central point and then solving the problem in the traditional way is to preserve the privacy of the users from whom the training data is collected,[8] but communication efficiency also plays

[8] A major motivation for introducing the FL framework was Gboard, a software made by Google which is used as the default keyboard on many Android devices [50].

an increasingly important role. FL can be further categorized into Horizontal FL (HFL) and Vertical FL (VFL) [75].

In HFL, each agent $k$ out of a total of $K$ agents in the system sees only a subsequence of the training sample $(x_{n_{k,i}}, y_{n_{k,i}})_{i=1}^{N_k}$. In principle, it is possible for each agent to train its own local ML model based on the locally available training sample. Depending on the application at hand, however, this can incur several difficulties:

- The locally available training subsamples may simply be too small to train an ML model and obtain an acceptable risk.
- The way in which the locally available training subsamples are drawn from the overall training sample may be such that the subsamples are not i.i.d. or do not follow $\mathcal{P}$ [81]. For instance, it is common for the subsamples to be biased towards certain labels in a way the overall training sample is not.

Distributed optimization algorithms can be used to carry out the training in a decentralized manner. They make, either at one central point or everywhere in the network, a trained ML model available that benefits from the whole training sample without transmitting it through the network in its entirety. There is a huge body of recent research (cf., e.g., [1, 5–9, 21, 39, 63, 67, 68, 70, 76, 79, 82–84], and the references therein) into ways to perform distributed optimization algorithms such as stochastic gradient descent exploiting OTA computation. This approach can achieve fundamentally more favorable scaling laws than would be possible otherwise.

In VFL, the data is distributed in a different way: In a system with $K$ agents, the statistical learning problem has a feature alphabet $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_K$ that is a Cartesian product of $K$ feature spaces. A feature $x \in \mathcal{X}$ can, therefore, be written as a tuple $x = (x_1, \ldots, x_K)$ and the training sample is of the form $((x_{1,n}, \ldots, x_{K,n}), y_n)_{n=1}^{N}$ where each agent $k$ has only the local training sample $(x_{k,n}, y_n)_{n=1}^{N}$. Correspondingly, when training is complete and a label needs to be estimated, each agent $k$ sees only the projection to $\mathcal{X}_k$ of the observed feature. Since the labeling function has the whole feature space $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_K$ as its domain, the arguments to compute it are not available at any single point in the network and it is, therefore, natural to attempt to compute the labeling function OTA. So there are two important research question in OTA-VFL:

- Given a training sample that is distributed as described above, how can we carry out a distributed training procedure exploiting OTA computation that scales better than linear in the number of agents involved?
- Given the trained model (which also is available only in a distributed manner), how can we compute the labeling function using the OTA approach?

The first question is quite similar to the main research question in OTA-HFL and there is some hope that tools from this field could be suitably adapted. The second question is more specific to the VFL scenario, and we note that many standard ML labeling functions naturally take the form of (weighted) sums. Examples are layers of neural networks (the activation function can be evaluated afterwards in

post-processing if necessary) and the linear support vector machines that have been used for OTA-VFL in [65]. Contrary to the OTA-HFL, there does not appear to be a large body of research on OTA-VFL. Besides [65] and our own results which we survey in Sect. 13.5, we are not aware of any works that propose to leverage OTA computation in a VFL scenario.

### *13.3.2 Consensus Over Wireless Channels*

Consensus problems deal with combining opinions of participating agents to achieve an agreement that encompasses their information about or subjective assessments of an object. They have originally appeared as statistical problems in which the opinions are probability distributions which have to be combined to form a consensus distribution. In [26] this is illustrated as a horse race betting problem where the agents' opinions are probability distributions on which horse will win the race. They place their bets according to these opinions and the overall track's odds that result from these bets are considered the consensus which in a certain way combines all the participating agents' opinions. The problem has subsequently been stated as one of combining various experts' opinions and researched extensively to aid with decision making in the context of management sciences (see, e.g., [29, 71], and the references therein).

The research on this theory has later been applied to problems of multisensor fusion and pattern recognition [12] and since found a multitude of other applications in engineering sciences [61]. In some of the engineering applications the nature of the difficulty of the problem has shifted significantly: Often, an opinion is simply a real number or vector and the way the opinions have to be combined to form the consensus is fully prescribed by the application at hand and is fairly simple compared to the original consensus problem: For instance, the consensus can be the arithmetic average (with applications, e.g., in formation control and flocking of autonomous vehicles [60]) or the maximum of the opinions (examples for applications include task assignment [17] and traffic automation [55]). In these applications, the challenge is that it is infeasible to aggregate the opinions in a central point because the communication cost or the time delay incurred would be prohibitive. In these cases, distributed consensus algorithms are used that seek to make the consensus value available to agents in a large network with a minimum of communication required between the agents [61].

In many applications, the communication links between the agents are wireless channels, and indeed, several agents can be linked to another agent via a broadcast or multiple-access wireless channel. Some works that exploit these properties to reach average or maximum consensus in a way that is more communication-efficient than would be possible with point-to-point communication are [43, 54–56]. We expect that theoretical analysis of OTA computation techniques could serve as a building block to enhance the efficiency and, in particular, the scaling behavior of the communication cost in the number of participating agents. Moreover, this way

it would be possible to provide additional theoretic error guarantees for consensus schemes that exploit the superposition of signals in the wireless channel. In [3], we have proposed a maximum consensus scheme which leverages analog OTA computation of sums to make the maximum of the agents' opinions available at the receiver in a multiple-access wireless channel with no fading but with additive noise. The OTA computation schemes we survey in this chapter can be used to extend these results to channels exhibiting fast fading [13]. It is, in particular, worth noting that the scheme proposed in [3] can OTA compute the maximum of the agents' opinions in a wireless channel although we do not expect the maximum function to satisfy Definition 13.8. This is achieved not through a single OTA computation but through a multi-step protocol that alternates between analog OTA computation of sums and digitally coded broadcast communication. We believe, therefore, that such multi-step protocols are a potentially promising approach to computing also other functions for which a representation as in Definition 13.8 is not known. This is at the cost of higher system and communication complexity, but a favorable scaling of communication cost in extremely large networks would be retained.

### *13.3.3 Compressed Sensing*

As detailed in Sect. 13.3.1, OTA computation can be applied to distributed ML in wireless networks, both in the Horizontal ML scenario and in the Vertical ML scenario. On the other hand, ML is a tool that is itself often used to solve Compressed Sensing problems (e.g., [2, 64, 73]). Therefore, one potential application of OTA computation is to apply it to distributed ML algorithms for sparse signal recovery, which would yield distributed Compressed Sensing schemes for wireless networks. These could, for instance, be used to facilitate communication tasks such as channel estimation and beamforming. The main advantage of the OTA-ML-aided approach to Compressed Sensing is the scaling behavior of the communication cost with the number of transmitters, which is often logarithmic or even constant. For the massive wireless networks that are envisioned for the future, this scaling behavior is not just advantageous for the conservation of communication resources, but can be expected to become absolutely essential given the expected growth of the number of wireless devices.

Works such as [9] use an OTA version of distributed Stochastic Gradient Descent (SGD) to solve the empirical risk minimization problem encountered in the training of ML models. Therefore, another prospective approach to exploit OTA computation for distributed Compressed Sensing would be to extend the SGD method so that it can be used to solve Compressed Sensing optimization problems such as LASSO or elastic net directly. This is in contrast to an approach that formulates an ML model to approach the Compressed Sensing problem and then solves the associated risk minimization problem OTA. To this end, promising research directions could be the extension of the OTA-SGD method so that it can be used to execute, e.g.,

subgradient and proximal gradient methods in a distributed fashion in wireless networks.

## 13.4 Distributed Function Approximation in Wireless Channels

In this section, we discuss our Distributed Function Approximation (DFA) scheme which we proposed in [13] and extended in [31, 33]. The goal in introducing it was to provide a flexible framework that can deal with such a large class of wireless channels that the scheme would be robust to departures from common assumptions on the system model such as Gaussianity of the fading and noise. At the same time, the class of functions for OTA computation should contain the most relevant ones in current applications (which are mainly weighted sums). It should also be large enough to provide flexibility and make the DFA scheme applicable in scenarios where functions that have not yet received much attention are computed OTA. Another important consideration in the design of the scheme was the distribution of the sources. Many existing works on OTA computation assume a particular source distribution for their theoretical analysis, and usually require that the transmitted values are independently distributed between the transmitters. Since this requirement is extremely difficult to check in practice, we have decided to not model the sources stochastically. Instead, we show that the bound on the approximation error is satisfied uniformly over all possible values of the sources. This yields a worst-case analysis with theoretically proven error guarantees that are valid for every distribution of the sources, even if there is arbitrary correlation between them. In addition, the error bounds are nonasymptotic in the sense that they are valid for any number of channel uses, not just for a sufficiently large one.

### *13.4.1 Class of Functions*

For the class of functions that can be computed OTA, it is important that they are not only nomographic (which as we have seen in Theorem 13.1 and its proof is a notion that is too weak in the presence of channel noise), but that they have a nomographic representation that is amenable to the processing of noisy values. We adapt Definition 13.4 to give a stronger notion of nomography, which means that it defines a smaller class of functions, but also provides a good basis on which error bounds can be argued.

**Definition 13.8 ([13, Definition 1])** Let $\mathcal{S}_1, \ldots, \mathcal{S}_K$ be measure spaces. We say that a function $f : \mathcal{S}_1 \times \cdots \times \mathcal{S}_K \to \mathbb{R}$ is in $\mathcal{F}_{\text{mon}}$ if there are measurable $f_1 : \mathcal{S}_1 \to \mathbb{R}, \ldots, f_K : \mathcal{S}_K \to \mathbb{R}$, which we call the *inner functions* and a measurable $F : \mathbb{R} \to \mathbb{R}$, which we call the *outer function*, such that all of the following hold:

- $f_1, \ldots, f_K, F$ are a nomographic representation of $f$; i.e., for all $s_1 \in \mathcal{S}_1, \ldots, s_K \in \mathcal{S}_K$,

$$f(s_1, \ldots, s_K) = F\left(\sum_{k=1}^{K} f_k(s_k)\right) \tag{13.17}$$

- $f_1, \ldots, f_K$ are bounded and there is a bounded measurable set $D_F \subseteq \mathbb{R}$ such that $f_1(\mathcal{S}_1) + \cdots + f_K(\mathcal{S}_K) \subseteq D_F$.
- There is a strictly increasing function $\Phi : [0, \infty) \to [0, \infty)$ with $\Phi(0) = 0$ and for all $x_1, x_2 \in D_F$,

$$|F(x_1) - F(x_2)| \leq \Phi(|x_1 - x_2|). \tag{13.18}$$

We call $\Phi$ an *increment majorant* of $F$.

The abbreviation *mon* in the subscript of $\mathcal{F}_{\text{mon}}$ refers to the existence of a monotonous increment majorant.

For a given $f \in \mathcal{F}_{\text{mon}}$, the inner and outer functions are not necessarily unique, however, when we consider a function $f \in \mathcal{F}_{\text{mon}}$, we implicitly fix a representation $f_1, \ldots, f_K, F$ of $f$ which satisfies Definition 13.8. With this representation fixed, we can define some additional properties.

**Definition 13.9** Given $f \in \mathcal{F}_{\text{mon}}$ (and a fixed representation $f_1, \ldots, f_K, F$), we let

$$\phi_{\min,k} := \inf_{s \in \mathcal{S}_k} f_k(s), \quad \phi_{\max,k} := \sup_{s \in \mathcal{S}_k} f_k(s). \tag{13.19}$$

The *total spread* of the inner part of $f$ is defined as

$$\bar{\Delta}(f) := \sum_{k=1}^{K} (\phi_{\max,k} - \phi_{\min,k}) \tag{13.20}$$

and the *maximum spread* is defined as

$$\Delta(f) := \max_{1 \leq k \leq K} (\phi_{\max,k} - \phi_{\min,k}). \tag{13.21}$$

As we will see below, the representations of functions in $\mathcal{F}_{\text{mon}}$ allow for the construction of pre- and post-processors in OTA computation that have provable error bounds on the quality of the receiver's estimate of $f(s_1, \ldots, s_K)$. Another very important property of $\mathcal{F}_{\text{mon}}$ is whether it contains the functions that are of interest in practical problems where OTA computation is applied. In the following, we list some examples of relevant subsets of $\mathcal{F}_{\text{mon}}$. More details on this can be found in [13, Section II-D].

- The special case where $F$ is the identity function. We call this class the class of *generalized linear functions* and it is the most important one for ML schemes since it contains weighted sums, but it is also significant for distributed consensus since it contains the arithmetic average function (a special case of a weighted sum).
- More generally, the special case where $F$ is Lipschitz-continuous and, more generally than this, the special case where $F$ is Hölder-continuous.
- For any $p \geq 1$ and compact $\mathcal{S}_1, \ldots, \mathcal{S}_K \subseteq \mathbb{R}$, the $p$-norm is in $\mathcal{F}_{\text{mon}}$.

### *13.4.2 System and Channel Model*

For our DFA schemes, we use the system model depicted in Fig. 13.1. The objective is to OTA compute a function $f(s_1, \ldots, s_K)$ in $\mathcal{F}_{\text{mon}}$ through a series of $M$ uses of a given channel. To this end, each transmitter $k$ uses a randomized pre-processing function $F_k^M$ which transforms the input value $s_k$ into an $M$-length sequence of channel input symbols. These sequences are superimposed in the channel to yield a sequence $Y^M$ of channel output symbols, which the receiver transforms to an estimate $\tilde{f}$ by applying a post-processing map $D^M$. In the DFA schemes we propose, for each $k$, the sequence $T_k^M$ is i.i.d. conditioned under $s_k$.

For the channel model, the basic idea is that our results should apply to fast-fading wireless channels. At the same time the error guarantees should be provably robust to common departures from the assumption that noise and fading are i.i.d. Gaussian (cf., e.g., [14, 51, 52]). We, therefore, assume that fading and noise are sub-Gaussian. This class of sub-Gaussian distributions contains besides Gaussian distributions all distributions with bounded support, which means that almost all practically relevant non-Gaussian disturbances are captured in this class (see [31, Section II-C] for details).

**Definition 13.10** For a real random variable $X$, we define

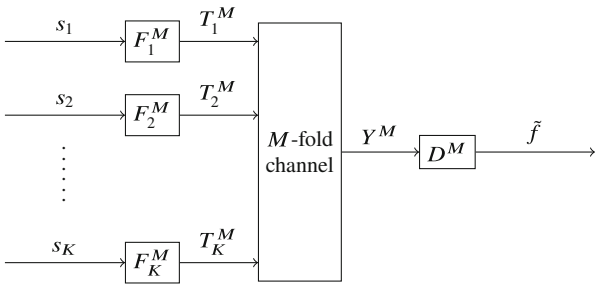



**Fig. 13.1** DFA System model

$$\tau(X) := \inf\left\{t > 0 : \forall \lambda \in \mathbb{R}\ \ \mathbb{E}\exp\left(\lambda(X - \mathbb{E}X)\right) \leq \exp\left(\lambda^2 t^2/2\right)\right\}. \tag{13.22}$$

If $\tau(X)$ is finite, we call $X$ a *sub-Gaussian* random variable, and $\tau(X)$ its *sub-Gaussian* norm.

We note that $\tau(\cdot)$ is indeed a semi-norm on the space of sub-Gaussian random variables [20, Theorem 1.1.2]. Examples of sub-Gaussian variables include:

- Gaussian variables: If $X$ is normally distributed with variance $\sigma^2$, we have $\tau(X) = \sigma$. This follows from the calculation of the moment-generating function of $X$.
- Bounded variables: If $|X - \mathbb{E}X| \leq c$ almost surely, then $\tau(X) \leq c$. This follows from Taylor's theorem [20, Example 1.1.2].

We consider a complex channel which is used $M$ times. The output of the channel at its $m$-th use is given by

$$Y(m) = \sum_{k=1}^{K} H_k(m)T_k(m) + N(m), \tag{13.23}$$

where the symbols are defined as follows:

- $T_k(m)$ is the complex channel symbol that transmitter $k$ transmits at the $m$-th channel use. We impose the peak power constraint $\forall k, m\ \ |T_k(m)|^2 \leq P$.
- $H_k(m)$ are the fading coefficients. We assume that they are centered and that their complex components have variance 1 and a sub-Gaussian norm uniformly bounded by $\sigma_F$.
- $N(m)$ is the additive noise at channel use $m$. We assume that it is centered with the sub-Gaussian norm of its complex dimensions uniformly bounded by $\sigma_N$. The total noise power $\sum_{m=1}^{M} \mathbb{E}\,|N(m)|^2$ is assumed to be known at the receiver.

### *13.4.3 The Case of Independent Fading and Noise*

In this section, we present our results from [13]. In addition to the assumptions made in Sect. 13.4.2, we assume that all the instances of fading and noise are stochastically independent, but we do not assume that they are necessarily identically distributed; we only need the uniform bounds on their sub-Gaussian norms stated in Sect. 13.4.2. Additionally, no instantaneous channel state information needs to be known at the transmitter or the receiver, only the general statistical properties stated in Sect. 13.4.2. Under these assumptions, we have the following result.

**Theorem 13.2 ([13, Theorem 1])** *Let $f \in \mathcal{F}_{\text{mon}}$ and fix a representation as in Definition 13.8. This representation induces $\Phi$ and, via Definition 13.9, the quantities $\Delta(f)$ and $\bar{\Delta}(f)$.*

*Then there are pre- and post-processing functions so that for every $\varepsilon > 0$, the absolute error of the estimate $\tilde{f}$ of $f(s_1, \ldots, s_K)$ satisfies*

$$\mathbb{P}\left(\left|\tilde{f} - f(s_1, \ldots, s_K)\right| \geq \varepsilon\right) \leq 2\exp\left(-\frac{M\Phi^{-1}(\varepsilon)^2}{2L\Phi^{-1}(\varepsilon) + 8L^2K}\right) + 2\exp\left(-\frac{M\Phi^{-1}(\varepsilon)^2}{2F\Phi^{-1}(\varepsilon) + 4F^2}\right), \tag{13.24}$$

*where*

$$L = \Delta(f)\sigma_F^2 \tag{13.25}$$

$$F = 3\sigma_F^2\bar{\Delta}(f) + 4\sigma_N\sigma_F\sqrt{\frac{\Delta(f)\bar{\Delta}(f)}{P}} + \frac{2\sigma_N^2\Delta(f)}{P}. \tag{13.26}$$

The proof of this theorem in [13] gives explicit pre- and post-processing operations. Intuitively, the pre-processing works by applying the inner functions $f_1, \ldots, f_K$ from Definition 13.8 to the distributed data followed by shifting and rescaling steps which ensures that the transmit power constraint is satisfied and finally applies a random phase shift to mitigate the effects of the unknown fading. The post-processor averages its received signal in such a way that it mitigates the random distortions incurred by the unknown fading and noise, then inverts the shifting and rescaling operations that were performed in pre-processing and finally applies the outer function $F$ from Definition 13.8.

It is of particular interest how these error bounds depend on $K$ since the unfavorable scaling of separation-based approaches with $K$ is one of the main motivations for using analog OTA computation in the first place. However, it is tricky to answer this question for general $f$, since $f$ itself has to change with $K$ and depending on how it does, the spreads introduced in Definition 13.9 can scale in a different way with $K$. This is a question of how $f$ scales its values with $K$, as the following two examples demonstrate.

*Example 13.2 ([13, Example 1])* Consider the sum function

$$f : [0, 1]^K \to \mathbb{R},\ (s_1, \ldots, s_K) \mapsto s_1 + \cdots + s_K. \tag{13.27}$$

Clearly, $f \in \mathcal{F}_{\text{mon}}$. Substituting the spreads of $f$ into Theorem 13.2 (see [13, Example 1] for details), we obtain the error bound

$$\mathbb{P}\left(\left|\tilde{f} - f(s_1, \ldots, s_K)\right| \geq \varepsilon\right) \leq 2\exp\left(-\frac{M\varepsilon^2}{4\sigma_F\varepsilon + 32K\sigma_F^4}\right) + 2\exp\left(-\frac{M\varepsilon^2}{4F\varepsilon + 16F^2}\right), \tag{13.28}$$

where

$$F = 3\sigma_F^2 K + \frac{4\sigma_N \sigma_F \sqrt{K}}{\sqrt{P}} + \frac{2\sigma_F^2}{P}. \tag{13.29}$$

Therefore, if we want to achieve a bounded approximation error in the case $K \to \infty$, we have to let $M$ grow proportionally with $K^2$.

*Example 13.3 ([13, Example 1])* Consider the arithmetic average function

$$f : [0, 1]^K \to \mathbb{R}, \ (s_1, \ldots, s_K) \mapsto \frac{s_1 + \cdots + s_K}{K}. \tag{13.30}$$

Clearly, $f \in \mathcal{F}_{\text{mon}}$. The error bound of Theorem 13.2 is in this case (see [13, Example 1] for details)

$$\begin{aligned} &\mathbb{P}\left(\left|\tilde{f} - f(s_1, \ldots, s_K)\right| \geq \varepsilon\right) \\ &\qquad \leq 2\exp\left(-\frac{MK\varepsilon^2}{4\sigma_F^2\varepsilon + 32\sigma_F^4}\right) + 2\exp\left(-\frac{M\varepsilon^2}{4F\varepsilon + 16F^2}\right), \end{aligned} \tag{13.31}$$

where

$$F = 3\sigma_F^2 + \frac{4\sigma_N \sigma_F}{\sqrt{PK}} + \frac{2\sigma_N^2}{PK}. \tag{13.32}$$

Therefore, we can achieve a bounded approximation error in the case $K \to \infty$ without having to let $M$ grow with $K$ at all.

### 13.4.4 The Case of Correlated Fading and Noise

In this section, we present our analog OTA computation results from [31, 33]. Compared to [13] where both the fading and the noise had to be independent, we introduced the possibility of correlations so that our results would also cover, e.g., the common case of a block fading channel (which exhibits correlated fading) or a channel with bursty additive noise (which exhibits correlated noise).

For $z \in \mathbb{C}$, we denote the real part of $z$ with $z^r$ and the imaginary part with $z^i$. We define

$$N := (N^r(1), N^i(1), \ldots, N^r(M), N^i(M))^T \tag{13.33}$$

and

$$H := (H(1), \ldots, H(2M))^T, \tag{13.34}$$

where for $m = 1, \ldots, M$,

$$H(2m-1) := ({H_1}^r(m), \ldots, {H_K}^r(m)) \tag{13.35}$$

$$H(2m) := ({H_1}^i(m), \ldots, {H_K}^i(m)). \tag{13.36}$$

Given these notations, our system model assumptions are

$$H = AR, \tag{13.37}$$

$$N = BR, \tag{13.38}$$

where $R$ is a vector of $2KM+2M$ independent random variables with sub-Gaussian norm at most 1, $A \in \mathbb{R}^{2MK \times 2MK+2M}$ and $B \in \mathbb{R}^{2M \times 2MK+2M}$. This replaces the independence assumption made in Sect. 13.4.3, while the assumptions made in Sect. 13.4.2 are still in place.

For the special case of i.i.d. standard Gaussian $R$, this linear transformation model specializes to arbitrarily dependent Gaussian fading and noise. It is, therefore, a straightforward generalization of the Gaussian case to replace $R$ with a sub-Gaussian vector that covers fading and noise which can exhibit arbitrary correlations, albeit not an arbitrary stochastic dependence structure.

Additionally, we need to quantitatively capture how strong the correlation between users is in the fading. To this end, we make the following definition.

**Definition 13.11** The fading is called *user-uncorrelated* if for every $k_1, k_2 \in \{1, \ldots, K\}$, for every $j \in \{r, i\}$ and for every $m \in \{1, \ldots, M\}$, the random variables $H^j_{k_1}(m)$ and $H^j_{k_2}(m)$ are independent.

We do not fully restrict the channel model under consideration to channels with user-uncorrelated fading, however. Instead, we assume that the matrix representing the fading is decomposed as

$$A = A_i + (A - A_i), \tag{13.39}$$

where $A_i$ is a matrix that would result in user-uncorrelated fading if substituted into the dependence model described above. This can immediately be seen to not introduce any additional restriction in the model since $A_i$ can, e.g., be chosen all-zero. However, the impact that the user-uncorrelated part $A_i$ and the user-correlated part $A - A_i$ have on the resulting error bounds differs, as can be observed in the following result which we have under the preceding assumptions.

**Theorem 13.3 ([31, 33, Theorem 1])** *Let $f \in \mathcal{F}_{\text{mon}}$ and fix a representation as in Definition 13.8. This representation induces $\Phi$ and, via Definition 13.9, the quantities $\Delta(f)$ and $\bar{\Delta}(f)$.*

*Then there are pre- and post-processing functions so that for every $\varepsilon > 0$, the absolute error of the estimate $\tilde{f}$ of $f(s_1, \ldots, s_K)$ satisfies*

$$\mathbb{P}\left(\left|\tilde{f} - f(s_1, \ldots, s_K)\right| \geq \varepsilon\right) \leq 2\exp\left(-\frac{M\Phi^{-1}(\varepsilon)^2}{16F + D + 4\Phi^{-1}(\varepsilon)L}\right) + 2\exp\left(-\frac{M\Phi^{-1}(\varepsilon)^2}{256F + 32\Phi^{-1}(\varepsilon)L}\right), \tag{13.40}$$

*where*

$$L = \left(\sqrt{\bar{\Delta}(f)}\|A\| + \sqrt{\frac{\Delta(f)}{P}}\|B\|\right)^2 \tag{13.41}$$

$$F = L\left(\sqrt{\frac{\bar{\Delta}(f)}{M}}\|A\|_F + \sqrt{\frac{\Delta(f)}{PM}}\|B\|_F\right)^2 \tag{13.42}$$

$$D = \left(4\sqrt{2M}\bar{\Delta}(f)\|(A + A_i)(A - A_i)^T\| + 4\frac{\Delta(f)}{\sqrt{PM}}\|AB^T\|_F\right)^2. \tag{13.43}$$

*Remark 13.1* $D$ is zero for an important subclass of cases. The first summand in (13.43) captures the effect of user-correlated fading which we have discussed above. In the case of user-uncorrelated fading it is 0. The second summand captures the effect of correlation between fading and noise and it is 0 in case fading and noise are stochastically independent.

A commonly considered example for a situation where this remark applies is the case of block fading. We denote the map that rounds down to the nearest integer with $\lfloor\cdot\rfloor$ and say that the channel follows a block fading model with block length $\beta$ if both of the following hold for all $k_1, k_2, m_1, m_2$:

- If $k_1 \neq k_2$ or $\lfloor(m_1 - 1)/\beta\rfloor \neq \lfloor(m_2 - 1)/\beta\rfloor$, the fading coefficients $H_{k_1}(m_1)$ and $H_{k_2}(m_2)$ are uncorrelated.
- Otherwise, the fading coefficients $H_{k_1}(m_1)$ and $H_{k_2}(m_2)$ are almost surely equal.

With this definition, we have the following immediate consequence of Theorem 13.3.

**Corollary 13.1 ([31, Corollary 3])** *If the fading is independent between users and for each user, we have a block fading model of block length $\beta$, the error bound is (13.40), where $D = 0$ and*

$$L = \left(\sqrt{\bar{\Delta}(f)\beta}\sigma_F + \sqrt{\frac{\Delta(f)}{P}}\sigma_N\right)^2$$

$$F = L\left(\sqrt{2K\bar{\Delta}(f)}\sigma_F + \sqrt{\frac{2\Delta(f)}{P}}\sigma_N\right)^2.$$

## 13.5 DFA Applications to VFL

In this section, we give an overview of the OTA-computed labeling functions for VFL scenarios that we propose in our works [13, 31].

The first one is a generalization of the idea in [65] where the labeling function of a linear support vector machine for binary classification is used. We note that it is possible to generalize this to the case of support vector machines with additive kernels [22]. Looking specifically at regression problems and their labeling functions, we then give a theoretically proven error guarantee on the additional error incurred through the OTA computation. However, the training has to be carried out offline with all of the training data available at a central point. Therefore, this approach is only suitable in cases in which the models do not or only very infrequently have to be retrained.

In [31], we look at a different way of constructing OTA-computed labeling functions for the case of binary classification based on the idea of adapting the Boosting technique (cf., e.g., [53, Chapter 6]). A Boosting labeling function has the form

$$g := \sum_{k=1}^{K} \alpha_k g_k, \tag{13.44}$$

where $\alpha_1, \ldots, \alpha_K$ are nonnegative weights and $g_1, \ldots, g_K$ are base labeling functions which can be from any ML model. The usual reason for employing Boosting is that sometimes, it is hard to construct a single ML model with a near-zero classification error while it is easier to construct many ML models with a classification error that is only slightly better than random guessing. One boosting algorithm which constructs a classifier of the form (13.44) that achieves an arbitrarily small error in this case as long as $K$ can be chosen large enough is called *AdaBoost* [53, Theorem 6.1]. It is apparent in (13.44) that $g$ satisfies Definition 13.8 and is, therefore, in $\mathcal{F}_{\text{mon}}$, so our OTA computation techniques are applicable. The idea is, therefore, to not necessarily use different ML models as the base labeling functions, but instead make them different by training each of them based on the locally available features. In [31], we propose two different ways of combining the base labeling functions:

- Train each base labeling function only locally and use equal weights $\alpha_1 = \cdots = \alpha_K = 1$.
- Use a straightforward adaptation of the AdaBoost technique to train the base labeling functions and calculate the weights.

The first possibility has the advantage that it does not rely on any communication at all during the training phase and is, therefore, suitable in cases in which the models have to be frequently retrained or the training sample is particularly large. The second possibility can for some problems yield better labeling functions: If, e.g., some of the participating agents see features that are stochastically independent (or close to independent) from the labels, this technique is better equipped to deal with this situation as it will simply adjust their weights to low values. The disadvantage of the second approach is that there is currently no distributed training procedure that leverages OTA computation. This results in a bad scaling behavior in extremely large networks. However, we do propose a distributed training procedure based on point-to-point and broadcast communication that is better in terms of communication efficiency than transmitting the whole training sample to a central place. Therefore, it is primarily suitable for networks of moderate size where frequent re-training is not necessary. Both of these approaches have the advantage that they are agnostic to the ML models they use as base labeling functions. Therefore, they provide a flexible approach to VFL classification that can be combined with other building blocks depending on the application at hand. It is even possible to use different ML models at different agents depending, e.g., on their computational capabilities. In [31], we give a theoretically guaranteed bound on the additional loss incurred by computing $g$ OTA. Moreover, we complement this in [31] with a numerical example where we combine decision tree classifiers OTA.

## 13.6 Security in OTA Computation

OTA computation schemes carry the promise that they can improve communication efficiency so dramatically in many cases of practical interest that they can be seen as an enabler for applications in massive wireless networks for which the communication cost or the time delay incurred would otherwise be prohibitive. However, there is also a flip side that has the potential to hinder widespread adoption: Some tools that enhance the properties of communication and are frequently used as building blocks in communication systems inherently rely on the principle of source-channel separation. Therefore, they cannot be adapted to work in a scenario where a joint source-channel approach is taken such as in OTA computation. One example of such a building block that is particularly important in modern communication systems is cryptography. OTA communication schemes as described in Sect. 13.4 are vulnerable to a number of attacks such as malicious transmitters participating in the scheme or attackers eavesdropping on the transmission, and it is unclear whether

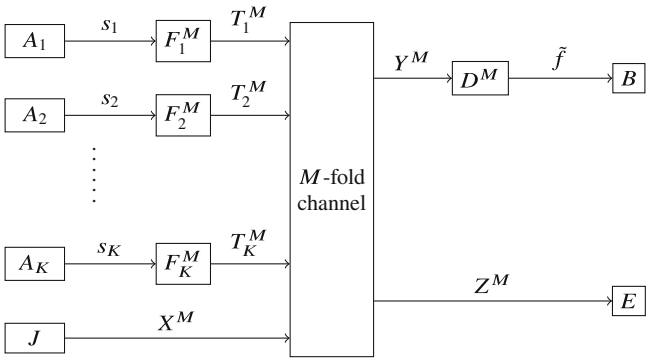



**Fig. 13.2** System model for distributed function approximation with security constraints

and how state-of-the-art cryptographic security could be adapted to defend against such threats.

At least for the latter kind of threat – attackers eavesdropping on the communication – information-theoretic security, while not adaptable in a straightforward fashion, provides a set of tools with which a defense can be developed. The ultimate goal in this direction should be full semantic security [11]. As a first step, we have proposed in [32] to extend the system model with a jammer as depicted in Fig. 13.2. This shows how information-theoretic security tools can be adapted to the OTA computation setting. The jammer can increase the variance of the eavesdropper's estimate of the quantity of interest, but not fully prevent it from obtaining an estimate.

The key assumption we make is that the received jamming signal must be stronger for the legitimate receiver than it is for the eavesdropper. This way, the legitimate receiver can exploit the dependencies which we carefully introduce into the jamming signal to reconstruct it exactly. To the eavesdropper, the received signal is almost equivalent to an i.i.d. jammed transmission. With the knowledge of the full jamming signal, the legitimate receiver can then cancel it from its received signal or at least mitigate its impact. The approximation of the OTA-computed function value is out of scope of the scheme and can be carried out, e.g., with the tools we survey in Sect. 13.4.

While our results on OTA computation rely heavily on the particular structure of wireless channels, the class of channels for which the reconstruction of the jamming signal is possible is much more general. This will become apparent in the following.

We are not aware of a similar system model having been proposed before for OTA computation, but we draw heavily from existing tools in information theory.

### 13.6.1 Information-Theoretic Preliminaries

Before we can state and discuss the result, we have to introduce some information-theoretic terminology and notation. A channel $W$ is a stochastic kernel mapping from an input alphabet to an output alphabet. Given a channel $W$ and a probability distribution $P$ on its input alphabet, we denote the induced joint input-output distribution with $Q_{P,W}$, the marginal on the output alphabet with $R_{P,W}$ and define the information density as

$$\mathbf{i}_{P,W}(x^M; y^M) := \frac{1}{M} \sum_{j=1}^{M} \log \frac{dW(x_j, \cdot)}{dR_{P,W}}(y_j). \tag{13.45}$$

The mutual information is defined as

$$\mathbf{I}_{P,W} := \mathbb{E}_{Q_{P,W}} \mathbf{i}_{P,W}(X; Y), \tag{13.46}$$

where $X$ and $Y$ denote the input and output random variable of the channel, respectively. We denote the $M$-fold product of a channel or a probability distribution with a superscript $M$.

A compound channel $(W_s)_{s \in \mathcal{S}}$ is a collection of channels. If a transmission is made through a compound channel, some $s \in \mathcal{S}$ is chosen in such a way that neither the transmitter nor the receiver know $s$ or the law according to which it is chosen, but it remains the same for all channel uses over the entire block length. Our result poses the requirement that the compound channel can be $(\delta, J)$-approximated. The formal definition can be found in [32, Definition 2], but we note here that many commonly considered classes of wireless channels such as Additive White Gaussian Noise channels and the fast-fading Gaussian channel have this property.. We use $W_{B(s_1,\ldots,s_K)}$ to denote the effective channel that maps the jammer's input $X$ to the output $Y$ of the legitimate receiver if the values of the transmitters are fixed at $s_1, \ldots, s_K$. Similarly, we use $W_{E(s_1,\ldots,s_K)}$ to denote the effective channel that maps the jammer's input $X$ to the output $Z$ of the eavesdropper if the values of the transmitters are fixed at $s_1, \ldots, s_K$.

A codebook $\mathcal{C}$ induces a jamming strategy as follows: The jammer picks a codeword from $\mathcal{C}$ uniformly at random and then transmits this codeword as the jamming signal.

### 13.6.2 Result and Discussion

We have now introduced the necessary notation to state our result for secure OTA computation.

**Theorem 13.4 ([32, Theorem 2])** *Let $P$ be a jammer input distribution. Suppose that for every $\delta > 0$, there is some $J(\delta)$ such that the compound channel $(W_{B(s_1,\ldots,s_K)})_{(s_1,\ldots,s_K)\in\mathcal{S}_1\times\cdots\times\mathcal{S}_K}$ can be $(\delta, J(\delta))$-approximated under $P$. Suppose further that for all $s_1 \in \mathcal{S}_1, \ldots, s_K \in \mathcal{S}_K$, the moment-generating function*

$$\mathbb{E}\exp(\lambda \cdot \mathbf{i}_{P,W_{E(s_1,\ldots,s_K)}}(X;Z)) \tag{13.47}$$

*of the information density exists and is finite at some point $\lambda > 0$. Given a number $R$ such that*

$$\sup_{s_1\in\mathcal{S}_1,\ldots,s_K\in\mathcal{S}_K} \mathbf{I}_{P,W_E(s_1,\ldots,s_K,\cdot,\cdot)} < R < \inf_{s_1\in\mathcal{S}_1,\ldots,s_K\in\mathcal{S}_K} \mathbf{I}_{P,W_B(s_1,\ldots,s_K,\cdot,\cdot)}, \tag{13.48}$$

*there are numbers $\gamma_1, \gamma_2 > 0$ such that for sufficiently large $M$, there is a codebook $\mathcal{C}$ such that under the jamming strategy induced by $\mathcal{C}$, the legitimate receiver $B$ can reconstruct the jammer's transmitted signal with an error not exceeding $\exp(-M\gamma_1)$, while the eavesdropper's output $\hat{R}_{W_E^M(s_1,\ldots,s_K),\mathcal{C}}$ satisfies*

$$\left\| \hat{R}_{W_E^M(s_1,\ldots,s_K),\mathcal{C}} - R^M_{P,W_E(s_1,\ldots,s_K)} \right\|_{\mathrm{TV}} < \exp(-M\gamma_2), \tag{13.49}$$

*where $\|\cdot\|_{\mathrm{TV}}$ denotes the total variation norm on signed measures.*

For a detailed discussion and a proof of this theorem, we refer the reader to [32]. Here, we briefly discuss the result and its impact and also summarize the information-theoretic ideas that are used to prove it.

Equation (13.47) is a technical requirement that is satisfied, for example, for all channels with finite alphabets and for all Gaussian channels whether or not they have fading. Equation (13.48) defines a number $R$ and at the same time imposes the requirement that the information term on the left has to be strictly smaller than the information term on the right. These information are a formalization of the signal strength at the eavesdropper and the legitimate receiver, respectively. Therefore, (13.48) is the formal statement of the above-noted condition that the signal strength of the jamming signal observed at the legitimate receiver has to be stronger than the signal strength observed at the eavesdropper. Note that other than through (13.48), neither the construction of the jamming scheme nor the recovery procedure at $B$ require knowledge of the eavesdropper's channel $W_E$. On the other hand, the result remains valid even if $E$ has full knowledge of $W_E$.

Under these conditions, the theorem claims the existence of a jamming strategy that has two properties: First, the legitimate receiver can fully reconstruct the jamming signal. This can be used to remove or at least mitigate its impact. For a detailed example of how this can be done in an application scenario, we refer the reader to [32, Section III]. Second, we have (13.49) which describes how the usefulness of the signal to the eavesdropper is bounded. It looks very similar to the semantic security criterion, but the values $s_1, \ldots, s_K$ appear inside the variational

distance so that it can actually not guarantee full semantic security. However, as we mentioned in the beginning of the section, it can at least guarantee that any estimate of the computed function by the eavesdropper has a higher variance than the estimate which the legitimate receiver can obtain. For details, we refer the reader to [32, Section III and Section V-A].

For the part of our result that says that the legitimate receiver can reconstruct the full jamming signal, we use an adapted result from the theory of compound channels which has been initiated in [15, 24, 72] and subsequently developed, e.g., in [4, 45, 66, 78]. It is known [45] that the classical results on compound channel capacity do not carry over to arbitrary compound channels, and the assumption made in the literature is usually that the channels have finite alphabets or exhibit a certain Gaussian structure. We use a more general condition which is also easier to treat in our framework, namely that the compound channel can be $(\delta, J)$-approximated for all $\delta > 0$ and suitable $J \in \mathbb{N}$. This notion of channel approximation follows to some extent the proof idea in [15] where it is shown that the finiteness of the alphabets implies approximability of the compound channel with a finite number of channels.

For the part of our result that says that to the eavesdropper, the jamming signal appears like i.i.d. noise, we make use of the theory of channel resolvability [40, 74] and of how it can be used to achieve information-theoretic secrecy [11, 16, 23]. The resolvability result we use is from [30].

## 13.7 Open Research Questions

We conclude this chapter with a brief discussion of directions that we think are promising for future research. These directions can be divided into three categories:

1. Improvement of the OTA computation schemes and their analysis and evaluation
2. Research on the applications of OTA computation schemes in distributed ML and consensus
3. Continuation of the research on secure OTA computation

Direction 1 is the most fundamental one in the sense that it can impact the possibilities for both of the other directions. It includes research on the improvement of the scheme itself so that it becomes more efficient or that it becomes applicable to more complex or more general system models, improvements of the theoretical analysis and more detailed numerical evaluations. The theoretical analysis could be enhanced with tighter error bounds or by expanding its applicability to more general system models, and it could be complemented with suitable converse bounds that give an idea of how tight the error guarantees are. The numerical evaluations we show in [31, 33] could profit from more detailed and, in particular, more realistic channel models that are already covered by the theoretical analysis, such as bursty interference, correlated fading and various other types of non-i.i.d and non-Gaussian components, and they could be complemented with hardware

demonstrators. Furthermore, other practically relevant complications in the system such as imperfect synchronization among the transmitters (against which we expect our schemes to have considerable robustness) could be tackled both on the theoretical and the empirical side.

Item 2 is the most significant one in terms of direct impact on technical applications. For HFL, we have noted in Sect. 13.3.1 that a huge body of research work already exists. Still, we see some possibilities for improvement with an application of our DFA schemes: For instance, we are not currently aware of OTA-HFL schemes that can deal with a lack of instantaneous channel state information both at the transmitter and the receiver and at the same time not rely on an assumption on the distribution of the sources (and such assumptions are very hard to verify in practice). In OTA-VFL, on the other hand, much more basic research questions are still open: We have suggested OTA computable ML labeling functions [13, 31] that are applicable for some cases of regression and binary classification problems and in addition, there are earlier OTA computable labeling functions available for binary classification [65], but we think that different labeling functions are necessary in order to be able to tackle larger classes of OTA-VFL problems. Additionally, there is still a need for more efficient OTA training procedures that have a better scaling behavior in the number of participating transmitters than the scheme we have proposed in [31]. For the application of distributed consensus, we believe that our DFA schemes may have the potential to yield OTA average-consensus schemes that can deal both with sub-Gaussian fading and additive channel noise and have better finite-time convergence guarantees than existing schemes.

Topic 3 has the potential to give answers to questions that have a growing relevance as the integration of OTA communication systems into real-world communication systems progresses, since we expect missing security guarantees to be a potential showstopper for many applications of OTA communication schemes. The objective in defending against eavesdropping attacks should be full semantic security, but research into active attacks that are specific to OTA computation schemes also deserves attention.

## References

1. Abad, M.S.H., Ozfatura, E., Gunduz, D., Ercetin, O.: Hierarchical federated learning across heterogeneous cellular networks. In: ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 8866–8870. IEEE, New York (2020)
2. Adler, A., Boublil, D., Elad, M., Zibulevsky, M.: A deep learning approach to block-based compressed sensing of images. arXiv preprint arXiv:1606.01519 (2016)
3. Agrawal, N., Frey, M., Stańczak, S.: A scalable max-consensus protocol for noisy ultra-dense networks. In: 2019 IEEE 20th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), pp. 1–5. IEEE, New York (2019)
4. Ahlswede, R.: Certain results in coding theory for compound channels. In: Proceedings of the Colloquium on Information Theory (1967)

5. Ahn, J.H., Simeone, O., Kang, J.: Wireless federated distillation for distributed edge learning with heterogeneous data. In: 2019 IEEE 30th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), pp. 1–6. IEEE, New York (2019)
6. Amiri, M.M., Duman, T.M., Gunduz, D.: Collaborative machine learning at the wireless edge with blind transmitters. arXiv preprint arXiv:1907.03909 (2019)
7. Amiri, M.M., Gündüz, D.: Computation scheduling for distributed machine learning with straggling workers. IEEE Trans. Signal Process. **67**(24), 6270–6284 (2019)
8. Amiri, M.M., Gündüz, D.: Federated learning over wireless fading channels. IEEE Trans. Wirel. Commun. **19**(5), 3546–3557 (2020)
9. Amiri, M.M., Gündüz, D.: Machine learning at the wireless edge: Distributed stochastic gradient descent over-the-air. IEEE Trans. Signal Process. **68**, 2155–2169 (2020)
10. Arnold, V.: On functions of three variables. Dokl. Akad. Nauk SSSR **114**, 679–681 (1957)
11. Bellare, M., Tessaro, S., Vardy, A.: Semantic security for the wiretap channel. In: Advances in Cryptology–CRYPTO 2012, pp. 294–311. Springer, Berlin (2012)
12. Benediktsson, J.A., Swain, P.H.: Consensus theoretic classification methods. IEEE Trans. Syst. Man Cybern. **22**(4), 688–704 (1992)
13. Bjelaković, I., Frey, M., Stańczak, S.: Distributed approximation of functions over fast fading channels with applications to distributed learning and the max-consensus problem. In: 2019 57th Annual Allerton Conference on Communication, Control, and Computing (Allerton), pp. 1146–1153 (2019). https://doi.org/10.1109/ALLERTON.2019.8919875
14. Blackard, K.L., Rappaport, T.S., Bostian, C.W.: Measurements and models of radio frequency impulsive noise for indoor wireless communications. IEEE J. Sel. Areas Commun. **11**(7), 991–1001 (1993)
15. Blackwell, D., Breiman, L., Thomasian, A.J.: The capacity of a class of channels. Ann. Math. Stat. **30**(4), 1229–1241 (1959)
16. Bloch, M.R., Laneman, J.N.: Strong secrecy from channel resolvability. Trans. Inf. Theory **59**(12), 8077–8098 (2013)
17. Brunet, L., Choi, H.L., How, J.: Consensus-based auction approaches for decentralized task assignment. In: AIAA Guidance, Navigation and Control Conference and Exhibit, p. 6839 (2008)
18. Buck, R.C.: Approximate complexity and functional representation. Tech. rep., Wisconsin University Madison Mathematics Research Center, Madison (1976)
19. Buck, R.C.: Nomographic functions are nowhere dense. Proc. Am. Math. Soc. **85**(2), 195–199 (1982)
20. Buldygin, V., Kozachenko, Y.: Metric Characterization of Random Variables and Random Processes. In: Cross Cultural Communication. American Mathematical Society, New York (2000). https://books.google.de/books?id=ePDXvIhdEjoC
21. Chang, W.T., Tandon, R.: Communication efficient federated learning over multiple access channels. arXiv preprint arXiv:2001.08737 (2020)
22. Christmann, A., Hable, R.: Consistency of support vector machines using additive kernels for additive models. Comput. Stat. Data Anal. **56**(4), 854–873 (2012)
23. Csiszár, I.: Almost independence and secrecy capacity. Probl. Inf. Transm. **32**(1), 40–47 (1996)
24. Dobrushin, R.L.: Optimum information transmission through a channel with unknown parameters. Radio Eng. Electron. **4**(12), 1–8 (1959)
25. Dong, J., Shi, Y., Ding, Z.: Blind over-the-air computation and data fusion via provable Wirtinger flow. IEEE Trans. Signal Process. **68**, 1136–1151 (2020)
26. Eisenberg, E., Gale, D.: Consensus of subjective probabilities: The pari-mutuel method. Ann. Math. Stat. **30**(1), 165–168 (1959)
27. El Gamal, A., Kim, Y.H.: Network information theory. Cambridge University, Cambridge (2011)
28. Eldar, Y.C., Kutyniok, G.: Compressed sensing: theory and applications. Cambridge University, Cambridge (2012)
29. French, S.: Group consensus probability distributions: a critical survey. In: Bemado, J.M., DeGroot, M.H., Lindley, D.V., Smith, A.F.M. (eds.) Bayesian Statistics II (1985)

30. Frey, M., Bjelakovic, I., Stanczak, S.: Resolvability on continuous alphabets. In: 2018 IEEE International Symposium on Information Theory (ISIT), pp. 2037–2041. IEEE, New York (2018)
31. Frey, M., Bjelakovic, I., Stanczak, S.: Over-the-air computation in correlated channels. IEEE Trans. Signal Process. **69**, 5739–5755 (2020). Preprint available at arXiv:2007.02648v2
32. Frey, M., Bjelakovic, I., Stanczak, S.: Towards secure over-the-air computation. In: IEEE Transactions on Information Forensics and Security (2020). Preprint available at arXiv:2001.03174v2
33. Frey, M., Bjelakovic, I., Stanczak, S.: Over-the-air computation in correlated channels. In: Information Theory Workshop 2020. IEEE, New York (2021). Accepted for publication
34. Gastpar, M., Vetterli, M.: Source-channel communication in sensor networks. In: Information Processing in Sensor Networks, pp. 162–177. Springer, Berlin (2003)
35. Goldenbaum, M., Stanczak, S.: Robust analog function computation via wireless multiple-access channels. IEEE Trans. Commun. **61**(9), 3863–3877 (2013)
36. Goldenbaum, M., Boche, H., Stańczak, S.: Harnessing interference for analog function computation in wireless sensor networks. IEEE Trans. Signal Process. **61**(20), 4893–4906 (2013)
37. Goldenbaum, M., Boche, H., Stańczak, S.: Nomographic functions: Efficient computation in clustered gaussian sensor networks. IEEE Trans. Wirel. Commun. **14**(4), 2093–2105 (2014)
38. Goldenbaum, M., Jung, P., Raceala-Motoc, M., Schreck, J., Stańczak, S., Zhou, C.: Harnessing channel collisions for efficient massive access in 5G networks: A step forward to practical implementation. In: 2016 9th International Symposium on Turbo Codes and Iterative Information Processing (ISTC), pp. 335–339. IEEE, New York (2016)
39. Gündüz, D., de Kerret, P., Sidiropoulos, N.D., Gesbert, D., Murthy, C.R., van der Schaar, M.: Machine learning in the air. IEEE J. Sel. Areas Commun. **37**(10), 2184–2199 (2019)
40. Han, T.S., Verdú, S.: Approximation theory of output statistics. Trans. Inf. Theory **39**(3), 752–772 (1993)
41. Hilbert, D.: Mathematische Probleme. Vortrag, gehalten auf dem internationalen Mathematiker-Congress zu Paris 1900. Gött. Nachr, pp. 253–297 (1900)
42. Hilbert, D.: Über die Gleichung neunten Grades. Math. Ann. **97**(1), 243–250 (1927)
43. Iutzeler, F., Ciblat, P., Jakubowicz, J.: Analysis of max-consensus algorithms in wireless channels. IEEE Trans. Signal Process. **60**(11), 6103–6107 (2012)
44. Kerdjidj, O., Ramzan, N., Ghanem, K., Amira, A., Chouireb, F.: Fall detection and human activity classification using wearable sensors and compressed sensing. J. Ambient. Intell. Humaniz. Comput. **11**(1), 349–361 (2020)
45. Kesten, H.: Some remarks on the capacity of compound channels in the semicontinuous case. Inf. Control **4**(2–3), 169–184 (1961)
46. Kolmogorov, A.N.: On the representation of continuous functions of several variables by superposition of continuous functions of one variable and addition. Dokl. Akad. Nauk SSSR **114**, 953–956 (1957)
47. Konečnỳ, J., McMahan, H.B., Ramage, D., Richtárik, P.: Federated optimization: Distributed machine learning for on-device intelligence. arXiv preprint arXiv:1610.02527 (2016)
48. Korner, J., Marton, K.: How to encode the modulo-two sum of binary sources (corresp.). IEEE Trans. Inf. Theory **25**(2), 219–221 (1979)
49. Liu, W., Zang, X., Li, Y., Vucetic, B.: Over-the-air computation systems: Optimization, analysis and scaling laws. IEEE Trans. Wirel. Commun. **19**(8), 5488–5502 (2020)
50. McMahan, B., Ramage, D.: Federated learning: Collaborative machine learning without centralized training data (2017). Google AI Blog. https://ai.googleblog.com/2017/04/federated-learning-collaborative.html, retrieved 02 March 2021
51. Middleton, D.: Non-gaussian noise models in signal processing for telecommunications: new methods an results for class a and class b noise models. IEEE Trans. Inf. Theory **45**(4), 1129–1149 (1999)

52. Middleton, D., Spaulding, A.D.: Elements of weak signal detection in non-gaussian noise environments. In: Poor, V., Thomas, J.B. (eds.) Advances in Statistical Signal Processing, vol. 2, pp. 137–215. JAI Press, Stamford (1993)
53. Mohri, M., Rostamizadeh, A., Talwalkar, A.: Foundations of Machine Learning. In: Adaptive Computation and Machine Learning. MIT Press, New York (2012)
54. Molinari, F., Stanczak, S., Raisch, J.: Exploiting the superposition property of wireless communication for average consensus problems in multi-agent systems. In: 2018 European Control Conference (ECC), pp. 1766–1772. IEEE, New York (2018)
55. Molinari, F., Dethof, A.M., Raisch, J.: Traffic automation in urban road networks using consensus-based auction algorithms for road intersections. In: 2019 18th European Control Conference (ECC), pp. 3008–3015. IEEE, New York (2019)
56. Molinari, F., Agrawal, N., Stanczak, S., Raisch, J.: Max-consensus over fading wireless channels. IEEE Transactions on Control of Network Systems **8**(2), 791–802 (2021)
57. Nazer, B., Gastpar, M.: Computation over multiple-access channels. IEEE Trans. Inf. Theory **53**(10), 3498–3516 (2007)
58. Nazer, B., Gastpar, M.: Compute-and-forward: Harnessing interference through structured codes. IEEE Trans. Inf. Theory **57**(10), 6463–6486 (2011)
59. Nazer, B., Cadambe, V.R., Ntranos, V., Caire, G.: Expanding the compute-and-forward framework: Unequal powers, signal levels, and multiple linear combinations. IEEE Trans. Inf. Theory **62**(9), 4879–4909 (2016)
60. Olfati-Saber, R.: Flocking for multi-agent dynamic systems: Algorithms and theory. IEEE Trans. Autom. Control **51**(3), 401–420 (2006)
61. Olfati-Saber, R., Fax, J.A., Murray, R.M.: Consensus and cooperation in networked multi-agent systems. Proc. IEEE **95**(1), 215–233 (2007)
62. Ordentlich, O., Zhan, J., Erez, U., Gastpar, M., Nazer, B.: Practical code design for compute-and-forward. In: 2011 IEEE International Symposium on Information Theory Proceedings, pp. 1876–1880. IEEE, New York (2011)
63. Ozfatura, E., Ulukus, S., Gündüz, D.: Distributed gradient descent with coded partial gradient computations. In: ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 3492–3496. IEEE, New York (2019)
64. Palangi, H., Ward, R., Deng, L.: Distributed compressive sensing: A deep learning approach. IEEE Trans. Signal Process. **64**(17), 4504–4518 (2016)
65. Ralinovski, K., Goldenbaum, M., Stańczak, S.: Energy-efficient classification for anomaly detection: The wireless channel as a helper. In: 2016 IEEE International Conference on Communications (ICC), pp. 1–6 (2016)
66. Root, W.L., Varaiya, P.P.: Capacity of classes of gaussian channels. SIAM J. Appl. Math. **16**(6), 1350–1393 (1968)
67. Seif, M., Tandon, R., Li, M.: Wireless federated learning with local differential privacy. arXiv preprint arXiv:2002.05151 (2020)
68. Sery, T., Cohen, K.: On analog gradient descent learning over multiple access fading channels. IEEE Trans. Signal Process. **68**, 2897–2911 (2020)
69. Steinwart, I., Christmann, A.: Support Vector Machines. In: Information Science and Statistics. Springer, Berlin (2008)
70. Sun, Y., Zhou, S., Gündüz, D.: Energy-aware analog aggregation for federated learning with redundant data. arXiv preprint arXiv:1911.00188 (2019)
71. Winkler, R.L.: The consensus of subjective probability distributions. Manag. Sci. **15**(2), B–61 (1968)
72. Wolfowitz, J.: Simultaneous channels. Arch. Ration. Mech. Anal. **4**(1), 371–386 (1959)
73. Wu, Y., Rosca, M., Lillicrap, T.: Deep compressed sensing. In: International Conference on Machine Learning, pp. 6850–6860. PMLR (2019)
74. Wyner, A.: The common information of two dependent random variables. Trans. Inf. Theory **21**(2), 163–179 (1975)
75. Yang, Q., Liu, Y., Chen, T., Tong, Y.: Federated machine learning: Concept and applications. ACM Trans. Intell. Syst. Technol. (TIST) **10**(2), 1–19 (2019)

76. Yang, K., Jiang, T., Shi, Y., Ding, Z.: Federated learning via over-the-air computation. IEEE Trans. Wirel. Commun. **19**(3), 2022–2035 (2020)
77. Yoo, J., Turnes, C., Nakamura, E.B., Le, C.K., Becker, S., Sovero, E.A., Wakin, M.B., Grant, M.C., Romberg, J., Emami-Neyestanak, A., et al.: A compressed sensing parameter extraction platform for radar pulse signal acquisition. IEEE J. Emerging Sel. Top. Circuits Syst. **2**(3), 626–638 (2012)
78. Yoshihara, K.: Coding theorems for the compound semi-continuous memoryless channels. In: Kodai Mathematical Seminar Reports, vol. 17, pp. 30–43. Department of Mathematics, Tokyo Institute of Technology, Tokyo (1965)
79. Zeng, Q., Du, Y., Leung, K.K., Huang, K.: Energy-efficient radio resource allocation for federated edge learning. arXiv preprint arXiv:1907.06040 (2019)
80. Zhan, J., Nazer, B., Gastpar, M., Erez, U.: MIMO compute-and-forward. In: 2009 IEEE International Symposium on Information Theory, pp. 2848–2852. IEEE, New York (2009)
81. Zhao, Y., Li, M., Lai, L., Suda, N., Civin, D., Chandra, V.: Federated learning with non-iid data. arXiv preprint arXiv:1806.00582 (2018)
82. Zhou, Z., Chen, X., Li, E., Zeng, L., Luo, K., Zhang, J.: Edge intelligence: Paving the last mile of artificial intelligence with edge computing. Proc. IEEE **107**(8), 1738–1762 (2019)
83. Zhu, G., Wang, Y., Huang, K.: Broadband analog aggregation for low-latency federated edge learning. IEEE Trans. Wirel. Commun. **19**(1), 491–506 (2019)
84. Zhu, G., Liu, D., Du, Y., You, C., Zhang, J., Huang, K.: Toward an intelligent edge: Wireless communication meets machine learning. IEEE Commun. Mag. **58**(1), 19–25 (2020)

# Chapter 14
# Information Theory and Recovery Algorithms for Data Fusion in Earth Observation


**Massimo Fornasier, Danfeng Hong, Gerhard Kramer, Lars Palzer, Michael Rauchensteiner, and Xiao Xiang Zhu**

## 14.1 General Framework

In this chapter we explore instances of the general nonlinear data model or data processing model

$$y = g(A(x)(x + n_s)) + n_m, \tag{14.1}$$

where $x$ is a source input, $y$ is a data or measurement output, $A(\cdot)$ is a linear map, possibly input dependent, and $n_s$, $n_m$ are noise terms at the source or measurements, respectively. The nonlinear function $g$ may be a source of measurement distortion, but it could also be a man-made design such as a multi-channel multi-bit quantization function or an activation function as in artificial neural networks. Model (14.1) represents a general framework for quantized compressed sensing, quasi-linear, bilinear and more general nonlinear compressed sensing, single and multi-index models, and artificial neural networks.

M. Fornasier (✉) · M. Rauchensteiner
Technical University of Munich, Garching, Germany
e-mail: massimo.fornasier@ma.tum.de; michael.rauchensteiner@ma.tum.de

D. Hong
Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing, China
e-mail: hongdf@aircas.ac.cn

G. Kramer · L. Palzer
Technical University of Munich, Munich, Germany
e-mail: gerhard.kramer@tum.de; lars.palzer@tum.de

X. X. Zhu
Technical University of Munich, Munich, Germany
e-mail: xiaoxiang.zhu@tum.de



G. Kutyniok et al. (eds.), *Compressed Sensing in Information Processing*, Applied and Numerical Harmonic Analysis, https://doi.org/10.1007/978-3-031-09745-4_14

Two fundamental tasks are associated with model (14.1).

- **Inversion** Given $g$, $A(\cdot)$, and knowledge of the stochastic behavior of the noise sources, we wish to establish the theoretical and practical invertibility of the model. This means to determine theoretically under which conditions, given the measurement $y$, it is possible to recover $x$ up to a given precision and possibly provide constructive/algorithmic solutions. The theoretical analysis aims at establishing the minimal conditions for invertibility and it may be based, e.g., on complexity analysis or asymptotic theories, such as the rate-distortion theory [8, 51, 66], both seeking optimal information-theoretic bounds. The practical inversion of the model (14.1) is often performed by suitable numerical optimizations with objective functions that incorporate misfit to data $y$ and priors on the source $x$, see, e.g., [18, 19].
- **Learning/identification** Some parameters of the model are unknown, e.g., the linear mixing map $A(\cdot)$ may be unknown as in dictionary learning problems, see, e.g., [4]. In this case, given a collection, called training set, of input-output pairs $\{(x_i, y_i) : i = 1, \ldots, N\}$ one seeks to identify the parameters that allow the model to reproduce the input-output mapping on the training set. This is the typical situation encountered, for instance, in the training or identification of neural networks.

In some applications, one may be interested even to perform both tasks, i.e., identification/learning and invertibility, simultaneously from one instance input-output pair $(x, y)$ only. This is, for instance, the case in bilinear compressed sensing and its applications in wireless communication [41], where both channel and signal needs to be recovered.

Learning/identification and inversion of model (14.1) are necessary tasks for many applications in data and signal processing. Besides the ones already mentioned above, we focus on and report here about data fusion in remote sensing [63, 70, 74]. In particular, the massive amount of available complementary multi-sensor satellite imagery offers an ideal basis for learning inter-sensor representations with, e.g., deep neural networks [48], which we found to be a powerful approach in various remote sensing areas [34, 54].

The rest of the chapter is organized as follows: In Sect. 14.2 we present theoretical results of *learning/identification* of model (14.1) in the context of feed-forward deep artificial neural networks. Section 14.3 presents results about quantized Compressed Sensing with message passing reconstruction, which is an *inversion* problem via a form of neural network. The final Sect. 14.4 shows novel models of the type (14.1) in signal processing for *real-life applications* in remote sensing and Earth observation.

## 14.2 Identification of Neural Networks: From One Neuron to Deep Neural Networks

In this section, we present results of learning/identification of generic feed-forward artificial neural networks, which can be interpreted in the general framework of our main model (14.1). Here we follow a compressed sensing approach in the sense that we establish exact recovery of all parameters of a neural network from the observation of the minimal amount of input-output pairs. Moreover, low-rank approximations to the space generated by tensor products of (entangled) weights play a crucial role.

### *14.2.1 From One Neuron to Deep Networks*

Deep learning has become an extremely successful approach with state-of-the-art performances on various applications such as speech recognition, image recognition, language translation, and as a novel method for scientific computing. In order to understand how deep learning works, it is perhaps beneficial to start with the simplest building block, i.e., the artificial neuron. An artificial neuron is a ridge function $f : \mathbb{R}^d \to \mathbb{R}$ of the type

$$f(x) = g(w^T x + \theta) = \rho(w^T x) = \rho(w \cdot x), \tag{14.2}$$

where $g : \mathbb{R} \to \mathbb{R}$ is a scalar univariate function and $w \in \mathbb{R}^d$ is the direction of the ridge function. Ridge functions are simple functions to be used as a phase-field (hyperplane) separatrix. Sums of such functions provide (soft) tilings/tessellation of the space. In general the set of linear combinations of ridge functions $\{f(x) = \sum_{i=1}^m \alpha_i g(w_i \cdot x + \theta_i)\}$ is dense in the continuous functions $\mathcal{C}$ for $g \in \mathcal{C}_b$. Hence, they constitute simple building blocks for composing high-dimensional functions.

Deep learning is about realizing complex tasks by means of highly parametrized functions $f : \mathbb{R}^d \to \mathbb{R}^{m_L}$, called deep artificial neural networks, composing layers of artificial neurons of the type

$$f(x) = f(x; (W, \tau)) := \rho_L(W_L^\top \rho_{L-1}(W_{L-1}^\top \dots \rho_1(W_1^\top x) \dots),$$

where $\rho_\ell(\cdot) = g(\cdot + \tau_\ell)$ are shifted activation functions, the shifts $\tau_\ell \in \mathbb{R}^{m_\ell}$ are bias parameters, and the matrices $W_\ell \in \mathbb{R}^{m_\ell \times m_{\ell+1}}$ collect the weight of each layer $\ell = 1, \dots, L$.

In practical applications the number of layers $L$, determining the depth of the network, and the dimensions $m_\ell \times m_{\ell+1}$ of the weight matrices $W_\ell$ are typically determined through heuristic considerations, whereas the weight matrices themselves and the shifts are learned based on training data. The training of a neural network is done by data misfit: Given training data $(x_i, y_i)$, $i = 1, \dots, N$ (for

instance, an image $x_i$ and a label $y_i$ indicating, e.g., whether the image is of cat or dog), one learns $(W, \tau)$ by minimization of the loss function

$$\mathcal{L}(W, \tau) = \frac{1}{N} \sum_{i=1}^{N} |y_i - f(x_i; (W, \tau))|^2 .$$

Mean squared error loss as above is one of the most popular misfit used in practice, but also the Kullback–Leibler divergence (relative entropy), or Wasserstein distances are discrepancies considered much in practice. The minimization of the misfit is usually performed by first order optimization algorithms, such as (stochastic) gradient descent and variations [57], because of their simplicity and scalability. In support of deep learning come the empirical evidence of being able of outperforming other methods (against "certified" benchmarks, e.g., ImageNet), but also the recent theoretical discoveries, e.g., [9, 10, 15] that show that deep artificial network can approximate very complicated high-dimensional functions without incurring in the curse of dimensionality.

### *14.2.2 Data Interpolation and Identification of Neural Networks*

For the number of samples $N = \mathcal{O}(\overline{W})$, where $\overline{W} = \sum_{\ell=1}^{L-1} m_\ell \times m_{\ell+1}$ is the complexity of the network, it is known that there exist data interpolating networks with 0-loss, see, e.g. [61, 72]. We call this situation the *realizable regime*, i.e., the data are realized by a network. In this case one expects multiple optimal networks, which may be due also to symmetries (permutation of neurons in one layer, and signs due to symmetries of activation functions, e.g., $\tanh(2x - 1) = -\tanh(-2x + 1)$). In the overparametrized regime $N \ll \overline{W}$ the loss landscape becomes increasingly "less nonconvex" and the number of possible optimal network may further increase. Surprisingly overparametrization and fitting do not cause overfitting [73]. This due to the fact that optimization methods used for training such as (stochastic) gradient descent perform an implicit regularization and they promote low complexity networks, see, e.g., [3].

The unique identifiability of neural networks from realizable input-output pairs has been considered in the literature for over three decades [2, 58]. Perhaps the most relevant result is the one of Fefferman [16]:

**Theorem 14.1 (Fefferman '94)** *A generic fully connected deep neural network with activation function* $g = \tanh$ *is uniquely identified by its output up to natural symmetries.*

We provide here the general lines of the proof of this remarkable result: The proof uses that tanh is a meromorphic function that is $i$-periodic. Then sums and compositions of dilated and shifted tanh are again regular outside countable poles,


**Fig. 14.1** Failure of stochastic gradient descent to identify a neural network in the realizable regime

which encode the weights and shifts of the network. If two networks coincide on $\mathbb{R}^d$ then by analytic extension they coincide on $\mathbb{C}^d$ and necessarily have the same poles, hence the same network architecture. The result has been generalized by Vlacic and Boelcskei [62] to non-fully connected networks with more general activation functions. However, a neural network remains fully determined by a *finite* number of parameters. Its identification is known since the 1990s to be in general an NP-hard problem [6, 38]. However, identification is not at all expected to generically require an infinite amount of training samples as assumed in the above mentioned results. Yet gradient descent will not converge to identification in the realizable regime $N = \mathcal{O}(\overline{W})$, see Fig. 14.1. This failure of such algorithm to perform identification is due to the non-convexity of the loss.

Identifiability means that given a sufficiently generic network no other network, smaller or larger, up to the above mentioned equivalences, can in fact realize the same task exactly (explainability and uniform stability). Robustness or stability mean that if a larger network performs approximately a task, then it may be reduced to a minimal and potentially significantly smaller network performing approximately the same task.

The scope of results [17, 20–22] can be summarized in the following

**Theorem 14.2 (Fornasier et al. 2018–2021)** *Generic networks of complexity $\overline{W}$ and with smooth activation functions can be stably and constructively identified up to natural symmetries from a number $N = \mathcal{O}(\overline{W})$ of (possibly actively chosen) samples.*

Below we would like to show instances of this result, starting from the most basic case of scalar shallow networks to arrive to the case of generic deep neural networks. The typical proof architecture goes as follows. We learn the parameters, comprised of weight matrices and shifts, in a two-step fashion.

- First, we use derivative information of the network to identify the so-called entangled weight matrices, which essentially encode the information contained in the standard weight matrices up to sign and scaling.
- Signs, scalings, and shifts of the network are identified by means of gradient descent for least squares.

Let us start showing how such results are proven with the simple case of shallow networks.

### 14.2.3 Shallow Feed-Forward Neural Networks

A neural network $f : \mathbb{R}^d \to \mathbb{R}$ with one hidden layer and one output node can be defined as

$$f(x) = \sum_{i=1}^{m} b_i g(w_i^T x + \theta_i) = \sum_{i=1}^{m} \rho_i(w_i^T x).$$

We start with the case $m \leq d$ and $\{w_1, \dots w_m\}$ linearly independent (below we also approach the overcomplete case $m > d$, see Remark 14.1). Differently from the case of the single neuron, the use of first order differentiation

$$\nabla f(x) = \sum_{i=1}^{m} \rho_i'(w_i^T x) w_i \in W = \operatorname{span}\{w_1, \dots, w_m\},$$

furnishes information about $W = \operatorname{span}\{w_1, \dots, w_m\}$ (active subspace identification, in a moment), but it does not on the single weights $w_i$. Higher order differentiation yields tensors

$$D^k f(x) = \sum_{i=1}^{m} \rho_i^{(k)}(x) \underbrace{w_i \otimes \cdots \otimes w_i}_{k-\text{times}},$$

which require that the $\rho_i$'s are sufficiently smooth. In a setting where the samples are actively chosen, it is generally possible to approximate these derivatives by finite differences. However, even for *passive sampling* there are ways to construct similar tensors, which rely on Stein's lemma or differentiation by parts or weak differentiation. If the density $p(x)$ of the sampling points $x_i$'s is (approx.) known, i.e., $d\mu_X(x) = p(x)dx$, then

$$\begin{aligned}
\frac{1}{N} \sum_{i=1}^{N} f(x_i)(-1)^k \frac{\nabla^k p(x_i)}{p(x_i)} &\approx \int_{\mathbb{R}^d} f(x)(-1)^k \frac{\nabla^k p(x)}{p(x)} p(x)dx \\
&= \int_{\mathbb{R}^d} \nabla^k f(x) d\mu_X(x) = \mathbb{E}_{x\sim\mu_X}[\nabla^k f(x)] \\
&= \sum_{i=1}^{m} \left( \int_{\mathbb{R}^d} \rho_i^{(k)}(w_i^T x) d\mu_X(x) \right) \underbrace{w_i \otimes \cdots \otimes w_i}_{k-\text{times}}.
\end{aligned}$$

In recent work [36], stable 1-rank decompositions of third order symmetric tensors ($k = 3$) have been used for the weights identification of one hidden layer neural networks. In this review chapter we show that using second derivatives ($k = 2$) surprisingly suffices and the corresponding error estimates reflect positively the lower order and potential of improved stability. Moreover, the technique extends to deeper networks and overcomplete cases.

**Theorem 14.3 (Fornasier, Vybíral, Daubechies)** *Let $m \leq d$ and let $f$ be a shallow network $f(x) = \sum_{i=1}^{m} \rho_i(w_i \cdot x)$, with $\rho_i \in C^3[-1,1]$, $\{w_i\} \subset \mathbb{R}^d$ are lin. independ. Let $\epsilon > 0$. Then by using at most $m\chi[(d+1)+(m+1)(m+2)/2]$ random exact point evaluations of $f$, which correspond to numerical differentiation of $f$ with stepsize $\epsilon$, there exists a constructive algorithm computing approximations $\{\hat{w}_i\}$ of the weights up to a sign, for which*

$$\left(\sum_{i=1}^{m} \|\hat{w}_i - w_i\|_2^2\right)^{1/2} \lesssim \epsilon,$$

*with probability at least $1 - m\delta(m\chi/m^2)$. Moreover, we can construct an approximating shallow net $\hat{f} : B_1^d \to \mathbb{R}$ with weights $\hat{w}_i$, such that*

$$\|f - \hat{f}\|_{L_\infty(B_1^d)} \lesssim \epsilon.$$

***Proof (Sketch)*** The proof of this result is based on the following arguments. By active subspace search based on first order differentiation it is possible to reduce the problem to the case $m = d$. We omit here more details and we refer to [22]. Once this dimensionality reduction is performed, the recovery strategy of the weights $w_i$ goes along the following steps:

- Recover an approximation to $\widehat{\mathcal{W}}$ of $\mathcal{W} = \text{span}\{w_i \otimes w_i, i = 1, \dots, m\} \subset \mathbb{R}^{m \times m}$ (by active or passive sampling) by using approximate *second order* differentiation;
- Perform a *whitening procedure*, which allows us to restrict our search to near orthonormal weights $w_1, \dots, w_m$ without loss of generality;
- Then we consider the following algorithm

$$\arg\max \|M\|_\infty, \quad \text{s.t.} \quad M \in \widehat{\mathcal{W}}, \|M\|_F \leq 1$$

to recover $w_i$'s—or their good approximation $\hat{w}_i$ (which is of course possible only up to the sign).

□

Let us describe concisely these steps in more detail.

#### 14.2.3.1 The Approximation to $\mathcal{W}$: Active Sampling

By PCA compute

$$\widehat{\mathcal{W}} = \mathrm{PCA}\{\Delta_\epsilon^2 f(x_j),\, j = 1, \ldots, m_{\mathcal{X}}\},$$

where

$$(\Delta_\epsilon^2 f(x))_{j,k} = \frac{f(x + \epsilon(e_j + e_k)) - f(x + \epsilon e_j) - f(x + \epsilon e_k) + f(x)}{\epsilon^2},$$

for $j, k = 1, \ldots, m$, is a finite difference approximation to the Hessian of $f$ a $x$. (Passive sampling is also possible.) For $x$ drawn at random and by applying in a suitable way Chernoff matrix bounds [60], one derives a probabilistic error estimate, in the sense that

$$\|P_{\mathcal{W}} - P_{\widehat{\mathcal{W}}}\|_{F \to F} \leq C m^{3/2} \epsilon,$$

with high probability, see Fig. 14.2.

#### 14.2.3.2 Whitening

Once $\widehat{\mathcal{W}} \approx \mathcal{W}$ is available, the whitening procedure is used to reduce the problem to the case where the weights are nearly orthonormal. We denote

$$\mathcal{S}(w_1, \ldots, w_m) = \inf\left\{\left(\sum_{i=1}^{m} \|w_i - z_i\|_2^2\right)^{1/2} : z_1, \ldots, z_m \quad \text{orthonormal basis in } \mathbb{R}^m\right\},$$

as measure of the level of orthogonality of the $w_i$'s.

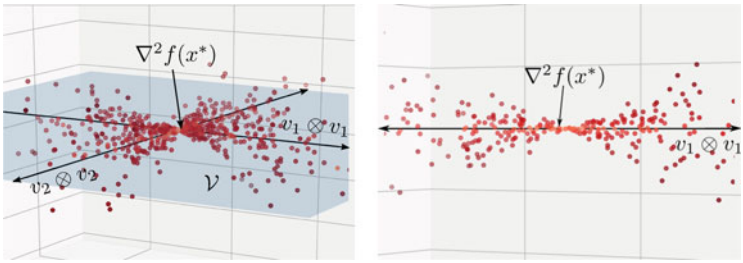



**Fig. 14.2** Concentration of Hessians

**Theorem 14.4 (Fornasier, Vybíral, Daubechies)** *Let $\gamma, \eta > 0$ be positive real numbers. Let $\|P_{\mathcal{W}} - P_{\widehat{\mathcal{W}}}\|_{F\to F} \leq \eta$ and let $\widehat{G} \in \widehat{\mathcal{W}}$ and $G = P_{\mathcal{W}}(\widehat{G})$ be positive definite with $\widetilde{G} \succcurlyeq \gamma I_m$, then*

$$\mathcal{S}(\sqrt{\lambda_1}\widehat{A}w_1, \ldots, \sqrt{\lambda_m}\widehat{A}w_m) \leq \frac{\eta\|\widehat{G}\|_F}{\gamma}$$

*and $\left\{\frac{\widehat{A}w_1}{\|\widehat{A}w_1\|_2}, \ldots, \frac{\widehat{A}w_m}{\|\widehat{A}w_m\|}\right\}$ are $\varepsilon$-nearly orthonormal, for $\varepsilon = \frac{\sqrt{2}\eta\|\widehat{G}\|_F}{\gamma}$, i.e.,*

$$\mathcal{S}\Big(\frac{\widehat{A}w_1}{\|\widehat{A}w_1\|_2}, \ldots, \frac{\widehat{A}w_m}{\|\widehat{A}w_m\|_2}\Big) \leq \frac{\sqrt{2}\eta\|\widehat{G}\|_F}{\gamma} =: \varepsilon.$$

*An optimal choice of $\widehat{G}$ can be obtained by solving*

$$\max_{\substack{\widetilde{G}\in\widehat{\mathcal{W}}\\ \|\widetilde{G}\|_F=1}} \min_{\substack{x\in\mathbb{R}^m\\ \|x\|_2=1}} x^T\widetilde{G}x.$$

*(Notice that the map $\widetilde{G} \to \min_{\substack{x\in\mathbb{R}^m\\ \|x\|_2=1}} x^T\widetilde{G}x$ is concave and one can use a projected gradient ascent method to compute $\widehat{G}$.)*

In view of the simple reformulation

$$f(W^Tx) = \sum_{i=1}^m \rho_i(w_i \cdot \widehat{A}^Tx) = \sum_{i=1}^m \widehat{\rho}_i(\frac{\widehat{A}w_i}{\|\widehat{A}w_i\|_2}w_i \cdot x) = \widehat{f}(x),$$

and Theorem 14.4, for $\widehat{\rho}_i(t) = \rho_i(t\|\widehat{A}w_i\|_2)$, we could further assume without loss of generality that the vectors $\{w_i : i = 1, \ldots, m\}$ are nearly orthonormal in first place.

#### 14.2.3.3 The Recovery Strategy of the Weights $w_i$

It should be clear that $\widehat{W}_i = P_{\widehat{\mathcal{W}}}(w_i \otimes w_i)$ are nearly optimal solutions for

$$\arg\max \|M\|_\infty, \quad \text{s.t.} \quad M \in \widehat{\mathcal{W}}, \|M\|_F \leq 1. \tag{14.3}$$

Under the assumption of near orthonormality of $w_i$ obtained by whitening, by studying first and second order optimality conditions for this constrained nonconvex program, one can also show that there are no relevant solutions other than $\widehat{W}_i$. A simple projected gradient ascent algorithm performs the search of optimal solutions. This concludes the details of the proof of Theorem 14.3.

### 14.2.4 Deeper Networks

The approach has been further extended to two hidden layers and deeper networks in [17, 21]. Recall that a feed-forward deep network is given by

$$f(x) = f(x; (W, \tau)) := \rho_L(W_L^\top \rho_{L-1}(W_{L-1}^\top \dots \rho_1(W_1^\top x)\dots)$$

or more formally

**Definition 14.1 (Feed-Forward Neural Network)** Let $L, m_0, \dots, m_L \in \mathbb{N}$ with $D = m_0$. For $\ell \in [L]$, consider weight matrices

$$W_\ell = \left(w_1^{[\ell]} | \dots | w_{m_\ell}^{[\ell]}\right) \in \mathbb{R}^{m_{\ell-1} \times m_\ell},$$

shifts $\tau_\ell \in \mathbb{R}^{m_\ell}$, and let $g : \mathbb{R} \to \mathbb{R}$ be an activation function. A feed-forward neural network with $m_L$ outputs is a function $f : \mathbb{R}^d \to \mathbb{R}^{m_L}$ computed via the recursive rule $y^{[0]}(x) = x$,

$$y^{[\ell]}(x) = g(W_\ell^\top y^{[\ell-1]} + \tau_\ell), \quad \ell \in [L]$$

and $f(x) = y^{[L]}(x)$, where $g$ is meant to be applied component-wise to non-scalar inputs. The components of $f$ are denoted by $f_p$ for $p \in [m_L]$ and we often write $\rho_\ell(\cdot) = g(\cdot + \tau_\ell)$. It will often be useful to refer to the number of neurons of the network as $m = m_1 + \dots + m_L$.

Let us assume now that an oracle provided us with matrices of the type

$$\widetilde{V}_\ell = \prod_{k=1}^{\ell-1} (W_k D_k) W_\ell \pi_\ell S_\ell \in \mathbb{R}^{D \times m_\ell}, \qquad \ell \in [L], \tag{14.4}$$

where $D_1, \dots, D_{L-1}, S_1, \dots, S_L$ are arbitrary invertible diagonal matrices and $\pi_1, \dots, \pi_L$ are permutation matrices. Then the network can be completely reparametrized by using the matrices $\widetilde{V}_\ell$ with fewer remaining free parameters $S_\ell, D_\ell, \tau_\ell$:

**Proposition 14.1 (Fiedler, Fornasier, Klock, Rauchensteiner)** *Assume* $\mathrm{rk}(\widetilde{V}_\ell) = m_\ell$ *for all* $\ell \in [L]$. *Then the feed-forward network* $\tilde{f}$ *defined by weight matrices* $\widetilde{W}_1^\top = S_1^{-1} \widetilde{V}_1^\top$ *and*

$$\widetilde{W}_{\ell+1}^\top = S_{\ell+1}^{-1} \widetilde{V}_{\ell+1}^\top (\widetilde{V}_\ell^\top)^\dagger S_\ell \widetilde{D}_\ell^{-1}, \quad \ell \in [L-1],$$

*with* $\widetilde{D}_\ell = \pi_\ell^\top D_\ell \pi_\ell$, *shifts* $\widetilde{\tau}_\ell = \pi_\ell^\top \tau_\ell$, *and activation functions* $g$ *satisfies* $\widetilde{f} \equiv \pi_L^\top \circ f$.

This parametrization allows to reduce the problem of identification from $\overline{W}$ parameters to $\mathcal{O}(\sqrt{\overline{W}})$. This reduction of complexity turns out key to allow gradient descent to be able to complete the identification (contrary to Fig. 14.1). Surprisingly matrices of the type (14.4) can be easily obtained again by *second order* differentiation: Denote diagonal matrices $G_\ell(x) := \operatorname{diag}\left(\rho'_\ell(W_\ell^\top y^{[\ell-1]}(x))\right) \in \mathbb{R}^{m_\ell \times m_\ell}$ and the so-called *entangled weights* are given by

$$V_\ell(x) := \left(\prod_{k=1}^{\ell-1} W_k G_k(x)\right) W_\ell. \tag{14.5}$$

**Proposition 14.2** *The Hessian of the $p$-output $f_p$ of a FFNN $f$ reads*

$$\begin{aligned}\nabla^2 f_p(x) &= \sum_{i=1}^{m_1} S^{[1]}_{p,i}(x)\left(w_i^{[1]} \otimes w_i^{[1]}\right) + \sum_{\ell=2}^{L}\sum_{i=1}^{m_\ell} S^{[\ell]}_{p,i}(x)\left(v_i^{[\ell]}(x) \otimes v_i^{[\ell]}(x)\right)\\ &= \sum_{\ell=1}^{L} V_\ell(x) S_p^{[\ell]}(x) V_\ell(x)^\top,\end{aligned}$$

*for $p \in [m_L]$. ($v_i^{[\ell]}(x)$ are columns of $V_\ell(x)$.)*

Hence, if the decomposition $\nabla^2 f_p(x) = \sum_{\ell=1}^{L} V_\ell(x) S_p^{[\ell]}(x) V_\ell(x)^\top$ would give access to $V_\ell(x)$ one could then apply Proposition 14.1. Unfortunately, such matrix decomposition is nonorthogonal and not immediately obtainable. However, combining concentration of measure and suitable optimizations it is possible to obtain $V_\ell(x)$ for some $x^*$ from sampling multiple Hessians. Let us show how this strategy works. First one needs to compute a suitable space $\widehat{\mathcal{V}} \approx \mathcal{V}$ spanned by symmetric rank-1 matrices:

- In view of the Lipschitz continuity of $x \to V_\ell(x)$, by sampling approximate Hessians $\Delta^2_\varepsilon f_p(x)$ from a distribution $x \sim \mu_X$ tightly concentrating, e.g., at $x^*$, Hessians cluster around a subspace $\mathcal{V} = \operatorname{span}\{v_i \otimes v_i\}$.
- The spanning rank-1 matrices $v_i \otimes v_i = v_i^{[\ell]}(x^*) \otimes v_i^{[\ell]}(x^*)$ are precisely made of entangled weight vectors $v_i^{[\ell]}(x^*)$, columns of $V_\ell(x^*)$;
- The subspace $\mathcal{V}$ can be again stably approximated $\widehat{\mathcal{V}} \approx \mathcal{V}$ by PCA of the point cloud $\{\Delta^2_\varepsilon f_p(x_i) : x_i \sim \mu_X\}$, see Fig. 14.2.

Then one uses optimization to find $v_i^{[\ell]}(x^*) \otimes v_i^{[\ell]}(x^*)$ near the subspace $\widehat{\mathcal{W}}$:

- Denote $\mathcal{V} = \{v_1 \otimes v_1, \ldots, v_m \otimes v_m\}$, where $m = m_1 + \cdots + m_L > d$.
- We denote the approximation error by $\|P_{\widehat{\mathcal{V}}} - P_{\mathcal{V}}\|_F =: \delta$, and an upper error bound $\nu := C_F - 1$, where $C_F > 1$ is the upper frame constant of $\{v_1, \ldots, v_m\}$ (hence the system is not expected to be near orthonormal but it is morally a near unit norm Parseval frame!).

- Instead of optimizing over matrices as done in (14.3), one can opt for more efficiently optimizing over vectors:

$$\max_{\|u\|_2=1} \Phi_{\widehat{\mathcal{V}}}(u) := \|P_{\widehat{\mathcal{V}}}(u \otimes u)\|_F^2.$$

**Theorem 14.5 (Fiedler, Fornasier, Klock, Rauchensteiner)** *Assume that $\nu$ and $\delta$ are sufficiently small. For each $i$ there exists a local maximizer $u_i^*$ of $\Phi_{\widehat{\mathcal{W}}}$ with $\Phi_{\widehat{\mathcal{W}}}(u_i^*) \geq 1 - \delta$ within the cap*

$$U_i := \left\{ u \in \mathbb{S}^{D-1} : \langle u, w_i \rangle \geq \sqrt{(1-3\delta)\frac{1-\nu}{1+\nu}} \right\}.$$

*Furthermore, for any constrained local maximizer $u \in \mathbb{S}^{d-1}$ of $\Phi_{\widehat{\mathcal{V}}}$ with $\Phi_{\widehat{\mathcal{V}}}(u) > 7\frac{1+\nu}{1-\nu}\delta$ and basis expansion $P_{\widehat{\mathcal{V}}}(u \otimes u) = \sum_{i=1}^K \sigma_i P_{\widehat{\mathcal{V}}}(v_i \otimes v_i)$ ordered according to $\sigma_1 \geq \ldots \geq \sigma_K$, we have*

$$\min_{s\in\{-1,1\}} \|u - s v_j\|_2 \leq \frac{\sqrt{2\nu \sum_{i=2}^K \sigma_i^2} + 2\delta}{(1-\nu)(1 - 6\frac{\nu}{1+\nu} - 13\delta) - \sqrt{6\frac{\nu}{1+\nu} + 13\delta} - 2\delta}.$$

*Remark 14.1* Notice that this procedure of identification of the $v_i \otimes v_i$ in $\mathcal{V}$ for $m > d$ now also solves the problem of identification of 1-hidden layer neural networks for the number of neurons $m > d$.

Once entangled weights are computed by Algorithm 1 as ensured by Theorem 14.5 and [42], see Fig. 14.3, one can proceed to identifying the residual parameters by gradient descent, see Fig. 14.4. This conclude the proof of Theorem 14.2. We compare stochastic gradient descent (SDG) with our algorithmic pipeline in Fig. 14.4. While SGD does not provide identification of the network, our method obtains full recovery of deep networks very efficiently.

**Algorithm 1:** Subspace power method

**input** : $P_{\widehat{\mathcal{V}}}$, stepsize $\gamma > 0$, number of iterations $J$
Sample $u_0 \sim \mathrm{Uni}(\mathbb{S}^{D-1})$
**for** $j = 1,\ldots,J$ **do**
  $u_j = P_{\mathbb{S}^{D-1}}(u_{j-1} + 2\gamma P_{\widehat{\mathcal{V}}}(u_{j-1} \otimes u_{j-1})u_{j-1})$
**end**
**output** : $u_J$

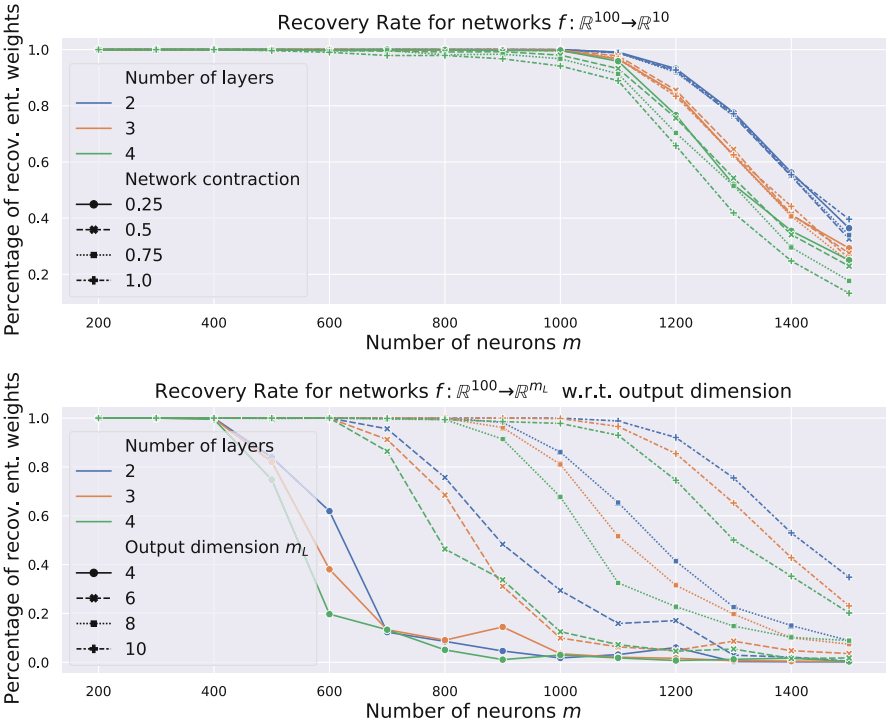



**Fig. 14.3** Percentage of entangled weights' recovery for different architectures (with fixed and variable number of outputs) with respect to the number of neurons

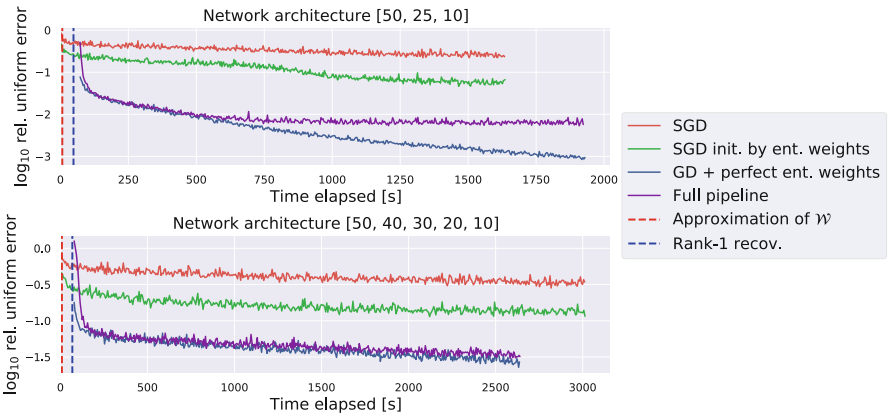



**Fig. 14.4** Comparison of stochastic gradient descent

## 14.3 Quantized Compressed Sensing with Message Passing Reconstruction

In this section, we study quantized compressed sensing (QCS) from a statistical inference point of view. Consider the model

$$Q_k = g\Big(\frac{1}{\sqrt{n}}\langle A_k, \mathsf{X}\rangle\Big), \qquad 1 \le k \le m, \tag{14.6}$$

where

- $\mathsf{X}$ is a vector in $\mathbb{R}^n$ whose entries $x_i$, $i = 1, \ldots, n$ are output by a memoryless source with distribution $P_X$,
- $A_k$ is the transposed $k$th row of $\mathbf{A} \in \mathbb{R}^{m\times n}$ which is a dense measurement matrix with iid $\mathcal{N}(0, 1)$ entries, and
- $g : \mathbb{R} \to \mathcal{Q}$ with $\#(\mathcal{Q}) = 2^b$ is a $b$-bit quantization function.

We assume that $P_X$, $g$, and $\mathbf{A}$ are known to the decoder. Based on this model, we can form the posterior distribution

$$P(\mathsf{x}|\mathsf{Q}, \mathbf{A}) \propto \prod_{i=1}^{n} P_X(x_i) \prod_{k=1}^{m} 1\left(Q_k = g\left(\frac{1}{\sqrt{n}}\langle A_k, \mathsf{x}\rangle\right)\right) \tag{14.7}$$

to compute the optimal estimator $\mathsf{E}[\mathsf{X}|\mathsf{Q}, \mathbf{A}]$ with respect to the *Minimum Mean Square Error (MMSE)* criterion. Unfortunately, finding the optimal estimator is computationally infeasible unless the dimensions are extremely small. A growing body of recent research, much of which is built on ideas and tools from statistical physics, has been focused on developing computationally feasible estimators that approximate the MMSE estimator and investigates the fundamental limits of the optimal estimator. We shall review some of these works below.

Our goal in this section is to investigate the RD trade-offs for a QCS system. To this end, we first review the Generalized Approximate Message Passing (GAMP) algorithm and apply it to our QCS setting in Sect. 14.3.1. There, we also numerically compare the performance for a Bernoulli-Gaussian source with the rate-distortion (RD) function. Section 14.3.2 conducts a similar study for the case of a distributed Bernoulli-Gaussian source. There, we extend the GAMP algorithm to the two-terminal setting.

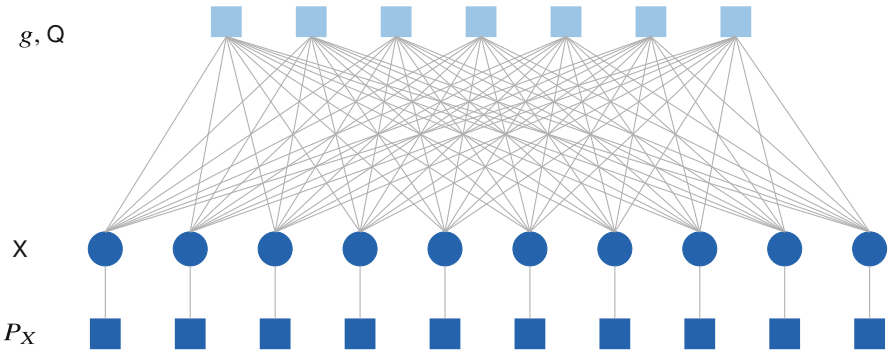



**Fig. 14.5** Graphical model for QCS. The light blue *factor nodes* represent the scalar quantizer $g$ and the observed quantized measurements $\mathsf{Q}$. The dark blue *variable nodes* represent the signal components, each of which has marginal distribution $P_X$

### *14.3.1 Bayesian Compressed Sensing via Approximate Message Passing*

Approximate message passing (AMP) is a computationally efficient iterative thresholding algorithm for large scale CS problems [13]. The paper [53] provided an extension to more general signal priors and elementwise output functions and established the term *Generalized Approximate Message Passing (GAMP)* that is widely used.

We give a brief sketch of the main ideas behind (G)AMP. The starting point for the derivation of AMP are the belief propagation (BP) equations corresponding to its graphical model, see Fig. 14.5. In this graphical model, the square *factor nodes* at the top represent the quantizer $g$ with the observations $\mathsf{Q}$, whereas the circular *variable nodes* represent the signal components about which the distribution $P_X$ is known as an initial condition for the algorithm. The BP algorithm then iteratively exchanges the available information (called *beliefs*) between the variable nodes and factor nodes. Unfortunately, this exchange of information involves tracking complicated probability measures and is unfeasible for applications such as CS. Loosely speaking, this challenge can be tackled by exploiting the fact that mixtures of many random variables tend to become Gaussian by the central limit theorem. Since a Gaussian distribution is fully specified by its mean and variance, these distributions can easily be tracked. Carefully using the central limit theorem and other approximations, one can then reduce the BP iterations to a sequence of matrix-vector multiplications and two scalar inference problems.

We will tailor the GAMP algorithm steps to QCS, as presented in [39]. The first scalar problem is related to the factor nodes. To this end, denote $\mu := \mathsf{E}[X^2]$, let $g : \mathbb{R} \to \{1, \ldots, 2^b\}$ be a quantization function, $(V, W) \overset{\text{iid}}{\sim} \mathcal{N}(0, 1)$, and consider the quantizer output

$$\tilde{Q} = g\left(\sqrt{\eta} \cdot V + \sqrt{\mu - \eta} \cdot W\right) \tag{14.8}$$

for $\eta \in [0, \mu]$. We interpret $\sqrt{\eta}V$ as side information and are interested in estimating $W$ from the quantized measurement. Define the two functions $g_{P_{\text{out}}} : \mathbb{R} \to \mathbb{R}$ and $h_{P_{\text{out}}} : \mathbb{R} \to \mathbb{R}$ via:

$$g_{P_{\text{out}}}(\tilde{q}, v, \mu - \eta; g) = \frac{1}{\sqrt{\mu - \eta}} \mathsf{E}[W \,|\, \tilde{Q} = \tilde{q}, \sqrt{\eta}V = v] \tag{14.9}$$

$$h_{P_{\text{out}}}(\tilde{q}, v, \mu - \eta; g) = \frac{1}{\mu - \eta} \left(1 - \mathsf{Var}[W | \tilde{Q} = \tilde{q}, \sqrt{\eta}V = v]\right). \tag{14.10}$$

The second inference problem is that of estimating a single $X \sim P_X$ from a measurement corrupted by Gaussian noise

$$\tilde{X} = X + N/\sqrt{\mathsf{snr}} \tag{14.11}$$

where $\mathsf{snr} \geq 0$ and $N \sim \mathcal{N}(0, 1)$ independent of $X$. Note that $\tilde{X}$ has a PDF irrespective of whether $X$ is discrete, continuous, or mixed. We define the two functions $g_{P_X} : \mathbb{R} \to \mathbb{R}$ and $h_{P_X} : \mathbb{R} \to \mathbb{R}$ via

$$\begin{aligned} g_{P_X}(\tilde{x}, \mathsf{snr}) &= \mathsf{E}[X \,|\, \tilde{X} = \tilde{x}] \\ h_{P_X}(\tilde{x}, \mathsf{snr}) &= \mathsf{Var}[X | \tilde{X} = \tilde{x}]. \end{aligned} \tag{14.12}$$

Taking vectors as inputs, the functions $g_{P_{\text{out}}}$, $h_{P_{\text{out}}}$, $g_{P_X}$, $h_{P_X}$, and $(\cdot)^{-1}$ are applied component-wise and $\odot$ denotes component-wise multiplication for vectors and matrices. The GAMP algorithm for QCS is given in Algorithm 2.

An important property of (G)AMP algorithms is that their asymptotic performance (as $n, m \to \infty$ with $m/n \to \alpha$) can be predicted via the *State Evolution (SE)*. We define two state variables - one for each scalar inference problem (14.8) and (14.11). The SE then iteratively recomputes the state variables via the functions $h_{P_{\text{out}}}$ and $h_{P_X}$ until convergence. The correctness of SE for GAMP has been proved in [37]. The SE procedure for QCS is given in Algorithm 3.

As an example, consider the Bernoulli-Gaussian spike source with distribution

$$P_X = (1 - p) \cdot \delta_0 + p \cdot \mathcal{N}(0, 1). \tag{14.13}$$

For this source, the estimation functions in (14.12) are

$$g_{P_X}(\tilde{x}, \mathsf{snr}) = \frac{\tilde{x}}{1 + \frac{(1-p)}{p}\sqrt{1 + \mathsf{snr}}\exp\left(-\frac{\mathsf{snr}^2\tilde{x}^2}{2(1+\mathsf{snr})}\right)} \cdot \frac{\mathsf{snr}}{1 + \mathsf{snr}} \tag{14.14}$$

**Algorithm 2:** GAMP for QCS [39]

**initialize**: $\mathsf{y}^0 = \mathsf{E}[\mathsf{X}]$
$\mathsf{v}_x^0 = \mathsf{Var}[\mathsf{X}]$
$\hat{\mathsf{s}}^0 = 0$
**for** $t = 1, 2, 3, \ldots$ **do**
  Factor update:
    $\mathsf{v}_p^t = \frac{1}{n}(\mathbf{A} \odot \mathbf{A})\mathsf{v}_x^{t-1}$
    $\hat{\mathsf{p}}^t = \frac{1}{\sqrt{n}}\mathbf{A}\mathsf{y}^{t-1} - \mathsf{v}_p^t \odot \hat{\mathsf{s}}^{t-1}$
    $\hat{\mathsf{s}}^t = g_{P_{\text{out}}}(\mathsf{q}, \hat{\mathsf{p}}^t, \mathsf{v}_p^t; g)$
    $\mathsf{v}_s^t = h_{P_{\text{out}}}(\mathsf{q}, \hat{\mathsf{p}}^t, \mathsf{v}_p^t; g)$
  Variable update:
    $\mathsf{v}_r^t = \frac{1}{n}(\mathbf{A} \odot \mathbf{A})^{\mathsf{T}}\mathsf{v}_s^t$
    $\hat{\mathsf{r}}^t = \mathsf{y}^{t-1} + (\mathsf{v}_r^t)^{-1} \odot \left(\frac{1}{\sqrt{n}}\mathbf{A}^{\mathsf{T}}\hat{\mathsf{s}}^t\right)$
    $\mathsf{y}^t = g_{P_X}(\hat{\mathsf{r}}^t, \mathsf{v}_r^t)$
    $\mathsf{v}_x^t = h_{P_X}(\hat{\mathsf{r}}^t, \mathsf{v}_r^t)$
**end**
**output** : $\mathsf{y}^t$

**Algorithm 3:** GAMP SE for QCS

**initialize**: $\mu = \mathsf{E}[X^2]$
$\eta_{\text{SE}}^0 = 0$
**for** $t = 1, 2, 3, \ldots$ **do**
  Factor update:
    $\mathsf{snr}^t = \alpha \cdot \mathsf{E}_{VY}\left[h_{P_{\text{out}}}(Q, V, \mu - \eta_{\text{SE}}^t; g)\right]$
  Variable update:
    $\eta_{\text{SE}}^t = \mu - \mathsf{E}_{\tilde{X}}\left[h_{P_X}\left(\tilde{X}, \mathsf{snr}^t\right)\right]$
**end**
**output** : $\text{MSE} = \mu - \eta_{\text{SE}}^t$

$$h_{P_X}(\tilde{x}, \mathsf{snr}) = \frac{1}{1 + \frac{(1-p)}{p}\sqrt{1+\mathsf{snr}}\exp\left(-\frac{\mathsf{snr}^2\tilde{x}^2}{2(1+\mathsf{snr})}\right)}\left(\frac{1}{1+\mathsf{snr}} + \left(\frac{\mathsf{snr}\cdot\tilde{x}}{1+\mathsf{snr}}\right)^2\right) - g_{P_X}(\tilde{x}, \mathsf{snr})^2. \tag{14.15}$$

The estimation functions on the quantizer side, Eqs. (14.9)–(14.10), are

$$g_{P_{\text{out}}}(\tilde{q}, v, \mu - \eta; g) = \frac{1}{\mu - \eta}\big(\mathsf{E}[Z \,|\, g(Z) = \tilde{q}] - v\big), \qquad Z \sim \mathcal{N}(v, \mu - \eta) \tag{14.16}$$

$$h_{P_{\text{out}}}(\tilde{q}, v, \mu - \eta; g) = \frac{1}{\mu - \eta}\left(1 - \frac{\mathsf{Var}[Z|g(Z) = \tilde{q}]}{\mu - \eta}\right), \quad Z \sim \mathcal{N}(v, \mu - \eta). \tag{14.17}$$

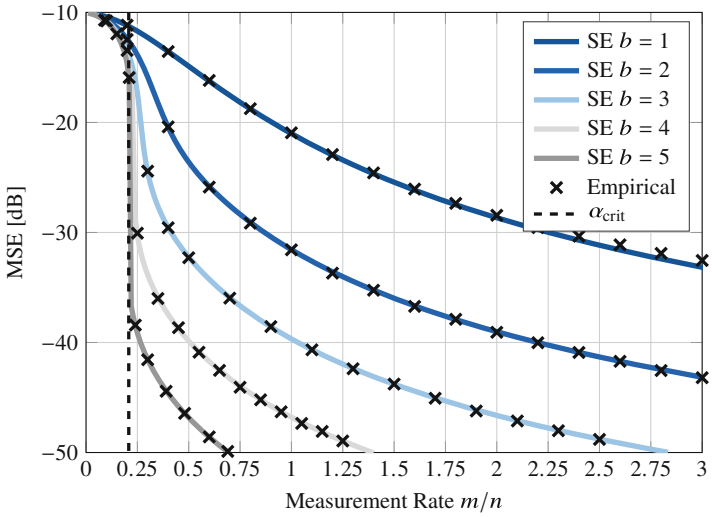



**Fig. 14.6** GAMP performance as predicted by SE and empirically observed for a Bernoulli-Gaussian source with $p = 0.1$ with $n = 5000$

Since $Z$ is a truncated Gaussian on the event $\{g(Z) = \tilde{q}\}$ for some $\tilde{q} \in \mathcal{Q}$, the above expectation and variance can easily be calculated numerically in terms of the Gaussian probability and cumulative density functions.

In Fig. 14.6, we compare the SE predictions of the asymptotic MSE with the errors empirically observed through simulations for different $b$ and $\alpha$. Here, we chose $P_X$ to be Bernoulli-Gaussian with $p = 0.1$ and the signal length $n = 5000$. For $b \geq 2$, we choose $g$ such that each quantization interval has probability $2^{-b}$ under the Gaussian measure with mean zero and variance $\mu$. For each $b$ and $\alpha$ we plot the median MSE of 250 experiments.

We further show the critical measurement rate $\alpha_{\text{crit}}$, at which the phase transition to perfect recovery happens for Gaussian matrices with noiseless and unquantized measurements. While the optimal estimator achieves perfect reconstruction for $\alpha > p$ even for Gaussian matrices, this is not the case for AMP [43, 44]. In this case, we can use SE to compute $\alpha_{\text{crit}} \approx 0.21$.

As expected, the MSE decreases with increasing $b$. Further, there is a sharp decline in the MSE for $\alpha > \alpha_{\text{crit}}$, which matches the phase transition in the limit of infinite quantization rate. We conclude that the SE predictions for these parameters are very accurate. For $\alpha \gg \alpha_{\text{crit}}$, the error decreases slowly in $\alpha$ as we are effectively *oversampling* the signal which is known to yield an error decrease inversely proportional to the sampling rate, see [35, Thm. 1].

### 14.3.2 Two-Terminal Bayesian QCS

AMP was extended to a distributed setting in [24] for unquantized two-terminal CS and termed *Multi-Terminal Approximate Message Passing (MAMP)*. This section combines the GAMP and MAMP algorithms for the distributed problem.

Formally, we have two generalized linear models

$$\begin{aligned} Q_1[k] &= g_1\Big(\frac{1}{\sqrt{n}}\langle \mathsf{A}_k^{(1)}, \mathsf{X}_1\rangle\Big), \qquad 1 \le k \le m_1 \\ Q_2[k] &= g_2\Big(\frac{1}{\sqrt{n}}\langle \mathsf{A}_k^{(2)}, \mathsf{X}_2\rangle\Big), \qquad 1 \le k \le m_2, \end{aligned} \tag{14.18}$$

where

- $(\mathsf{X}_1, \mathsf{X}_2)$ are output by a memoryless source with distribution $P_{X_1X_2}$,
- $\mathbf{A}^{(1)} \in \mathbb{R}^{m_1\times n}$ and $\mathbf{A}^{(2)} \in \mathbb{R}^{m_2\times n}$ are the measurement matrices, each with iid $\mathcal{N}(0, 1)$ entries, and $\mathsf{A}_k^{(j)}$ is the transposed $k$th row of $\mathbf{A}^{(j)}$,
- $g_1 : \mathbb{R} \to \mathcal{Q}_1$ and $g_2 : \mathbb{R} \to \mathcal{Q}_2$ are two quantization functions with $b_1$ and $b_2$ bits, respectively.

The graphical model for this setting is depicted in Fig. 14.7. We see that the two terminals are connected only via the knowledge of the joint distribution of the two signals. To get the *Multi-Terminal Generalized Approximate Message Passing (MGAMP)* reconstruction algorithm, we combine the GAMP and MAMP steps in an obvious way without giving any formal derivations. To this end, recall the two scalar channels (14.8) and (14.11). The first channel was related to the quantization of the measurements. As this happens individually in the two terminals, those factor updates are also done individually in the MGAMP algorithm and we can reuse the functions $g_{P_{\text{out}}}$ and $h_{P_{\text{out}}}$ given in (14.9)–(14.10). For the additive noise channel in Eq. (14.11), we now have two parallel noise channels

$$\begin{aligned} \tilde{X}_1 &= X_1 + Z_1/\sqrt{\mathsf{snr}_1} \\ \tilde{X}_2 &= X_2 + Z_2/\sqrt{\mathsf{snr}_2}, \end{aligned} \tag{14.19}$$

where $(X_1, X_2) \sim P_{X_1X_2}$ and $Z_1$ and $Z_2$ are independent of each other and $(X_1, X_2)$, and each have distribution $\mathcal{N}(0, 1)$. Define the functions $g^{(1)}_{P_{X_1X_2}}$, $g^{(2)}_{P_{X_1X_2}}$, $h^{(1)}_{P_{X_1X_2}}$, and $h^{(2)}_{P_{X_1X_2}}$ (all $\mathbb{R}^2 \to \mathbb{R}$) via

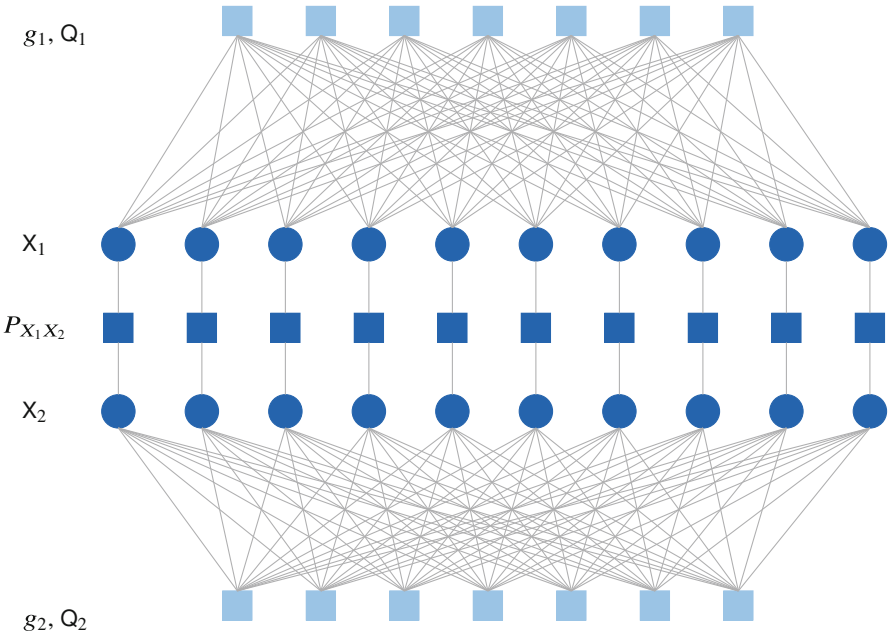



**Fig. 14.7** Graphical model for two-terminal QCS. The light blue *factor nodes* represent the scalar quantizers and the observed quantized measurements at each terminal. The dark blue *variable nodes* represent the signal components of the two terminals and the knowledge of their joint distribution

$$\begin{aligned} g^{(1)}_{P_{X_1X_2}}(\tilde{x}_1, \tilde{x}_2, \mathsf{snr}_1, \mathsf{snr}_2) &= \mathsf{E}[X_1 \,|\, \tilde{X}_1 = \tilde{x}_1, \tilde{X}_2 = \tilde{x}_2] \\ g^{(2)}_{P_{X_1X_2}}(\tilde{x}_1, \tilde{x}_2, \mathsf{snr}_1, \mathsf{snr}_2) &= \mathsf{E}[X_2 \,|\, \tilde{X}_1 = \tilde{x}_1, \tilde{X}_2 = \tilde{x}_2] \\ h^{(1)}_{P_{X_1X_2}}(\tilde{x}_1, \tilde{x}_2, \mathsf{snr}_1, \mathsf{snr}_2) &= \mathsf{Var}[X_1 | \tilde{X}_1 = \tilde{x}_1, \tilde{X}_2 = \tilde{x}_2] \\ h^{(2)}_{P_{X_1X_2}}(\tilde{x}_1, \tilde{x}_2, \mathsf{snr}_1, \mathsf{snr}_2) &= \mathsf{Var}[X_2 | \tilde{X}_1 = \tilde{x}_1, \tilde{X}_2 = \tilde{x}_2]. \end{aligned} \tag{14.20}$$

For vectors, these functions are again applied component-wise. The MGAMP algorithm is described more precisely in Algorithm 4. Similarly, the behavior of MGAMP can be predicted by its SE, which is given in Algorithm 5.

As an example, we perform MGAMP experiments and compute the SE predictions for a distributed Bernoulli-Gaussian spike source

$$P_{X_1X_2} = (1-p)\cdot\delta_0 + p\cdot\mathcal{N}\left(0, \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}\right) \tag{14.21}$$

for some $\rho \in (-1, 1)$. The scalar quantizers $g_1$ and $g_2$ are again chosen to maximize the entropies of their outputs, i.e., they partition the real line into intervals of equal

**Algorithm 4:** MGAMP for QCS

**initialize**: for $j = 1, 2$, set

$$\mu_j = \mathsf{E}\left[X_j^2\right]$$
$$\mathsf{y}_j^0 = \mathsf{E}\left[\mathsf{X}_j\right]$$
$$\mathsf{v}_{x_j}^0 = \mathsf{Var}\left[\mathsf{X}_j\right]$$
$$\hat{\mathsf{s}}_j^0 = \mathsf{0}$$

**for** $t = 1, 2, 3, \ldots$ **do**

  Factor update: for $j = 1, 2$, set

    $\mathsf{v}_{p_j}^t = \frac{1}{n}\left(\mathbf{A}^{(j)} \odot \mathbf{A}^{(j)}\right)\mathsf{v}_{x_j}^{t-1}$

    $\hat{\mathsf{p}}^t = \frac{1}{\sqrt{n}}\mathbf{A}^{(j)}\mathsf{y}_j^{t-1} - \mathsf{v}_{p_j}^t \odot \hat{\mathsf{s}}_j^{t-1}$

    $\hat{\mathsf{s}}_j^t = g_{P_{\text{out}}}(\mathsf{q}_j, \hat{\mathsf{p}}_j^t, \mathsf{v}_{p_j}^t; g_j)$

    $\mathsf{v}_{s_j}^t = h_{P_{\text{out}}}(\mathsf{q}_j, \hat{\mathsf{p}}_j^t, \mathsf{v}_{p_j}^t; g_j)$

  Variable update:

    Linear step: for $j = 1, 2$, set

      $\mathsf{v}_{r_j}^t = \frac{1}{n}\left(\mathbf{A}^{(j)} \odot \mathbf{A}^{(j)}\right)^{\mathsf{T}}\mathsf{v}_{s_j}^t$

      $\hat{\mathsf{r}}_j^t = \mathsf{y}_j^{t-1} + (\mathsf{v}_{r_j}^t)^{-1} \odot \left(\frac{1}{\sqrt{n}}\mathbf{A}^{(j)\mathsf{T}}\hat{\mathsf{s}}_j^t\right)$

    Nonlinear step: for $j = 1, 2$, set

      $\mathsf{y}_j^t = g_{P_{X_1X_2}}^{(j)}(\hat{\mathsf{r}}_1^t, \hat{\mathsf{r}}_2^t, \mathsf{v}_{r_1}^t, \mathsf{v}_{r_2}^t)$

      $\mathsf{v}_{x_j}^t = h_{P_{X_1X_2}}^{(j)}(\hat{\mathsf{r}}_1^t, \hat{\mathsf{r}}_2^t, \mathsf{v}_{r_1}^t, \mathsf{v}_{r_2}^t)$

**end**

**output** : $\mathsf{y}_1^t, \mathsf{y}_2^t$

**Algorithm 5:** MGAMP State Evolution for QCS

**initialize**: for $j = 1, 2$, set

$$\mu_j = \mathsf{E}\left[X_j^2\right]$$
$$\eta_j^0 = 0$$

**for** $t = 1, 2, 3, \ldots$ **do**

  Factor update: for $j = 1, 2$, set

    $\mathsf{snr}_j^t = \alpha_j\, \mathsf{E}_{QY}\left[h_{P_{\text{out}}}(Q, V, \mu_j - \eta_j^t; g_j)\right]$

  Variable update: for $j = 1, 2$, set

    $\eta_i^t = \mu_j - \mathsf{E}_{\tilde{X}_1\tilde{X}_2}\left[h_{P_{X_1X_2}}^{(j)}\left(\tilde{X}_1, \tilde{X}_2, \mathsf{snr}_1^t, \mathsf{snr}_2^t\right)\right]$

**output** : MSE $\mu_j - \eta_j^t$ for $j = 1, 2$

probability under the Gaussian measure. Let $\tilde{\mathsf{x}} = [\tilde{x}_1, \tilde{x}_2]^{\mathsf{T}}$. For this source, the estimation functions in (14.12) can be computed to be

$$g_{P_{X_1X_2}}^{(1)}(\tilde{x}_1, \tilde{x}_2, \mathsf{snr}_1, \mathsf{snr}_2) = \frac{1}{1 + \frac{(1-p)}{p}\frac{\mathcal{N}(\tilde{\mathsf{x}}; 0, \Sigma_0)}{\mathcal{N}(\tilde{\mathsf{x}}; 0, \Sigma_1)}} \cdot \begin{bmatrix}1 & \rho\end{bmatrix}\Sigma_1^{-1}\tilde{\mathsf{x}} \tag{14.22}$$

and

$$h^{(1)}_{P_{X_1X_2}}(\tilde{x}_1, \tilde{x}_2, \mathsf{snr}_1, \mathsf{snr}_2)$$
$$= \frac{1 - \begin{bmatrix}1 & \rho\end{bmatrix} \Sigma_1^{-1} \begin{bmatrix}1 \\ \rho\end{bmatrix} + \left(\begin{bmatrix}1 & \rho\end{bmatrix} \Sigma_1^{-1} \tilde{\mathbf{x}}\right)^2}{1 + \frac{(1-p)}{p} \frac{\mathcal{N}(\tilde{\mathbf{x}}; \mathbf{0}, \Sigma_0)}{\mathcal{N}(\tilde{\mathbf{x}}; \mathbf{0}, \Sigma_1)}} - g^{(1)}_{P_{X_1X_2}}(\tilde{x}_1, \tilde{x}_2, \mathsf{snr}_1, \mathsf{snr}_2)^2, \tag{14.23}$$

where

$$\Sigma_0 = \begin{bmatrix} 1/\mathsf{snr}_1 & 0 \\ 0 & 1/\mathsf{snr}_2 \end{bmatrix} \quad \text{and} \quad \Sigma_1 = \begin{bmatrix} 1 + 1/\mathsf{snr}_1 & \rho \\ \rho & 1 + 1/\mathsf{snr}_2 \end{bmatrix}. \tag{14.24}$$

The functions $g^{(2)}_{P_{X_1X_2}}$ and $h^{(2)}_{P_{X_1X_2}}$ are computed similarly. Since the functions $g_{P_{\text{out}}}$ and $h_{P_{\text{out}}}$ depend only on the quantizer and are computed individually in the two terminals, we can reuse (14.16)–(14.17).

For our experiments, we chose the measurement rates and quantizers to be the same at both terminals. Thus, the average MSE is also the same at both terminals. Figure 14.8a plots the SE and experimental results for $p = 0.1$, $n = 5000$ and the correlation coefficient $\rho = 0.9$. Observe that SE again accurately predicts the experimental performance. Figure 14.8b compares the SE predictions for $\rho = 0.9$ (solid lines) and $\rho = 0$ (dotted lines). Observe that for small measurement rates, a high correlation can be exploited to reduce the estimation error. For larger rates, the performance is nearly identical in both cases. Observe also that the phase transition is reduced to a measurement rate of approximately $\alpha_{\text{crit}} \approx 0.15$ (as compared to $\alpha_{\text{crit}} \approx 0.21$ in Fig. 14.6) at each terminal.

## 14.4 Signal Processing in Earth Observation

In the Earth observation (EO) task, the continuous improvement of signal and image processing techniques is the key to retrieving valuable information from the ever-growing remote sensing data acquired every day. Information retrieval can be usually performed by solving the general inverse problem given in (14.1), where the measurements in $Y$ have significantly smaller dimension than the sparse, jointly sparse, low-rank signals in $X$. The matrix $A$ represents the system information with the respect to the variable $X$ in a nonlinear form. $N_s$ and $N_m$ are specified as model errors and noises, and the function $g(\cdot)$ takes into account quantization effects while acquiring the measurements.

Two categories of applications incorporating models of the type shown in (14.1), which are of particularly high demand for current and next-generation optical EO missions, will be unfolded in the scope of the following chapter.

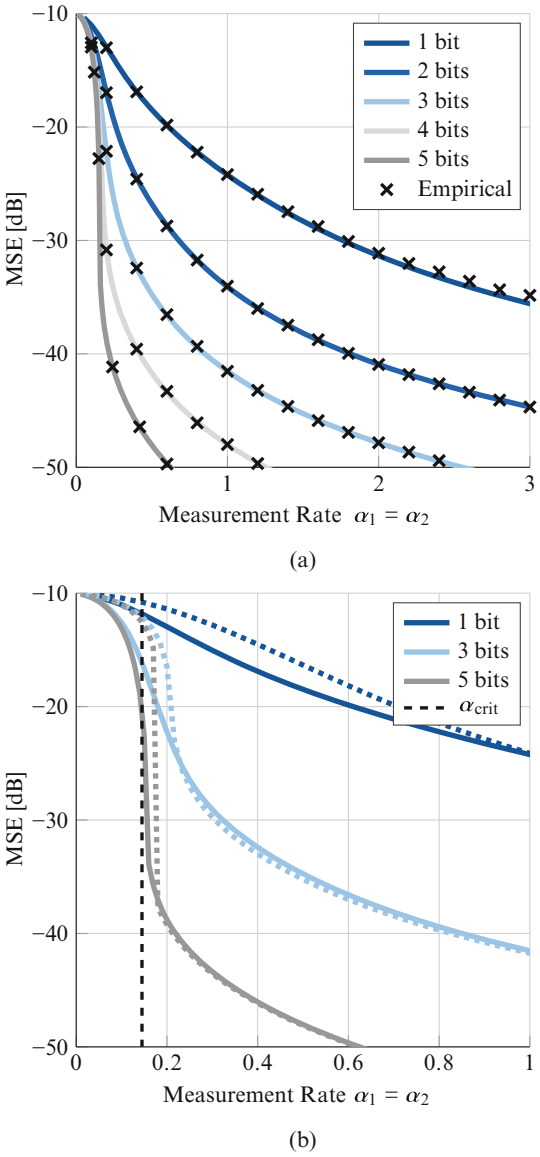



**Fig. 14.8** Comparison of SE and empirical performance for MGAMP with equal measurement rates and quantizers at both terminal. The MSE is the same at both terminals. (**a**) $p = 0.1$, $\rho = 0.9$. (**b**) $\rho = 0.9$ corresponds to the solid line and $\rho = 0$ to the dotted lines

### *14.4.1 Multi-Sensor and Multi-Resolution Data Fusion*

Remotely sensed optical imaging systems compromise on either detailed spectral mapping capacity, which allows to discriminate and classify materials, or high-spatial resolution, which elucidates the scene geometry. Figure 14.9 illustrates

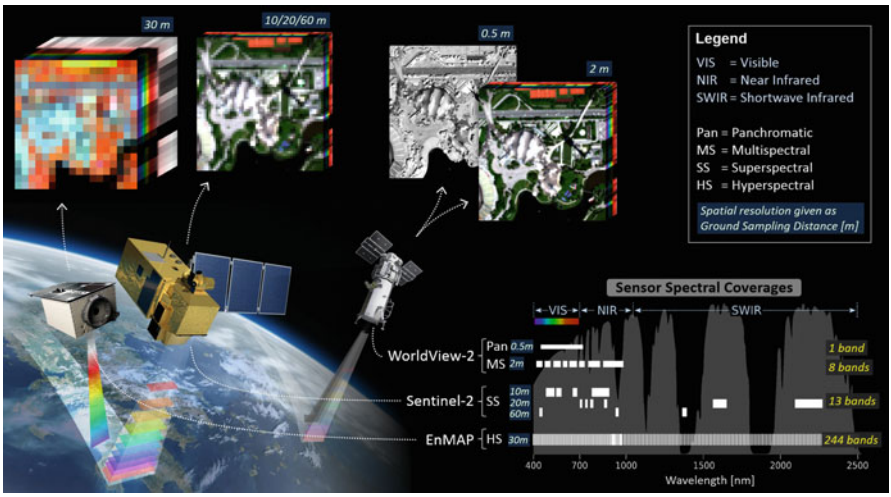



**Fig. 14.9** Illustration of the trade-off between spectral and spatial resolution for different optical remote sensing instruments [23], e.g., WorldView-2, Sentinel-2, and EnMAP

the trade-off between spatial and spectral resolution of remote sensing images, including several well-known EO satellite missions, i.e.,

- EnMAP: The next-generation German hyperspectral (HS) instrument featuring 244 spectral bands with 30 m ground sampling distance (GSD);
- Sentinel-2: A superspectral (SS) instrument featuring 13 spectral bands with 10 m, 20 m, and 60 m GSD;
- WorldView-2: A platform carrying both a multispectral (MS) image with 8 bands at a 2 m GSD and a single broadband panchromatic (PAN) sensor at a GSD of 0.5 m.

The trade-off between spectral and spatial resolution has been long a great challenging issue for monitoring heterogeneous surfaces, such as urban areas that are characterized by both high levels of spatial detail and large variants of materials and objects. Multi-sensor or multi-resolution fusion (e.g., HS and MS data fusion) is indispensable for mitigating the physical limitations associated with individual sensors to a great extent [69]. The idea is to fuse data acquired by two complementary instruments to obtain both high-spectral and high-spatial resolution.

Since the dimension of the fusion product is always higher than the summed dimensions of the input data, a difficulty arises in maximizing both spatial and spectral resolutions due to the very large number of degrees of freedom. More precisely, the larger the difference between the spatial resolutions or numbers of spectral channels of the input data, the larger the number of degrees of freedom. Consequently, the fusion problem becomes increasingly ill-conditioned and fusion products deviate significantly from the ground truth [31, 70].

Recently, much research was conducted by many well-known groups to render this type of problem well-posed by inferring prior knowledge about sparse or jointly sparse representations. The approach followed in the most recent studies is a sparsity-promoting model with an error term, denoted as $f(Z, Y_h, Y_l)$ corresponding to an underdetermined system, which is derived from a sensor observation model. This model describes the relationship between the target high-spatial and high-spectral resolution image $Z$, the high-spatial resolution measurement (first observed image) $Y_h$ and the high-spectral resolution measurement (second observed image) $Y_l$ [64]. The system can be written as

$$\min_{Z',X} f(Z', Y_h, Y_l) + \lambda h(Z', X), \tag{14.25}$$

where $\lambda \in \mathbb{R}$ is a regularization parameter and $Z' = \phi(Z)$ is a low-dimensional (compressed) representation of the high-dimensional target image $Z$. The regularization term can be expressed as follows

$$h(Z', X) = ||Z' - \mathcal{P}(A, X)||_p^q, \tag{14.26}$$

where $\mathcal{P}$ denotes an operator that generates image patches as products of a set $A$ of dictionaries with corresponding sparse coefficients $X$, and that forms this set of all patches into a full 3-D image (one spectral and two spatial dimensions).

In [74], the (jointly) sparse coefficients in the $i$-th patch $X_i$ under reconstruction are estimated from the local patch measurements $Y_i$ by solving a system that can be generalized to a matrix version of the model (14.1) as follows

$$Y_i = g_i(A_i X_i + N_{s,i}) + N_{m,i}. \tag{14.27}$$

In the simplest case, with $Y_i = [y_{i,1}, y_{i,2}, ..., y_{i,d}]$ describing patch measurements in $d$ adjacent correlated spectral channels, the joint sparse property of the coefficients $X_i = [x_{i,1}, x_{i,2}, ..., x_{i,d}]$ corresponding to correlated channels is promoted via the group sparsity ($\ell_{2,1}$-norm):

$$\min_{X_i} \frac{1}{2}||A_i X_i - Y_i||_F^2 + \mu||X_i||_{2,1}. \tag{14.28}$$

Beyond the sparse representation based approaches, enormous efforts have been recently made to enhance the fusion performance using deep learning (DL) models (owing to their powerful ability in data representation). In our work [69], we found that DL-based fusion methods can be also described as a special case of the general model (14.1), which can be formulated as

$$\begin{aligned} Y_h &= g_h(A\phi_h(X) + N_{s,h}) + N_{m,h}, \\ Y_l &= g_l(\phi_l(A)X + N_{s,l}) + N_{m,l}, \end{aligned} \tag{14.29}$$

where $\phi_h(X)$ and $\phi_l(A)$ can be defined as the linear function with respect to the variables $X$ and $A$, respectively, i.e., $\phi_h(X) = XR$, $\phi_l(A) = CA$. $C$ and $R$ represent the point spread function (PSF) and spectral response function (SRF) from the high-resolution HS image to the high-resolution MS image and the low-resolution HS image, respectively. This is a typically coupled spectral unmixing framework for HS and MS image fusion [45]. Following it, we developed a coupled unmixing network with a well-designed cross-attention module for unsupervised HS superresolution, called Coupled Unmixing Nets with Cross-Attention (CUCaNet). In CUCaNet, the functions $\phi_h$ and $\phi_l$ can be estimated by learning two-stream convolutional neural networks (CNNs), which can be solved by minimizing the following optimization problem [68]:

$$\min_{A,X} \frac{1}{2}||Y_h - A\phi_h(X)||_F^2 + \frac{1}{2}||Y_l - \phi_l(A)X||_F^2, \tag{14.30}$$

with the physical constraints $A \geq 0$, $X \geq 0$, $1^\top X = 1^\top$. By optimizing Eq. (14.28), we then obtain the to-be-estimated high-resolution HS image by $AX$. Figure 14.10 illustrates the proposed CUCaNet from the spectral unmixing perspective.

We highlight the fusion performance of our CUCaNet on an example data, i.e., Pavia University HS data, which has been widely applied for various applications [32, 33], in comparison with several state-of-the-art fusion methods, as shown in Fig. 14.11. These advanced methods include GSA [1], CNMF [71], CSU [45], FUSE [65], HySure [55], NSSR [12], STEREO [40], CSTF [46], LTTR [11], unsupervised uSDN [52], and supervised MHFnet [67]. In general, our designed nonlinear system, CUCaNet, tends to recover more detailed information of high-resolution HS images, showing the superiority in the multi-sensor and multi-resolution data fusion task.

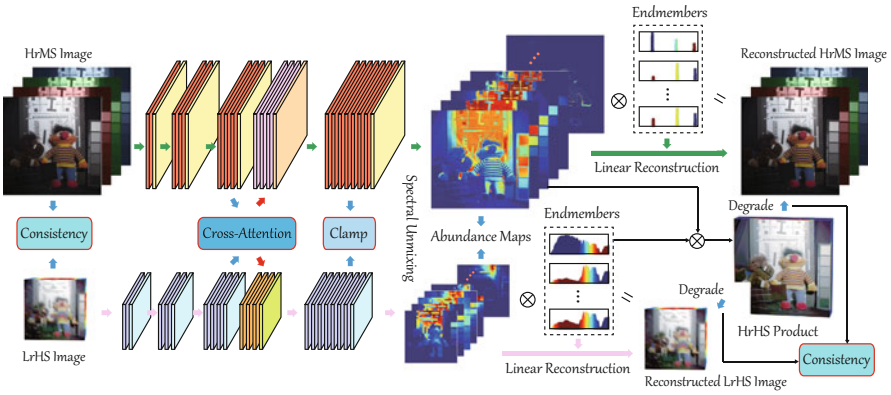



**Fig. 14.10** An illustration of our CUCaNet from the spectral unmixing perspective

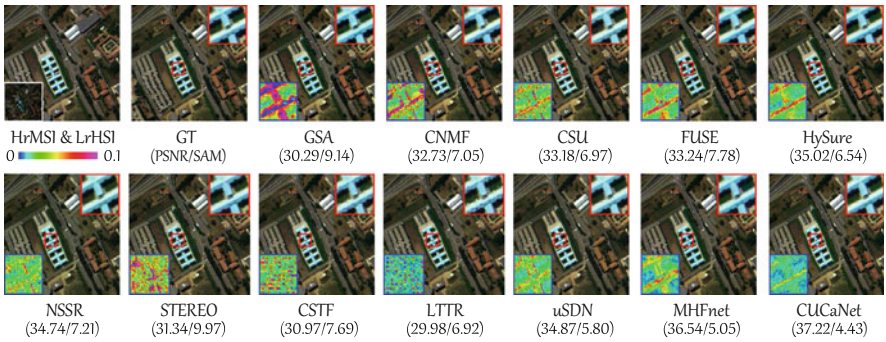



**Fig. 14.11** The fusion performance on the Pavia University dataset (cropped area) of different compared approaches. Region of interests (ROIs) are zoomed in 3 times in the right top corner, and the residual maps between enhanced images and ground truth maps are shown in the left lower corner. The best results are shown in bold

### *14.4.2 Hyperspectral Unmixing Accounting for Spectral Variability*

The low spatial resolution of remotely sensed HS sensors causes multiple materials to be merged in single pixels. That is, the spectral profiles corresponding to individual pixels are compositions of multiple material spectral signatures. Spectral unmixing aims at identifying all contributing materials and their relative contributions to the mixed pixels' spectral signals. Considering the large number of potential candidate materials related to the small number of materials actually contributing to each pixel, sparse property is a natural prior knowledge for RS image processing.

Recently, advanced models were developed that represent real-world scattering and mixing scenarios more accurately than the commonly used Linear Mixing Model (LMM). However, one main factor, i.e., nonlinearity, hinders the LMM's ability to accurately unmix the HS data. In HS imaging, spectral signals usually suffer from the nonlinear mixing, due to the multiple scattering and intimate mixing of materials. As another factor, a source of errors in the unmixing process is the spectral material variability of identical and nearly identical materials in multiple measurements under different illumination, atmospheric, and observation conditions [27, 29].

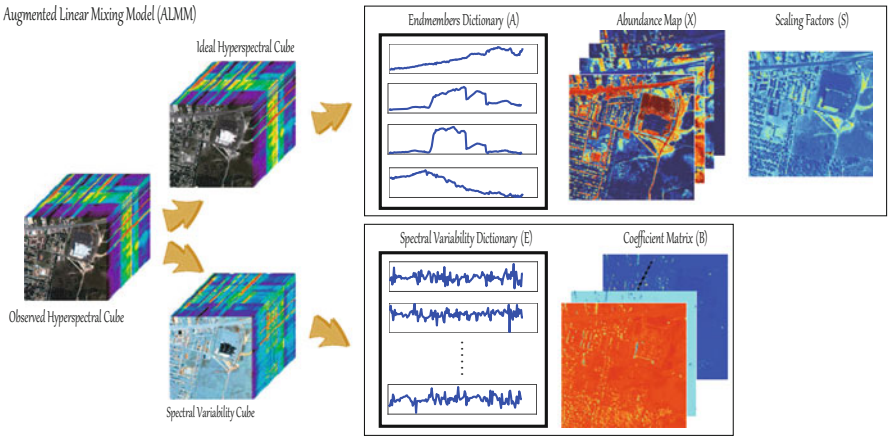



**Fig. 14.12** Model illustration of our proposed ALMM for spectral unmixing

We include these insights in our recent Augmented Linear Mixing Model (ALMM) [29], which further develops the Extended Linear Mixing Model (ELMM) [14] and the spectral variability-aware Perturbed Linear Mixing Model (PLMM) [59]. The ALMM unmixing model is given by

$$Y = g((A + \Delta A)X + N_s) + N_m, \tag{14.31}$$

where $A$ and $\Delta A$ are called spectral endmember and variability matrices, respectively. In our work [29], we approximate nonlinear scattering effects by the following simplified version of the ALMM model:

$$Y = AXS + EB + N, \tag{14.32}$$

where $S$ denotes the scaling factors, $E$ is the spectral variability matrix, and $B$ represents the coefficients corresponding to the variable $E$. Figure 14.12 illustrates these quantities based on representative HS data.

One way to obtain the unknown quantities in (1.24), as proposed in [29], is to solve an optimization problem that is penalized by multiple regularization terms to enforce physically meaningful properties:

$$\begin{aligned} \min_{X,B,S,E} \ & \frac{1}{2}||Y - AXS - EB||_F^2 + \alpha\Phi(X) + \beta\Psi(B) + \gamma\Upsilon(E) \\ & \text{s.t. } \mathrm{X} \succeq 0, \ \mathrm{S} \succeq 0. \end{aligned} \tag{14.33}$$

The regularization terms for the abundances $X$, spectral variability coefficients $B$, and spectral variability dictionary $E$ approximate physical conditions well by using the following models:

$$\begin{aligned}\Phi(X) &= ||X||_{1,1} = \sum_{k=1}^{N} ||x_k||_1, \\ \Psi(B) &= \frac{1}{2}||B||_F^2, \\ \Upsilon(E) &= \frac{1}{2}||A^\top E||_F^2 + \frac{1}{2}||E^\top E - I||_F^2.\end{aligned} \tag{14.34}$$

In addition, non-negativity, i.e., $X \succeq 0$, $S \succeq 0$, is usually taken into account to meet the reasonable physical assumptions in the spectral unmixing process. Also, the sum-to-one constraint is another important prior that needs to be considered and imposed on the abundance maps. The variables $\mathbf{X}$ and $\mathbf{S}$ are bundled together, e.g., Eq. (14.33), further leading to the difficulty to satisfy the sum-to-one constraint on $\mathbf{X}$ in the practical case. As a trade-off, our ALMM work approximates the strong constraint by adopting a scaled constrained least squares unmixing.

Differently from the ALMM's refined modeling, the new subspace-based unmixing framework is developed, called subspace unmixing with low-rank attribute embedding (SULoRA) [27], being capable of handling various spectral variabilities in a more robust and generalized fashion. It is well known that the signals in the high-dimensional space are complex and noisy. Directly processing and performing spectral unmixing on such signals is really challenging. For this reason, a creative subspace-based unmixing strategy is mathematically modeled as

$$\begin{aligned}Y &= Y^{'} + N^{'}, \ \text{s.t.}\ \mathrm{Y}^{'} = \Theta \mathrm{Y}, \\ Y^{'} &= \Theta A X + N^{''},\end{aligned} \tag{14.35}$$

where $\Theta$ is a subspace projection with the low-rank attribute, $Y^{'}$ is the corresponding subspace representation of the original HS data $Y$ after the low-rank attribute embedding, and $N'$ and $N''$ denote different-level model noises.

According to the subspace model in Eq. (14.35), spectral unmixing can be conducted by optimizing the following constrained optimization problem:

$$\min_{X,\Theta} \ \frac{1}{2}||\Theta(Y - AX)||_F^2 + \Phi(\Theta) + \Upsilon(X) \ \ \text{s.t.}\ \mathrm{X} \succeq 0. \tag{14.36}$$

The problem (14.36) is highly ill-posed, and a feasible solution to solve this problem is to regularize the variables $X$ and $\Theta$ by adding prior knowledge. We then have the two following regularization terms $\Phi(\Theta)$ and $\Upsilon(X)$ with respect to the variables $\Theta$ and $X$ below.

- Subspace Regularization $\Phi(\Theta)$: we regularize the subspace projections $\Theta$ with a low-rank attribute to transfer the original HS data into a more robust subspace, which is approximately formulated in the form of nuclear norm, i.e., $||\Theta||_*$. More specifically, we aim at learning a low-rank subspace projection that can be

considered as a correlative filtering bank to address various spectral variabilities and meanwhile reduce the computational cost to a great extent. Beside, we also expect to make the structural information consistent as much as possible. The prior can be modeled as $||Y - \Theta Y||_F$. To sum up, the resulting subspace regularization term can be written as follows

$$\Phi(\Theta) = \frac{\alpha}{2}||Y - \Theta Y||_F^2 + \beta||\Theta||_*. \tag{14.37}$$

- Abundance Regularization $\Upsilon(X)$: due to the material sparsity about abundance maps in the HS scene, this abundance regularization term parameterized by the penalty parameter $\gamma$ can be represented by

$$\Upsilon(X) = \gamma||X||_{1,1}. \tag{14.38}$$

Similarly to our ALMM model, scaled constrained least squares unmixing is applied to estimate the scaling factors of endmembers to relax the sum-to-one constraint in the unmixing model. An alternating direction method of multipliers (ADMM) solver [7, 30] is designed to optimize our two models.

Furthermore, due to the limitations of linearized models in data fitting and representation, the ability to accurately unmix the HS data remains to be improved. For this reason, we further developed a self-supervised learning unmixing framework inspired by advanced deep networks, called WU-Net [28]. In WU-Net, endmember extraction and spectral unmixing are jointly performed by two subnetworks, i.e., endmember network, unmixing network, in a nonlinear system. The two subnetworks share the partial to-be-estimated parameters, e.g., $f(W_1, b_1, \bullet)$. The whole system can be modeled as

$$\min_{W_1,b_1,W_2,b_2} \frac{1}{2}||Y - g(W_2, b_2, f(W_1, b_1, Y))||_F^2 + \frac{1}{2}||L - f(W_1, b_1, Y)||_F^2, \tag{14.39}$$

where $f$ and $g$ denote the encoder and decoder networks with respect to the weights $W$ and biases $b$, respectively, and $L$ is the pseudo endmembers extracted by existing endmember extraction methods, e.g., vertex component analysis (VCA) [47]. Figure 14.13 illustrates the proposed unmixing framework (WU-Net) based on a nonlinear system.

Similarly, we also visualize the abundance maps to qualitatively evaluate the unmixing performance of different advanced unmixing algorithms on the AVIRIS Jasper Ridge dataset, as shown in Fig. 14.14. They are

- Non-DL (linearized) unmixing approaches: fully constrained least squares unmixing (FCLSU) [25], partial constrained least squares unmixing (PCLSU) [26], sparse unmixing by variable splitting and augmented Lagrangian (SUnSAL) [5], subspace unmixing with low-rank attribute embedding (SULoRA) [27], and ALMM [29];

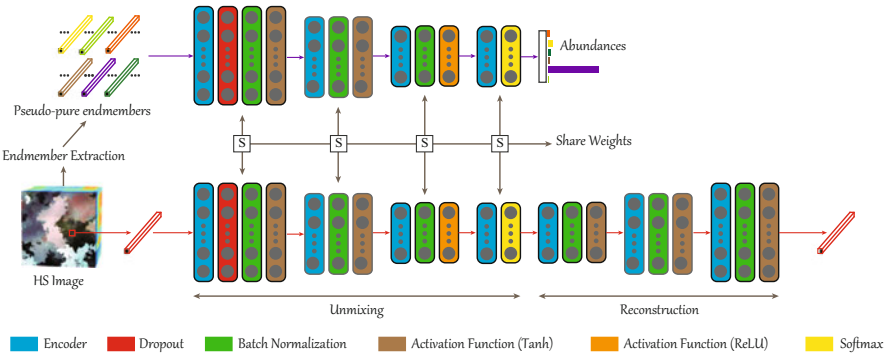



**Fig. 14.13** An overview illustration for the proposed WU-Net architecture in spectral unmixing

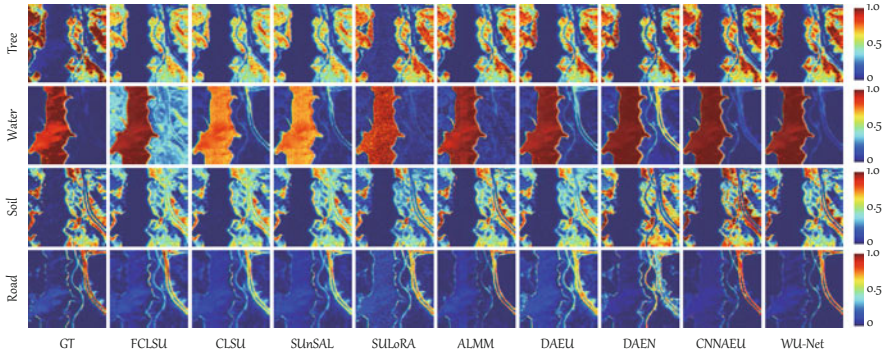



**Fig. 14.14** Qualitative comparison of abundance maps using different unmixing methods on the AVIRIS Jasper Ridge dataset. The first column is the ground truth (GT) of abundance maps for four different materials

- DL (nonlinearized) unmixing approaches: DAEU [49], deep autoencoder networks (DAEN) [56], CNNAEU [50], and WU-Net [28].

As expected, our linearized methods, i.e., SULoRA, ALMM, yield more similar results with ground truth (GT), while the nonlinear deep method (WU-Net) shows more realistic abundance estimation compared to other competitors.

## References

1. Aiazzi, B., Baronti, S., Selva, M.: Improving component substitution pansharpening through multivariate regression of ms + pan data. IEEE Trans. Geosci. Remote Sens. **45**(10), 3230–3239 (2007)
2. Albertini, F., Sontag, E.D., Maillot, V.: Uniqueness of weights for neural networks. In: Artificial Neural Networks with Applications in Speech and Vision, pp. 115–125. Chapman and Hall, London (1993)

3. Bah, B., Rauhut, H., Terstiege, U., Westdickenberg, M.: Learning deep linear neural networks: Riemannian gradient flows and convergence to global minimizers. arXiv preprint arXiv:1910.05505 (2019)
4. Behboodi, A., Rauhut, H., Schnoor, E.: Generalization bounds for deep thresholding networks. training **3**, 11 (2020)
5. Bioucas-Dias, J., Figueiredo, M.: Alternating direction algorithms for constrained sparse regression: Application to hyperspectral unmixing. In: Proceedings of the WHISPERS, pp. 1–4. IEEE, New York (2010)
6. Blum, A.L., Rivest, R.L.: Training a 3-node neural network is np-complete. Neural Netw. **5**(1), 117–127 (1992). https://doi.org/10.1016/S0893-6080(05)80010-3 http://www.sciencedirect.com/science/article/pii/S0893608005800103
7. Boyd, S., Parikh, N., Chu, E.: Distributed optimization and statistical learning via the alternating direction method of multipliers. Now Publishers Inc (2011)
8. Cohen, A., Daubechies, I., Guleryuz, O., Orchard, M.: On the importance of combining wavelet-based nonlinear approximation with coding strategies. IEEE Trans. Inf. Theory **48**(7), 1895–1921 (2002)
9. Daubechies, I., DeVore, R., Foucart, S., Hanin, B., Petrova, G.: Nonlinear Approximation and (Deep) relu Networks. Constructive Approximation (2021). https://doi.org/10.1007/s00365-021-09548-z
10. DeVore, R., Hanin, B., Petrova, G.: Neural network approximation. arXiv preprint arXiv:2012.14501 (2020)
11. Dian, R., Li, S., Fang, L.: Learning a low tensor-train rank representation for hyperspectral image super-resolution. IEEE Trans. Neural Netw. Learn. Syst. **30**(9), 2672–2683 (2019)
12. Dong, W., Fu, F., Shi, G., Cao, X., Wu, J., Li, G., Li, X.: Hyperspectral image super-resolution via non-negative structured sparse representation. IEEE Trans. Image Process. **25**(5), 2337–2352 (2016)
13. Donoho, D.L., Maleki, A., Montanari, A.: Message passing algorithms for compressed sensing. Proc. US Nat. Acad. Sci. **106**(45), 18914–18919 (2009)
14. Drumetz, L., Veganzones, M.A., Henrot, S., Phlypo, R., Chanussot, J., Jutten, C.: Blind hyperspectral unmixing using an extended linear mixing model to address spectral variability. IEEE Trans. Image Process. **25**(8), 3890–3905 (2016)
15. Elbrächter, D., Grohs, P., Jentzen, A., Schwab, C.: DNN Expression Rate Analysis of High-Dimensional PDEs: Application to Option Pricing. Constr. Approx. **55**(1), 3–71 (2021). https://doi.org/10.1007/s00365-021-09541-6
16. Fefferman, C.: Reconstructing a neural net from its output. Revista Matematica Iberoamericana **10**, 507–555 (1994)
17. Fiedler, C., Fornasier, M., Klock, T., Rauchensteiner, M.: Stable recovery of entangled weights: Towards robust identification of deep neural networks from minimal samples. Appl. Comput. Harmon. Anal. (2021)
18. Fornasier, M., Rauhut, H.: Iterative thresholding algorithms. Appl. Comput. Harmon. Anal. **25**(2), 187–208 (2008). https://doi.org/10.1016/j.acha.2007.10.005. https://www.sciencedirect.com/science/article/pii/S1063520307001157
19. Fornasier, M., Rauhut, H.: Recovery algorithms for vector-valued data with joint sparsity constraints. SIAM J. Numer. Anal. **46**(2), 577–613 (2008). https://doi.org/10.1137/0606668909
20. Fornasier, M., Schnass, K., Vybiral, J.: Learning functions of few arbitrary linear parameters in high dimensions. Found. Comput. Math. **12**(2), 229–262 (2012). https://doi.org/10.1007/s10208-012-9115-y
21. Fornasier, M., Klock, T., Rauchensteiner, M.: Robust and Resource-Efficient Identification of Two Hidden Layer Neural Networks. In: Constructive Approximation (2021). https://doi.org/10.1007/s00365-021-09550-5
22. Fornasier, M., Vybíral, J., Daubechies, I.: Robust and resource efficient identification of shallow neural networks by fewest samples. Information and Inference: A Journal of the IMA **10**(2), 625–695 (2021). https://doi.org/10.1093/imaiai/iaaa036

23. Grohnfeldt, C.H.: Multi-sensor data fusion for multi- and hyperspectral resolution enhancement based on sparse representations. Dissertation, Technische Universität München, München (2017)
24. Haghighatshoar, S.: Compressed sensing of memoryless sources: A deterministic Hadamard construction. Ph.D. thesis, EPFL, New York (2014)
25. Heinz, D.C., Chang, C.I.: Fully constrained least squares linear spectral mixture analysis method for material quantification in hyperspectral imagery. IEEE Trans. Geosci. Remote Sens. **39**(3), 529–545 (2001)
26. Heylen, R., Burazerovic, D., Scheunders, P.: Fully constrained least squares spectral unmixing by simplex projection. IEEE Trans. Geosci. Remote Sens. **49**(11), 4112–4122 (2011)
27. Hong, D., Zhu, X.X.: Sulora: Subspace unmixing with low-rank attribute embedding for hyperspectral data analysis. IEEE J. Sel. Top. Signal Process. **12**(6), 1351–1363 (2018)
28. Hong, D., Chanussot, J., Yokoya, N., Heiden, U., Heldens, W., Zhu, X.X.: Wu-net: A weakly-supervised unmixing network for remotely sensed hyperspectral imagery. In: Proceedings of the IGARSS, pp. 373–376. IEEE, New York (2019)
29. Hong, D., Yokoya, N., Chanussot, J., Zhu, X.X.: An augmented linear mixing model to address spectral variability for hyperspectral unmixing. IEEE Trans. Image Process. **28**(4), 1923–1938 (2019)
30. Hong, D., Yokoya, N., Chanussot, J., Zhu, X.X.: Cospace: Common subspace learning from hyperspectral-multispectral correspondences. IEEE Trans. Geosci. Remote Sens. **57**(7), 4349–4359 (2019)
31. Hong, D., Yokoya, N., Ge, N., Chanussot, J., Zhu, X.X.: Learnable manifold alignment (lema): A semi-supervised cross-modality learning framework for land cover and land use classification. ISPRS J. Photogramm. Remote Sens. **147**, 193–205 (2019)
32. Hong, D., Wu, X., Ghamisi, P., Chanussot, J., Yokoya, N., Zhu, X.X.: Invariant attribute profiles: A spatial-frequency joint feature extractor for hyperspectral image classification. IEEE Trans. Geosci. Remote Sens. **58**(6), 3791–3808 (2020)
33. Hong, D., Gao, L., Yao, J., Zhang, B., Plaza, A., Chanussot, J.: Graph convolutional networks for hyperspectral image classification. IEEE Trans. Geosci. Remote Sens. **59**(7), 5966–5978 (2021)
34. Hong, D., Gao, L., Yokoya, N., Yao, J., Chanussot, J., Du, Q., Zhang, B.: More diverse means better: Multimodal deep learning meets remote-sensing imagery classification. IEEE Trans. Geosci. Remote Sens. **59**(5), 4340–4354 (2021)
35. Jacques, L., Laska, J.N., Boufounos, P.T., Baraniuk, R.G.: Robust 1-bit compressive sensing via binary stable embeddings of sparse vectors. IEEE Trans. Inf. Theory **59**(4), 2082–2102 (2013)
36. Janzamin, M., Sedghi, H., Anandkumar, A.: Beating the perils of non-convexity: Guaranteed training of neural networks using tensor methods. arXiv preprint arXiv:1506.08473 (2015)
37. Javanmard, A., Montanari, A.: State evolution for general approximate message passing algorithms, with applications to compressed sensing. Inf. Inference J. IMA **2**(2), 115–144 (2013)
38. Judd, S.: On the complexity of loading shallow neural networks. J. Complexity **4**(3), 177–192 (1988). https://doi.org/10.1016/0885-064X(88)90019-2. http://www.sciencedirect.com/science/article/pii/0885064X88900192
39. Kamilov, U.S., Goyal, V.K., Rangan, S.: Message-passing de-quantization with applications to compressed sensing. IEEE Trans. Sig. Proc. **60**(12), 6270–6281 (2012)
40. Kanatsoulis, C.I., Fu, X., Sidiropoulos, N.D., Ma, W.K.: Hyperspectral super-resolution: A coupled tensor factorization approach. IEEE Trans. Signal Process. **66**(24), 6503–6517 (2018)
41. Kech, M., Krahmer, F.: Optimal injectivity conditions for bilinear inverse problems with applications to identifiability of deconvolution problems. SIAM J. Appl. Algebra Geom. **1**(1), 20–37 (2017). https://doi.org/10.1137/16M1067469
42. Kileel, J., Pereira, J.M.: Subspace power method for symmetric tensor decomposition and generalized PCA. arXiv preprint arXiv:1912.04007 (2019)

43. Krzakala, F., Mézard, M., Sausset, F., Sun, Y., Zdeborová, L.: Probabilistic reconstruction in compressed sensing: algorithms, phase diagrams, and threshold achieving matrices. J. Stat. Mech: Theory Exp. **2012**(8), P08009 (2012)
44. Krzakala, F., Mézard, M., Sausset, F., Sun, Y.F., Zdeborová, L.: Statistical-physics-based reconstruction in compressed sensing. Phys. Rev. X **2**(2), 021005 (2012)
45. Lanaras, C., Baltsavias, E., Schindler, K.: Hyperspectral super-resolution by coupled spectral unmixing. In: Proceedings of the ICCV, pp. 3586–3594 (2015)
46. Li, S., Dian, R., Fang, L., Bioucas-Dias, J.M.: Fusing hyperspectral and multispectral images via coupled sparse tensor factorization. IEEE Trans. Image Process. **27**(8), 4118–4130 (2018)
47. Nascimento, J.M., Dias, J.M.: Vertex component analysis: A fast algorithm to unmix hyperspectral data. IEEE Trans. Geosci. Remote Sens. **43**(4), 898–910 (2005)
48. Palsson, F., Sveinsson, J.R., Ulfarsson, M.O.: Multispectral and hyperspectral image fusion using a 3-D-convolutional neural network. IEEE Trans. Geosci. Remote Sens. **14**(5), 639–643 (2017)
49. Palsson, B., Sigurdsson, J., Sveinsson, J.R., Ulfarsson, M.O.: Hyperspectral unmixing using a neural network autoencoder. IEEE Access **6**, 25646–25656 (2018)
50. Palsson, B., Ulfarsson, M.O., Sveinsson, J.R.: Convolutional autoencoder for spatial-spectral hyperspectral unmixing. In: Proceedings of the IGARSS, pp. 357–360. IEEE, New York (2019)
51. Palzer, L., Timo, R., Kramer, G.: Compression for letter-based fidelity measures. In: preprint (2017)
52. Qu, Y., Qi, H., Kwan, C.: Unsupervised sparse Dirichlet-net for hyperspectral image super-resolution. In: Proceedings of the CVPR, pp. 2511–2520 (2018)
53. Rangan, S.: Generalized approximate message passing for estimation with random linear mixing. In: Proceedings of the IEEE International Symposium Information Theory, pp. 2168–2172. St. Petersburg, Russia (2011)
54. Rasti, B., Hong, D., Hang, R., Ghamisi, P., Kang, X., Chanussot, J., Benediktsson, J.: Feature extraction for hyperspectral imagery: The evolution from shallow to deep: Overview and toolbox. IEEE Geosci. Remote Sens. Mag. **8**(4), 60–88 (2020)
55. Simoes, M., Bioucas-Dias, J., Almeida, L.B., Chanussot, J.: A convex formulation for hyperspectral image superresolution via subspace-based regularization. IEEE Trans. Geosci. Remote Sens. **53**(6), 3373–3388 (2014)
56. Su, Y., Li, J., Plaza, A., Marinoni, A., Gamba, P., Chakravortty, S.: DAEN: Deep autoencoder networks for hyperspectral unmixing. IEEE Trans. Geosci. Remote Sens. **57**(7), 4309–4321 (2019)
57. Sun, R.: Optimization for deep learning: theory and algorithms. arXiv preprint arXiv:1912.08957 (2019)
58. Sussmann, H.J.: Uniqueness of the weights for minimal feedforward nets with a given input-output map. Neural Netw. **5**(4), 589–593 (1992). https://doi.org/10.1016/S0893-6080(05)80037-1. http://www.sciencedirect.com/science/article/pii/S0893608005800371
59. Thouvenin, P.A., Dobigeon, N., Tourneret, J.Y.: Hyperspectral unmixing with spectral variability using a perturbed linear mixing model. IEEE Trans. Signal Process. **64**(2), 525–538 (2015)
60. Tropp, J.A.: User-friendly tail bounds for sums of random matrices. Found. Comput. Math. **12**(4), 389–434 (2012). https://doi.org/10.1007/s10208-011-9099-z
61. Vershynin, R.: Memory capacity of neural networks with threshold and rectified linear unit activations. SIAM J. Math. Data Sci. **2**(4), 1004–1033 (2020). https://doi.org/10.1137/20M1314884
62. Vlačić, V., Bölcskei, H.: Affine symmetries and neural network identifiability. Adv. Math. **376**, 107485 (2020)
63. Wei, Q., Bioucas-Dias, J., Dobigeon, N., Tourneret, J.Y.: Hyperspectral and Multispectral Image Fusion Based on a Sparse Representation. IEEE Trans. Geosci. Remote Sens. **53**(7), 3658–3668 (2015). https://doi.org/10.1109/TGRS.2014.2381272

64. Wei, Q., Bioucas-Dias, J., Dobigeon, N., Tourneret, J.Y.: Hyperspectral and multispectral image fusion based on a sparse representation. IEEE Trans. Geosci. Remote Sens. **53**(7), 3658–3668 (2015)
65. Wei, Q., Dobigeon, N., Tourneret, J.Y.: Fast fusion of multi-band images based on solving a sylvester equation. IEEE Trans. Image Process. **24**(11), 4109–4121 (2015)
66. Weidmann, C., Vetterli, M.: Rate distortion behavior of sparse sources. IEEE Trans. Inf. Theory **58**(8), 4969–4992 (2012). https://doi.org/10.1109/TIT.2012.2201335
67. Xie, Q., Zhou, M., Zhao, Q., Meng, D., Zuo, W., Xu, Z.: Multispectral and hyperspectral image fusion by ms/hs fusion net. In: Proceedings of the CVPR, pp. 1585–1594 (2019)
68. Yao, J., Cao, X., Zhao, Q., Meng, D., Xu, Z.: Robust subspace clustering via penalized mixture of gaussians. Neurocomputing **278**, 4–11 (2018)
69. Yao, J., Hong, D., Chanussot, J., Meng, D., Zhu, X., Xu, Z.: Cross-attention in coupled unmixing nets for unsupervised hyperspectral super-resolution. In: Proceedings of the ECCV, pp. 208–224. Springer, Berlin (2020)
70. Yokoya, N., Grohnfeldt, C., Chanussot, J.: Hyperspectral and Multispectral Data Fusion: A comparative review of the recent literature. IEEE Geosci. Remote Sens. Mag. **5**(2), 29–56 (2017). https://doi.org/10.1109/MGRS.2016.2637824
71. Yokoya, N., Yairi, T., Iwasaki, A.: Coupled nonnegative matrix factorization unmixing for hyperspectral and multispectral data fusion. IEEE Trans. Geosci. Remote Sens. **50**(2), 528–537 (2011)
72. Yun, C., Sra, S., Jadbabaie, A.: Small ReLU networks are powerful memorizers: a tight analysis of memorization capacity. In: Advances in Neural Information Processing Systems, vol. 32, pp. 15558–15569. Curran Associates, Inc., Red Hook (2019)
73. Zhang, C., Bengio, S., Hardt, M., Recht, B., Vinyals, O.: Understanding deep learning requires rethinking generalization. In: International Conference on Learning Representations (2017)
74. Zhu, X.X., Grohnfeldt, C., Bamler, R.: Exploiting Joint Sparsity for Pansharpening: The J-SparseFI Algorithm. IEEE Trans. Geosci. Remote Sens. **54**(5), 2664–2681 (2016). https://doi.org/10.1109/TGRS.2015.2504261

# Chapter 15
# Sparse Recovery of Sound Fields Using Measurements from Moving Microphones

**Fabrice Katzberg and Alfred Mertins**

## 15.1 Problem Formulation and Signal Model

Consider a stationary sound source inside a closed room, emitting the sound pressure signal $s(t)$ with $t \in \mathbb{R}$. Sound-reflecting walls and obstacles lead to a reverberant listening environment and, thus, to a sensed signal $p(t)$ that contains a running superposition of successively delayed and attenuated direct sounds, early-reflection peaks, and diffuse late-field reverberations with exponential decay. For a specific listening position in space, the time-dependent sequence of received sound pressure is characterized by the room impulse response (RIR) $h(t)$. From the physical point, $h(t)$ represents a solution to the acoustic wave equation at the listening point for Dirac delta excitation at $t = 0$ and appropriate boundary conditions. From the signal-processing point of view, $h(t)$ is the filter describing the acoustic transmission path from the source to the receiver location.

Assuming constant atmospheric conditions, $h(t)$ may be modeled as linear time-invariant (LTI) system, and sound propagation is described in terms of linear convolution according to

$$p(t) = s(t) * h(t) = \int_{-\infty}^{\infty} s(t-\tau)h(\tau)\mathrm{d}\tau. \tag{15.1}$$

By spatially extending the particular listening point $\mathbf{x} = (x, y, z)$ to the volume $\Omega = \mathbb{R}^3$, the sound pressure field with respect to both time $t$ and receiver location $\mathbf{x} \in \Omega$ is given by

F. Katzberg · A. Mertins (✉)
Institute for Signal Processing, University of Lübeck, Lübeck, Germany
e-mail: f.katzberg@uni-luebeck.de; alfred.mertins@uni-luebeck.de



G. Kutyniok et al. (eds.), *Compressed Sensing in Information Processing*, Applied and Numerical Harmonic Analysis, https://doi.org/10.1007/978-3-031-09745-4_15

$$p(\mathbf{x}, t) = \int_{-\infty}^{\infty} s(t - \tau) h(\mathbf{x}, \tau) \mathrm{d}\tau, \tag{15.2}$$

where $h(\mathbf{x}, t)$ is the spatially varying RIR from the source position to the point $\mathbf{x}$. Due to the LTI assumption, perception of multiple acoustic events from different sources is modeled subject to the superposition $p(\mathbf{x}, t) = \sum_q s_q(t) * h_q(\mathbf{x}, t)$, with the index $q$ denoting the signals belonging to the $q$-th sound source.

### *15.1.1 General Problem*

Without loss of generality, the following descriptions consider the term sound field as the spatio-temporal RIR $h(\mathbf{x}, t)$ for a particular configuration of one sound source. The recovery of sound-field information is a common inverse problem that is crucial in many audio applications related to sound-field analysis, auditory scene synthesis, and channel equalization [13, 22, 44]. Measuring the sound field is a sampling problem that can be solved with limitations that basically arise due to a restricted number of spatially varying sampling positions.

Measurement and reconstruction strategies often follow the key idea of representing $h(\mathbf{x}, t)$ in terms of a weighted superposition of basis functions $f_p(\mathbf{x}, t)$,

$$h(\mathbf{x}, t) \approx \sum_p a_p f_p(\mathbf{x}, t), \tag{15.3}$$

where the discrete set of coefficients $a_p$ describing $h(\mathbf{x}, t)$ is referred to as sound-field parameters. These parameters are encoded in spatio-temporal samples of $p(\mathbf{x}, t)$ and may be decoded by solving the corresponding inverse problem. The resulting estimates of $a_p$ allow for sound-field reconstruction according to (15.3).

For a broadband signal with maximum frequency $f_\mathrm{c}$, sampling and reconstruction of $h(\mathbf{x}, t)$ requires capturing sound waves in space with minimal wavelength

$$\lambda = \frac{c_0}{f_\mathrm{c}}, \tag{15.4}$$

where $c_0$ denotes the sound velocity. Thus, regular sampling according to the Nyquist–Shannon sampling theorem demands spatial intervals

$$\Delta_\xi \leq \frac{c_0}{2 f_\mathrm{c}} \quad \forall \xi \in \{x, y, z\}, \tag{15.5}$$

in order to avoid aliasing in space. This involves a huge number of sampling positions. A static array of microphones will most likely never be dense enough to enable measurements without significant problems for very high audio frequencies,

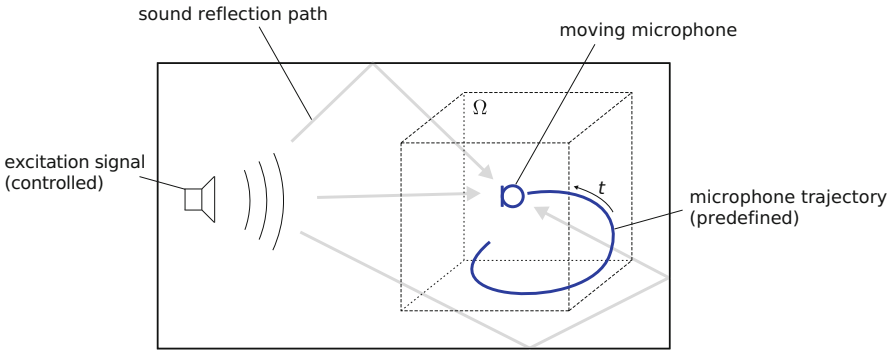



**Fig. 15.1** Dynamic sound-field sampling in an echoic environment with a fixed source configuration

and the effort for exact calibration and spatial positioning would be very high. For example, the sampling of $h(\mathbf{x}, t)$ with $f_\mathrm{c} = 17\,\mathrm{kHz}$ requires microphones at about $10^6$ spatial measuring points inside a volume of $1\,\mathrm{m}^3$. A dynamic approach with only one moving microphone may relax the spatial sampling problem. At this, complete position information is needed, based on either a controlled predefined trajectory or a tracking of the microphone positions. A sketch of the considered dynamic measurement setup is given in Fig. 15.1. As usual, the impact of the microphone on the sound field is considered negligible.

### *15.1.2 Sparse Signal Structures*

Besides sparse signal structure along the temporal dimension for $t < T_\mathrm{m}$, due to discrete reflection paths before a certain mixing time $T_\mathrm{m}$, one major characteristic of $h(\mathbf{x}, t)$ is the inherent connection of its dimensions in frequency domain. For positions sufficiently far away from the sound source and room walls, evanescent sound waves can be ignored, and wave propagation follows the spectral relationship

$$\kappa_x^2 + \kappa_y^2 + \kappa_z^2 = \frac{\omega^2}{c_0^2} \tag{15.6}$$

between the spatial frequencies $\kappa_x$, $\kappa_y$, $\kappa_z$ in $\mathrm{rad\,m^{-1}}$ and the temporal angular frequency $\omega = 2\pi f$ in $\mathrm{rad\,s^{-1}}$. The velocity of the sound waves is $c_0 = \omega/\check{\kappa}$, with the angular wavenumber $\check{\kappa} = |\boldsymbol{\kappa}|$ and the wave vector $\boldsymbol{\kappa} = (\kappa_x, \kappa_y, \kappa_z)$. Since air is regarded as a non-dispersive medium for frequencies within the human hearing range, $c_0$ is independent of $\omega$. Thus, the speed of sound is only a function of atmospheric conditions inside the closed room, e.g., temperature and pressure, which are assumed to be constant according to the LTI model. In consequence,

(15.6) provides a direct connection between the temporal and spatial frequencies of the Fourier transform $H(\boldsymbol{\kappa}, \omega) = \mathcal{F}(h(\mathbf{x}, t))$. If $H(\boldsymbol{\kappa}, \omega)$ is bandlimited in time domain to $\omega_c$, then it is also bandlimited in the spatial domain by $\check{k}_c = \omega_c / c_0$. Moreover, (15.6) reveals that the four-dimensional spectrum $H(\boldsymbol{\kappa}, \omega)$ ideally lives on the three-dimensional surface of a hypercone along the temporal frequency axis $\omega$. Especially at lower temporal frequencies, the conical shape is dominated by a sparse set of frequency combinations. Without the far-field assumption, $H(\boldsymbol{\kappa}, \omega)$ would also be occupied at $\check{\kappa} > \omega_c / c_0$ due to evanescent waves. However, for that case, the energy outside the conical shape decays rapidly along the spatial frequency axes. Detailed analysis of $h(\mathbf{x}, t)$ and the corresponding sampling conditions are given in [2].

The sparse signal structures of $h(\mathbf{x}, t)$ and $H(\boldsymbol{\kappa}, \omega)$, respectively, are excellent prerequisites for compressive sensing and, thus, have been successfully exploited in several methods for obtaining qualified sound-field reconstruction from spatially undersampled stationary measurements [3, 4, 20, 21, 30–32, 45, 50–52]. This chapter recapitulates a recently developed compressed-sensing framework for a dynamic sampling procedure with moving microphones [25, 26]. The dynamic method recovers sparse components of the conically shaped sound-field spectrum. At this, the spectral hypercone is defined in terms of Cartesian coordinates by using regular samples in time, which are directly provided by the microphone, and designed notional grid positions in space. This parameterization in terms of multidimensional regular sampling leads to a highly structured inverse problem and simple mathematical expressions.

The chapter is organized as follows. In Sect. 15.1, the sound-field sampling model is described, and the uniform-grid design is introduced for representing nonequidistant spatial samples subject to a virtual grid in space. The dynamic sampling model and the sparse recovery procedure in frequency domain are presented in Sect. 15.3. Based on a spectrally flat excitation in time and space dimensions, a trajectory-dependent coherence analysis is given in Sect. 15.4. Using the simple expressions from Sect. 15.4, a fast update scheme is specified in Sect. 15.5 that allows for the direct manipulation of trajectory positions, in order to reduce the coherence of the resulting sensing matrix. Finally, a summary is given in Sect. 15.6.

## 15.2 Multidimensional Sampling and Reconstruction

Microphones generate samples at uniform points in time with high acquisition rates. However, considering the spatial dimensions, they are, in general, located at nonequidistant positions unless costly calibrated measurement setups are used. In dynamic setups, spatio-temporal sampling of one moving microphone is sufficient for gathering entire sound-field information. The dynamic microphone performs non-uniform sampling in space at time-varying positions along the measurement trajectory. For simplicity, the underlying spatio-temporal sampling model is described in this section first for uniform sampling in time and non-uniform sampling in space.

The extension to the time-varying component of the dynamic case is carried out in Sect. 15.3.2.

### 15.2.1 Temporal Sampling Model

The temporal bandwidth of observations $p(\mathbf{x}, t)$ is limited with an analog low-pass filter blocking all frequencies above the cutoff frequency $f_\mathrm{c}$. The model parameter $f_\mathrm{c}$ is determined by the considered application.

Let $T = 1/f_\mathrm{s}$ denote the sampling interval of the microphone with temporal sampling frequency $f_\mathrm{s}$ fulfilling the Nyquist–Shannon sampling theorem $f_\mathrm{s} \geq 2f_\mathrm{c}$. This leads to measurements at equidistant sampling points $t_n = nT$, where $n \in \mathbb{Z}$ is the discrete time variable. Supposing that the amplitude of $h(\mathbf{x}, t)$ vanishes into the noise level at $t_n > t_{L-1}$, temporal sampling of the sound pressure field $p(\mathbf{x}, t)$ is modeled by

$$p(\mathbf{x}, n) = \sum_{m=0}^{L-1} s(n-m)h(\mathbf{x}, m) + \eta_1(\mathbf{x}, n), \tag{15.7}$$

where $\eta_1(\mathbf{x}, n)$ models the measurement noise.

### 15.2.2 Spatial Sampling Model

Non-uniform samples can be represented in terms of bandlimited interpolation from notional regular samples at $\mathbf{x}_\# = (x_\#, y_\#, z_\#)$ on a uniform grid spanning $\Omega$. This leads to a parameterization model where the observation of the sound field at receiver location $\mathbf{x}_r = (x_r, y_r, z_r)$ is implicitly described by regular sampling according to

$$p(\mathbf{x}_r, t) = s(t) * h(\mathbf{x}_r, t) = s(t) * \sum_{\mathbf{g}\in\mathbb{Z}^3} h(\mathbf{g}, t) f(\mathcal{G}(\mathbf{x}_r) - \mathbf{g}), \tag{15.8}$$

where discrete grid points $\mathbf{g} = (g_x, g_y, g_z)$ express real-world grid positions in

$$\Omega_\# = \left\{ \mathbf{x}_\# \in \Omega \mid \mathbf{x}_\# = \left(g_x \Delta_x, g_y \Delta_y, g_z \Delta_z\right), \left(g_x, g_y, g_z\right) \in \mathbb{Z}^3 \right\} \tag{15.9}$$

subject to spacings $\Delta_\xi$ in pursuance of (15.5), $\mathcal{G} : \mathbb{R}^3 \to \mathbb{R}^3$ converts from real-world positions to fractional grid coordinates, and $f(\mathcal{G}(\mathbf{x}_r) - \mathbf{g})$ is the multivariate sinc kernel that is separable on the Cartesian grid,

$$f(\mathcal{G}(\mathbf{x}_r) - \mathbf{g}) = \operatorname{sinc}(x_r/\Delta_x - g_x)\operatorname{sinc}(y_r/\Delta_y - g_y)\operatorname{sinc}(z_r/\Delta_z - g_z), \quad (15.10)$$

with

$$\operatorname{sinc}(x) = \frac{\sin(\pi x)}{\pi x}. \quad (15.11)$$

### *15.2.3 Spatio-Temporal Measurement Model*

By merging the sampling models for the particular dimensions, the sampled spatio-temporal sound field can be represented in terms of

$$p(\mathbf{x}_r, n) = \sum_{m=0}^{L-1} s(n-m) \sum_{\mathbf{g}\in\mathbb{Z}^3} h(\mathbf{g}, m) f(\mathcal{G}(\mathbf{x}_r) - \mathbf{g}) + \eta_1(\mathbf{x}_r, n). \quad (15.12)$$

Samples $p(\mathbf{x}_r, n)$ encode the sound-field parameters $h(\mathbf{g}, m)$ that allow for the spatio-temporal reconstruction by analogy with (15.3). The parameters are provided in the form of the signal $h(\mathbf{g}, m)$, which is the uniformly sampled version of the spatially varying RIR at regular grid points $\mathbf{g}$ and discrete delays $m \in \{0, \ldots, L-1\}$. The temporal interval $T$ is explicitly predefined by the sampling rate for the microphone signal. Due to the backward model (15.8) with (15.9), spatial intervals $\Delta_\xi$ are free design parameters of the sampling problem. For $T$ and $\Delta_\xi$ satisfying the Nyquist–Shannon sampling theorem, the recovery of $h(\mathbf{g}, m)$ enables the reconstruction of $h(\mathbf{x}, t)$ by use of appropriate anti-imaging filters.

For the error-free case and an infinite-length model of the involved signals, ideal reconstruction of the continuous sound field $h(\mathbf{x}, t)$ is accomplished by a separable four-dimensional sinc filter with unlimited support. This is not feasible in practice. However, despite truncating the temporal signals to limited taps $m \in \{0, \ldots, L-1\}$, finite-length interpolation filters allow for reasonable approximations in the time dimension due to the exponential energy decay of RIRs for higher-order reflection paths. In the spatial dimensions, a hard limitation of the signals to a bounded observation window is more critical in several aspects.

#### 15.2.3.1 Finite-Length Observations in Space

So far, the notional grid points $\mathbf{x}_\# \in \Omega_\#$ are defined by an unbounded set in $\mathbb{R}^3$. This ideal situation is not realizable in real-world setups. Measurement apertures are always of finite length and, therefore, have a direct filtering effect on the original signals to be measured. In this sense, a signal that is originally sparse in some domain could lose this characteristic when being sampled. This issue is described

in the following, and appropriate solutions are given for the designated sound-field sampling model.

In practice, spatial interpolation is based on a limited number of grid positions $\bar{\Omega}_{\#} \subset \Omega_{\#}$ in a bounded cuboid measurement volume $\bar{\Omega} \subset \Omega$. By defining an odd number of samples in each dimension, $G = G_x G_y G_z$ discrete grid points $\mathbf{g}$ in

$$\Gamma = \left\{-\frac{G_x-1}{2}, \ldots, \frac{G_x-1}{2}\right\} \times \left\{-\frac{G_y-1}{2}, \ldots, \frac{G_y-1}{2}\right\} \times \left\{-\frac{G_z-1}{2}, \ldots, \frac{G_z-1}{2}\right\}$$

span a volume of size $XYZ$ with $X = \Delta_x(G_x - 1)$, $Y = \Delta_y(G_y - 1)$, $Z = \Delta_z(G_z - 1)$. Applying this hard limitation to the spatial sampling model (15.8) initially induces a rectangular windowing of the original sound-field signal according to

$$\bar{h}(\mathbf{x}, t) = h(\mathbf{x}, t)\bar{w}_{3\mathrm{D}}(\mathbf{x}), \tag{15.13}$$

where the multivariate window $\bar{w}_{3\mathrm{D}}(\mathbf{x}) = \bar{w}_X(x)\,\bar{w}_Y(y)\,\bar{w}_Z(z)$ is composed of individually scaled rectangular windows in terms of $\bar{w}_X(x) = \mathrm{rect}(x/X)$. Since

$$\bar{W}_X(\kappa_x) = \int_{-\frac{X}{2}}^{\frac{X}{2}} \mathrm{e}^{-\mathrm{i}\kappa_x x}\mathrm{d}x = X\,\mathrm{sinc}\left(\frac{X\kappa_x}{2\pi}\right), \tag{15.14}$$

the windowing in (15.13) translates in frequency domain to the convolution

$$\begin{aligned}\bar{H}(\boldsymbol{\kappa}, \omega) &= H(\boldsymbol{\kappa}, \omega) * (\bar{W}_{3\mathrm{D}}(\boldsymbol{\kappa})\delta(\omega))\,(2\pi)^{-3} \\ &= \frac{XYZ}{(2\pi)^3}\int_{\mathbb{R}^3} H(\boldsymbol{\kappa} - \mathbf{k}, \omega)\,\mathrm{sinc}\left(\frac{Xk_x}{2\pi}\right)\mathrm{sinc}\left(\frac{Yk_y}{2\pi}\right)\mathrm{sinc}\left(\frac{Zk_z}{2\pi}\right)\mathrm{d}\mathbf{k}.\end{aligned} \tag{15.15}$$

This influence of the sampling setup on the original signal $H(\boldsymbol{\kappa}, \omega)$, i.e., blurring of spatial frequencies with a sinc filter kernel, is critical owing to multiple reasons. On the one hand, the spatial interpolation of $h(\mathbf{x}, t)$ from the spatially sampled spectrum

$$\bar{H}_{\mathrm{s}}(\boldsymbol{\kappa}, \omega) = \frac{1}{\Delta_x\Delta_y\Delta_z}\sum_{\mathbf{k}\in\mathbb{Z}^3} \bar{H}\left(\left(\kappa_x - \frac{2\pi k_x}{\Delta_x}, \kappa_y - \frac{2\pi k_y}{\Delta_y}, \kappa_z - \frac{2\pi k_z}{\Delta_z}\right), \omega\right) \tag{15.16}$$

is inaccurate, especially due to the fact that spatial anti-aliasing filters are hardly applicable in the spatial dimensions. The reconstruction becomes erroneous, which, for our backward sampling model (15.8), in turn induces perturbations for both the sampling itself and the actual reconstruction. On the other hand, the filter $\bar{W}_{3\mathrm{D}}(\boldsymbol{\kappa})$ destroys the conical shape of $H(\boldsymbol{\kappa}, \omega)$ and reduces spectral sparsity substantially. Frequency allocation is impaired by a cross-shaped blurring due to the separated three-dimensional sinc kernel. In order to preserve the conical structure and the sparse frequency localization, it would be desirable to have a spatial observation window that leads to a radially shaped, fast decaying filter that maintains compact

support of $\bar{H}(\boldsymbol{\kappa}, \omega)$ on the spherical shell

$$\sqrt{\kappa_x^2 + \kappa_y^2 + \kappa_z^2} \pm \epsilon = \frac{|\omega|}{c_0}, \tag{15.17}$$

with $\epsilon \in \mathbb{R}_+$ being as small as possible.

To improve spatial reconstruction from finite samples, (15.15) and (15.16) indicate that either the size $V = XYZ$ of the observation window can be chosen larger, possibly larger than an actual volume of interest $\bar{\Omega}_\mathrm{i} \subset \bar{\Omega}$, or, alternatively, the spatial sampling intervals $\Delta_\xi$ can be chosen smaller, well above the Nyquist rate. For the direct sampling of positions $\mathbf{x}_\# \in \bar{\Omega}_\#$, both procedures enlarge the measurement effort in hardware and/or sampling time. For the inverse spatial sampling problem (15.8), $V$ and $\Delta_\xi$ are free design parameters and can be arbitrarily adjusted in a tradeoff with the number of variables to be recovered from linear measurement equations. Considering a compressed-sensing-based recovery strategy, such an increase of variables can be irrelevant or even beneficial as far as sparsity in the sparse signal representation is maintained or even accentuated.

Based on a single sample at position $\mathbf{x}_r \in \bar{\Omega}$, the reverse sampling model (15.8) allows for the incorporation of a predefined spatial observation window as

$$p(\mathbf{x}_r, t) = s(t) * \sum_{\mathbf{g} \in \Gamma} h(\mathbf{g}, t) w_{3\mathrm{D}}(\mathbf{g}) w_{3\mathrm{D}}^{-1}(\mathbf{g}) \varphi_{3\mathrm{D}}(\mathcal{G}(\mathbf{x}_r) - \mathbf{g}) + \eta_2(\mathbf{x}_r, t), \tag{15.18}$$

where the signal $h(\mathbf{g}, t)$ finally loses its ideally bandlimited character in the space dimension, $\varphi_{3\mathrm{D}}(\mathcal{G}(\mathbf{x}_r) - \mathbf{g})$ is a three-dimensional finite-length sinc filter approximation, $\eta_2(\mathbf{x}_r, t)$ comprises the errors due to spatial sampling and interpolation, respectively, and $w_{3\mathrm{D}}(\mathbf{g})$ is a sampled window function that should be designed properly, in particular, subject to the demand for preserving sparsity and conical shape of the sound-field spectrum. Since (15.18) poses an inverse problem, the coefficients should be $w_{3\mathrm{D}}(\mathbf{g}) \neq 0 \, \forall \mathbf{g} \in \Gamma$. A good choice, for example, is the three-dimensional window consisting of a Hamming window in each dimension conformable to

$$w_{G_x}(g_x) = 0.54 - 0.46 \cos\left( \frac{2\pi \left( g_x - \frac{G_x - 1}{2} \right)}{G_x - 1} \right). \tag{15.19}$$

Regarding the continuous signal, this selection suggests an observation window of length $X = \Delta_x (G_x - 1)$ composed of

$$w_X(x) = \left( 0.54 + 0.46 \cos\left( 2\pi \frac{x}{X} \right) \right) \mathrm{rect}\left( \frac{x}{X} \right), \tag{15.20}$$

having the Fourier transform

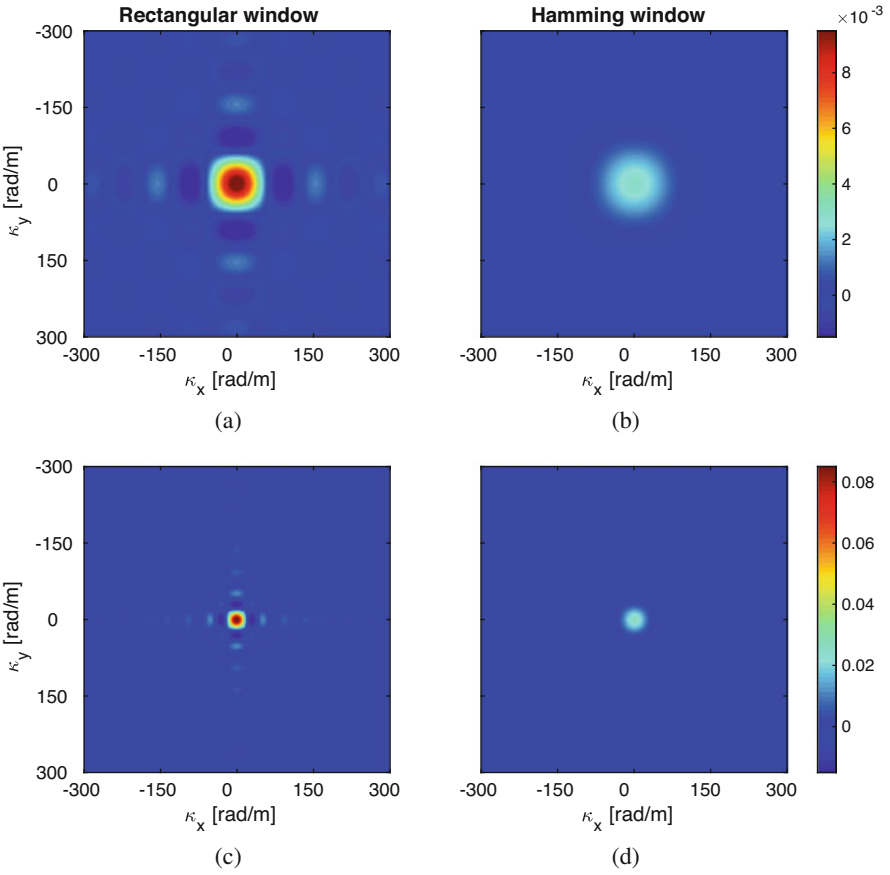



**Fig. 15.2** Two-dimensional spectral filters for a rectangular observation window (left) and a Hamming window (right) considering sizes (**a**), (**b**) $X = Y = 0.1$ m and (**c**), (**d**) $X = Y = 0.3$ m

$$W_X(\kappa_x) = 0.23X \operatorname{sinc}\left(1 - \frac{X\kappa_x}{2\pi}\right) + 0.23X \operatorname{sinc}\left(1 + \frac{X\kappa_x}{2\pi}\right) + 0.54X \operatorname{sinc}\left(\frac{X\kappa_x}{2\pi}\right). \tag{15.21}$$

In contrast to (15.14), (15.21) constructs a filter $W_{3D}(\boldsymbol{\kappa}) = W_X(\kappa_x) W_Y(\kappa_y) W_Z(\kappa_z)$ having nearly omnidirectional directivity pattern. The three-dimensional filter is more compact and leads to a spectrum $\bar{H}(\kappa, \omega) = H(\kappa, \omega) * (W_{3D}(\boldsymbol{\kappa})\delta(\omega))(2\pi)^{-3}$ that preserves both the sparse frequency localization and the conical structure according to (15.17). A comparison of both filter types is visualized in Fig. 15.2 for the two-dimensional case.

#### 15.2.3.2 Linear Equations for Parameter Recovery in Terms of Uniform Grids

By choosing appropriate design parameters, the spatio-temporal sampling inside $\bar{\Omega}_\mathrm{i} \subset \bar{\Omega}$ is finally modeled by linear measurement equations

$$\begin{aligned} p(\mathbf{x}_r, n) &= \sum_{m=0}^{L-1} \sum_{\mathbf{g}\in\Gamma} s(n-m) w_{\mathrm{3D}}^{-1}(\mathbf{g}) \varphi_{\mathrm{3D}}(\mathcal{G}(\mathbf{x}_r) - \mathbf{g})\, h(\mathbf{g}, m) w_{\mathrm{3D}}(\mathbf{g}) + \eta(\mathbf{x}_r, n) \\ &= \sum_{m=0}^{L-1} \sum_{\mathbf{g}\in\Gamma} \bar{s}_{\mathbf{x}_r}(\mathbf{g}, n-m)\, \bar{h}(\mathbf{g}, m) + \eta(\mathbf{x}_r, n), \end{aligned} \tag{15.22}$$

where

$$\bar{s}_{\mathbf{x}_r}(\mathbf{g}, m) = s(m) w_{\mathrm{3D}}^{-1}(\mathbf{g}) \varphi_{\mathrm{3D}}(\mathcal{G}(\mathbf{x}_r) - \mathbf{g}) \tag{15.23}$$

can be regarded as the sampled spatio-temporal excitation on the multidimensional grid, $\bar{h}(\mathbf{g}, m)$ is the corresponding sampled spatio-temporal RIR, and $\eta(\mathbf{x}_r, n)$ is a perturbation term pooling measurement noise as well as any systematic and model-induced errors. The knowledge about the temporal excitation sequence $s(n)$ and controlled or tracked microphone positions $\mathbf{x}_r$ allows for setting up the system of linear equations

$$\mathbf{p} = \mathbf{A}\bar{\mathbf{h}} + \boldsymbol{\eta} \tag{15.24}$$

that describes the spatio-temporal sampling subject to joint variables in $\bar{\mathbf{h}}$. The measurement vector $\mathbf{p} \in \mathbb{R}^{MR}$ contains the stack of $R$ microphone signals each of length $M$, $\mathbf{A} \in \mathbb{R}^{MR\times P}$ is the sampling matrix, $\bar{\mathbf{h}} \in \mathbb{R}^{P}$ encapsulates $P = LG$ windowed sound-field parameters, and $\boldsymbol{\eta} \in \mathbb{R}^{M}$ is the perturbation vector. The inverse problem (15.24) enables the calculation of estimates $\hat{h}(\mathbf{g}, n) = \hat{\bar{h}}(\mathbf{g}, n) w_{\mathrm{3D}}^{-1}(\mathbf{g})$ that may be used for spatial reconstruction inside $\bar{\Omega}_\mathrm{i}$ according to the interpolation

$$h(\mathbf{x}, n) \approx \sum_{\mathbf{g}\in\Gamma} \hat{h}(\mathbf{g}, n) \varphi_{\mathrm{3D}}(\mathcal{G}(\mathbf{x}) - \mathbf{g}). \tag{15.25}$$

For the case, where (15.24) is solved directly without exploiting sparsity in frequency domain, the simple choice $w_{\mathrm{3D}}(\mathbf{g}) = 1$ is convenient. The finally estimated sound-field parameters $\hat{h}(\mathbf{g}, n)$ describe the unweighted spatio-temporal RIR, uniformly sampled at $L$ delays in time and at $G = G_x G_y G_z$ grid positions in multidimensional space.

Other parameterization models are, of course, also possible for describing spatio-temporal samples, e.g., using quincunx [2] or spherical patterns [40] instead of

the uniform-grid model. For the direct measurement of such patterns using a correspondingly shaped array, quincunx and spherical configurations may reduce the number of sampling positions needed for adequate interpolation inside a three-dimensional target region $\bar{\Omega}_\mathrm{i}$. Especially, the demand for a spatial observation window that preferably maintains the compact support of the spectrum within a narrow spherical shell according to (15.17), in fact, leads to a parameterization model in terms of spherical coordinates [28] where spherical harmonics for the angular part and spherical Bessel functions for the radial part are used (assuming a source-free volume $\bar{\Omega}_\mathrm{i}$). However, this may lead to ill-posed problems due to several frequency-radius combinations where the Bessel functions cross zero [33, 39, 40]. Furthermore, for spatial sampling scenarios without costly calibrated arrays and arbitrary positions $\mathbf{x}_r \in \bar{\Omega}_\mathrm{i}$, spherical and also quincunx patterns that parameterize $h(\mathbf{x}, t)$ result in inverse problems that are not straightforward in modeling and analysis, especially for the extension to the more complicated case of a moving microphone to be introduced in Sect. 15.3. For quincunx sampling, non-trivial interpolation filters that are not separable in the spatial dimensions must be calculated, clearly raising the complexity in the design of the measurement matrix.

Given the speed of sound $c_0$, the uniform-grid parameterization requires a spacing of the notional grid that satisfies $\Delta_\xi = \alpha^{-1} c_0 (2 f_\mathrm{c})^{-1}$ with oversampling factor $\alpha \geq 1$ for each dimension $\xi \in \{x, y, z\}$, in order to avoid spatial aliasing and allow for a sound-field reconstruction by means of (15.25). This model results in a highly structured sampling problem, also for the dynamic case, and allows for straightforward error analyses and extensions [25, 26]. The separable low-pass filter

$$\varphi_{3\mathrm{D}}(\mathcal{G}(\mathbf{x}) - \mathbf{g}) = \varphi(x/\Delta_x - g_x)\, \varphi(y/\Delta_y - g_y)\, \varphi(z/\Delta_z - g_z) \tag{15.26}$$

is easy to calculate and leads to a well-arranged block composition of $\mathbf{A}$. Based on this, simple expressions enable an efficient evaluation of the expected estimation error as a function of spatial sampling positions or, for the dynamic case, of the microphone trajectory. The measurement matrix $\mathbf{A}$ only contains the source signal and spatial interpolation coefficients that depend on the microphone trajectory relative to the modeled grid. The effort for setting up and solving the system is extremely low. The separability of dimensions also enables the efficient implementation of sparsifying transforms for the compressed-sensing framework presented in Sect. 15.3.

In its original form, the uniform-grid model comprises

$$P_{\Omega_\mathrm{c}}(V, \omega_l) = \left\lceil \alpha \, \frac{\omega_l \sqrt[D]{V}}{\pi c_0} + 1 \right\rceil^D \tag{15.27}$$

parameters to be estimated for recovering temporal frequency $\omega_l$ inside a $D$-dimensional cubical region $\Omega_\mathrm{c}$ of size $V$. Even if the sound propagation is actually three-dimensional, parameter recovery is possible along a line ($D = 1$), on a plane ($D = 2$), or within a volume ($D = 3$). As can be seen from (15.27), a reduction

of the dimensions in the target volume is directly transferred to a reduction of the dimensions of the uniform grid to be recovered. In frequency domain, these parameter numbers are substantially reduced as there are sparse coefficients ideally located along the conical surface (15.6). However, reducing spatial dimensions of the observation inherently diminishes spectral structure according to $\kappa_x^2 + \kappa_y^2 \leq \omega^2/c_0^2$ (planar case) and $\kappa_x^2 \leq \omega^2/c_0^2$ (linear case) due to the released spatial variables in the three-dimensional propagation medium.

## 15.3 Sparse Sound-Field Recovery in Frequency Domain

The spatio-temporal sampling model (15.24) can be directly extended to a sparse recovery problem in discrete Fourier domain. For the special case of stationary sound-field sampling, the resulting sensing matrix possesses a well-known structure that immediately allows us to apply the existing theory and methods for sparse recovery. This interesting link will be provided in Sect. 15.3.1. Subsequently, in Sect. 15.3.2, the spatio-temporal sampling model is first extended to the dynamic measurement procedure with moving microphones. Then, in Sect. 15.3.3, the corresponding formulation of the sparse recovery problem along the spectral hypercone is given. Finally, Sect. 15.3.4 shows how deterministic source sequences can be used in order to reduce complexity for setting up and solving the dynamic sampling problem, and Sect. 15.3.5 presents an efficient recovery algorithm that directly exploits the block structure of the dynamic-sensing matrix.

### *15.3.1 Basic Ideas for the Simplified Stationary Case*

As a starting point, let us take the linear system model (15.24) for describing a stationary setup with $R$ microphones at $R$ positions $\mathbf{x}_r \in \bar{\Omega}_\mathrm{i}$. Then, assuming no a priori knowledge about spectral structures, sparse sound-field recovery in frequency domain may be accomplished by solving the problem

$$\underset{\mathbf{c}\in\mathbb{C}^P}{\operatorname{argmin}} \ \|\mathbf{p} - \mathbf{B}_\mathrm{stat}\,\mathbf{c}\|_2^2 \quad \text{s.t.} \quad \|\mathbf{c}\|_0 \leq K, \tag{15.28}$$

with the sensing matrix $\mathbf{B}_\mathrm{stat} = \mathbf{A}\boldsymbol{\Psi}^{-1}$, sparse frequency parameters in $\mathbf{c} = \boldsymbol{\Psi}\bar{\mathbf{h}}$, where the support in $\mathbf{c}$ is quantified as $\|\mathbf{c}\|_0 = |\{i : c_i \neq 0\}|$, and the sparsifying transformation matrix $\boldsymbol{\Psi} \in \mathbb{C}^{P\times P}$ that performs the four-dimensional discrete Fourier transform (DFT) on the vectorized sound-field signal in $\mathbf{h}$. The problem (15.28) is NP-hard [34] and is typically solved using either a relaxation into an $\ell_1$-minimization problem allowing for convex optimization, such as basis pursuit [12, 14], LASSO [46], or Dantzig selector [11], or a greedy algorithm such as the orthogonal matching pursuit (OMP) [48], compressed-sensing matching pursuit

(CoSaMP) [35], or iterative hard thresholding (IHT) [5, 6]. Based on sparse estimates in $\hat{\mathbf{c}}$, the unweighted sound-field parameters are obtained as

$$\hat{\mathbf{h}} = (\mathbf{W}^{-1} \otimes \mathbf{I}_L)\boldsymbol{\Psi}^{-1}\hat{\mathbf{c}}, \tag{15.29}$$

where $\otimes$ denotes the Kronecker product, the diagonal matrix $\mathbf{W} = \mathrm{diag}\{\mathbf{w}\}$ encapsulates the weights $\mathbf{w} = [w_{3\mathrm{D}}(\mathbf{g}_1), \ldots, w_{3\mathrm{D}}(\mathbf{g}_G)]$ from the chosen window function, and $\mathbf{I}_L$ is the $L \times L$ identity matrix.

In order to guarantee stable and robust recovery of any (approximately) $K$-sparse signal, any arbitrary set of $K$ columns of the sensing matrix $\mathbf{B}$ must build up a nearly orthogonal system, which is formalized by the so-called restricted isometry property (RIP) [10]. Verifying the RIP of a matrix is a combinatorial NP-hard problem [47]; however, for the special case of stationary sound-field sampling according to (15.28), the multidimensional sampling problem can be reduced to single one-dimensional recovery problems that have been well studied in the compressed-sensing literature. On the one hand, there is the sampling and deconvolution problem in the temporal domain, which is trivial for a spectrally flat excitation sequence, especially in Fourier domain, and, anyway, is not the critical part in (15.28) since the microphones do not vary positions (see below in (15.33)). On the other hand, there is the interpolation problem in the spatial dimensions. In frequency domain, assuming spectrally flat interpolation filters (see Sect. 15.4.3), this problem translates to nonequidistant sampling of trigonometric polynomials and leads to a specifically structured sensing matrix, also for multivariate extensions. Thus, sound-field sampling at random positions in space corresponds to the random sampling of trigonometric polynomials. Related probabilistic guarantees for the sensing matrix and the sparse recovery have been investigated in [41, 42].

For a non-random distribution of microphone positions, the coherence property

$$\mu(\mathbf{B}) = \max_{1 \le u \ne v \le P} \frac{|\langle \mathbf{b}_u, \mathbf{b}_v \rangle|}{\|\mathbf{b}_u\|_2 \, \|\mathbf{b}_v\|_2}, \tag{15.30}$$

may be used in practice as an indicator for RIP guarantees, where $\mathbf{b}_u$ denotes the $u$-th column of $\mathbf{B}$ [17]. The coherence of $\mathbf{B}$ directly affects the upper bound of its RIP constant that, in turn, determines upper bounds for recovery errors induced by measurement noise, the $K$-sparse signal approximation, and a mismatch of $\mathbf{B}$ (e.g., due to inaccurate calibration of microphone positions) [8, 10, 15, 16, 23].

The stationary case (15.28) can be seamlessly incorporated into the existing multidimensional compressed-sensing frameworks that are known, for example, from distributed sensing problems, and exploit the structures of Kronecker product-based measurement and sparsifying matrices for efficient calculations [9, 18]. Let us briefly outline the corresponding relationship in the following.

Suppose that the sound-field parameters in $\mathbf{h}$ are concatenated first along the time dimension and then along the $x$, $y$, and $z$ dimensions in succession. By defining $\mathbf{F}_U \in \mathbb{C}^{U \times U}$ to perform the normalized DFT with $\mathbf{F}\mathbf{F}^H = \mathbf{I}_U$, the sensing matrix for the considered stationary sampling problem can be represented as

$$\begin{aligned}\mathbf{B}_{\text{stat}} &= \mathbf{A}\boldsymbol{\Psi}^H \\ &= \left(\boldsymbol{\Phi}\mathbf{W}^{-1} \otimes \mathbf{S}\right)\left(\mathbf{F}_{G_z}^H \otimes \mathbf{F}_{G_y}^H \otimes \mathbf{F}_{G_x}^H \otimes \mathbf{F}_L^H\right) \\ &= \boldsymbol{\Phi}\mathbf{W}^{-1}\boldsymbol{\Psi}_{\text{3D}}^H \otimes \mathbf{S}\mathbf{F}_L^H, \end{aligned} \tag{15.31}$$

where $\mathbf{S} \in \mathbb{R}^{M\times L}$ is the convolution matrix of the excitation, $\boldsymbol{\Phi} = [\boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_R]^T$ with $\boldsymbol{\varphi}_r = [\varphi_{\text{3D}}(\mathcal{G}(\mathbf{x}_r) - \mathbf{g}_1), \ldots, \varphi_{\text{3D}}(\mathcal{G}(\mathbf{x}_r) - \mathbf{g}_G)]^T$ comprises the interpolation coefficients corresponding to the $G$ grid positions with respect to the $r$-th microphone position, and $\boldsymbol{\Psi}_{\text{3D}} = \mathbf{F}_{G_z} \otimes \mathbf{F}_{G_y} \otimes \mathbf{F}_{G_x}$ performs the multivariate DFT along the spatial dimensions. The Kronecker-based expressions in (15.31) can be used to apply efficient Kronecker-based recovery algorithms [9] and achieve efficient coherence calculations. For example, by using the formula

$$\mu(\mathbf{C}_1 \otimes \mathbf{C}_2) = \max\{\mu(\mathbf{C}_1), \mu(\mathbf{C}_2)\} \tag{15.32}$$

involving the matrices $\mathbf{C}_1, \mathbf{C}_2$ with normalized columns (proof in [24]), it can be seen that, for a spectrally flat excitation sequence, the coherence of $\mathbf{B}_{\text{stat}}$ is determined by the spatial components only, according to

$$\mu(\mathbf{B}_{\text{stat}}) = \mu(\boldsymbol{\Phi}\mathbf{W}^{-1}\boldsymbol{\Psi}_{\text{3D}}^H), \tag{15.33}$$

with $\mathbf{W}^{-1}$ determined by the arbitrary window design, and $\boldsymbol{\Phi}$ being directly dependent on the selection of static microphone positions.

### 15.3.2 From Static to Dynamic Sensing

Consider the linear measurement equations (15.22) with an additional time-varying relationship of the spatial measurement positions. Instead of capturing field information over constant positions $\mathbf{x}_r$ at each sampling point $n$, data is now acquired at $R$ locations $\mathbf{x}_r(n)$ changing over time. By defining $\mathbf{x}_r : \mathbb{Z} \to \mathbb{R}^3$ as the trajectories performed by $R$ moving microphones inside the target volume $\bar{\Omega}_{\text{i}}$, we obtain a dynamic measurement setup that generates samples at uniform points in time and, generally, at varying and non-uniform positions in space. In this case, only one microphone ($R = 1$) moving along the sampled trajectory $\mathbf{x}(n) = (x(n), y(n), z(n))$ and measuring the signal $p(\mathbf{x}(n), n) = s(n) * h(\mathbf{x}, n)|_{\mathbf{x}=\mathbf{x}(n)} + \eta(\mathbf{x}(n), n)$ is sufficient for gathering the entire sound-field information. This setup is considered in the following descriptions. An outline of the involved signals and parameters is given in Fig. 15.3. The expansion to $R > 1$ dynamic microphones is straightforward and may reduce the acquisition time.

According to (15.22), spatio-temporal samples of the moving microphone can be described as

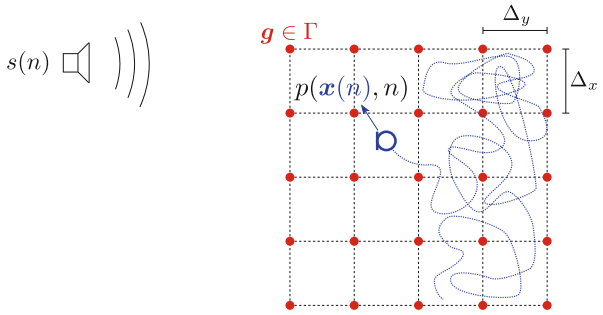



**Fig. 15.3** Dynamic sampling principle. The design variables $\Delta_x$ and $\Delta_y$ model a notional grid in space (red dots). The microphone samples $p(\mathbf{x}(n), n)$ at trajectory positions $\mathbf{x}(n)$ can be represented in terms of the unknown sound field at that grid

$$p(\mathbf{x}(n), n) = \sum_{m=0}^{L-1} \sum_{\mathbf{g} \in \Gamma} s(n-m) w_{\text{3D}}^{-1}(\mathbf{g}) \varphi_{\text{3D}}(\mathcal{G}(\mathbf{x}(n)) - \mathbf{g})\, \bar{h}(\mathbf{g}, m) + \eta(\mathbf{x}(n), n). \tag{15.34}$$

Due to the LTI assumption, single dynamic samples $p(\mathbf{x}(n), n)$ share the same sound-field parameters inside $\bar{\Omega}$ and, by knowing the trajectory of the moving microphone, can be jointly represented by

$$\mathbf{p} = \sum_{\mathbf{g} \in \Gamma} w_{\mathbf{g}}^{-1} \mathbf{D}_{\mathbf{g}} \mathbf{S}\, \bar{\mathbf{h}}_{\mathbf{g}} + \boldsymbol{\eta}, \tag{15.35}$$

where the vector $\mathbf{p} \in \mathbb{R}^M$ collects the $M$ dynamic samples along the trajectory,

$$\mathbf{p} = [p(\mathbf{x}(0), 0), \ldots, p(\mathbf{x}(M-1), M-1)]^T, \tag{15.36}$$

$\boldsymbol{\eta} \in \mathbb{R}^M$ is the perturbation vector,

$$\boldsymbol{\eta} = [\boldsymbol{\eta}(\mathbf{x}(0), 0), \ldots, \boldsymbol{\eta}(\mathbf{x}(M-1), M-1)]^T, \tag{15.37}$$

$\bar{\mathbf{h}}_{\mathbf{g}} \in \mathbb{R}^L$ contains the spatially windowed RIR at the notional grid point $\mathbf{g}$,

$$\bar{\mathbf{h}}_{\mathbf{g}} = \left[\bar{h}(\mathbf{g}, 0), \ldots, \bar{h}(\mathbf{g}, L-1)\right]^T, \tag{15.38}$$

$\mathbf{D}_{\mathbf{g}} \in \mathbb{R}^{M \times M}$ is a diagonal matrix stacking all $M$ interpolation coefficients acquired during dynamic sampling for the grid RIR at point $\mathbf{g}$,

$$\mathbf{D}_{\mathbf{g}} = \text{diag}\left\{[\varphi(\mathcal{G}(\mathbf{x}(0)) - \mathbf{g}), \ldots, \varphi(\mathcal{G}(\mathbf{x}(M-1)) - \mathbf{g})]\right\}, \tag{15.39}$$

$w_{\mathbf{g}} = w_{\text{3D}}(\mathbf{g})$ denotes the scalar weighting at position $\mathbf{g}$ according to the designed observation window, and $\mathbf{S} \in \mathbb{R}^{M \times L}$ is the convolution matrix of the source signal $s(n)$ constructed by $\mathbf{S} = [\mathbf{s}(0), \ldots, \mathbf{s}(M-1)]^T$ with

$$\mathbf{s}(n) = [s(n), s(n-1), \ldots, s(n-L+1)]^T . \tag{15.40}$$

This representation of dynamic samples leads to the system of linear equations

$$\mathbf{p} = \mathbf{A}_{\text{dyn}}\,\bar{\mathbf{h}} + \boldsymbol{\eta}, \tag{15.41}$$

where the vector $\bar{\mathbf{h}} \in \mathbb{R}^{LG}$ is the concatenation of weighted grid RIRs,

$$\bar{\mathbf{h}} = \left[\bar{\mathbf{h}}_{\mathbf{g}_1}^T, \bar{\mathbf{h}}_{\mathbf{g}_2}^T, \ldots, \bar{\mathbf{h}}_{\mathbf{g}_G}^T\right]^T , \tag{15.42}$$

and the measurement matrix $\mathbf{A} \in \mathbb{R}^{M \times LG}$ follows the block structure

$$\begin{aligned} \mathbf{A}_{\text{dyn}} &= \left[w_{\mathbf{g}_1}^{-1}\mathbf{D}_{\mathbf{g}_1}\mathbf{S},\ w_{\mathbf{g}_2}^{-1}\mathbf{D}_{\mathbf{g}_2}\mathbf{S},\ \ldots,\ w_{\mathbf{g}_G}^{-1}\mathbf{D}_{\mathbf{g}_G}\mathbf{S}\right] \\ &= \left[\mathbf{D}_{\mathbf{g}_1}\mathbf{S},\ \mathbf{D}_{\mathbf{g}_2}\mathbf{S},\ \ldots,\ \mathbf{D}_{\mathbf{g}_G}\mathbf{S}\right]\left(\mathbf{W}^{-1} \otimes \mathbf{I}_L\right) \end{aligned} \tag{15.43}$$

that consists of $G$ repetitions of $\mathbf{S}$ along to columns in order to model temporal excitation at each grid point in space. In contrast to the stationary case, the Toeplitz structure of the convolution matrices is distorted by the diagonal matrices $\mathbf{D}_{\mathbf{g}}$ that scale the rows of $\mathbf{S}$ differently according to the instantaneous variations of the dynamic microphone position $\mathbf{x}(n)$. Dynamic samples are represented in an indivisible spatio-temporal sense.

### *15.3.3 Sparse Recovery Along the Spectral Hypercone*

The adaptation of the dynamic sampling problem to a sparse recovery strategy in frequency domain is straightforward. Also, the reduction of the search space to the a priori known hyperconical structure along the temporal frequency axis is simple due to the uniform-grid parameterization model.

Let us define the discrete spatial frequency variables $\mathbf{k} = (k_x, k_y, k_z)$ with

$$k_x \in \left\{-\tfrac{G_x-1}{2}, \ldots, \tfrac{G_x-1}{2}\right\}, k_y \in \left\{-\tfrac{G_y-1}{2}, \ldots, \tfrac{G_y-1}{2}\right\}, k_z \in \left\{-\tfrac{G_z-1}{2}, \ldots, \tfrac{G_z-1}{2}\right\}, \tag{15.44}$$

and the discrete temporal frequency variable $l \in \left\{-\frac{L-1}{2}, \ldots, \frac{L-1}{2}\right\}$ ($L$ is set odd), translating to sampled frequencies according to

$$\kappa_{k_x} = 2\pi\,\Delta_x^{-1}\frac{k_x}{G_x}, \quad \kappa_{k_y} = 2\pi\,\Delta_y^{-1}\frac{k_y}{G_y}, \quad \kappa_{k_z} = 2\pi\,\Delta_z^{-1}\frac{k_z}{G_z}, \tag{15.45}$$

and $\omega_l = 2\pi f_{\text{s}}\frac{l}{L}$. By using the relationship (15.17) and considering an appropriate cubical observation window design (see Sect. 15.2.3.1) with $\sqrt[3]{G} = G_x = G_y = G_z$

and $\Delta = \Delta_x = \Delta_y = \Delta_z$, the support of sparse DFT coefficients in $\mathbf{c} = \boldsymbol{\Psi}\bar{\mathbf{h}}$ can be constrained to the $C < P$ discrete frequency locations $(\mathbf{k}, l)$, where the spatial frequencies constitute spherical shells along the temporal frequencies $l$ subject to

$$\sqrt{k_x^2 + k_y^2 + k_z^2} = \alpha_\mathrm{p} \, |l| \, \frac{\sqrt[3]{G}}{L} \pm \epsilon_\mathrm{s}, \tag{15.46}$$

with the proportioning factor $\alpha_\mathrm{p} = \Delta f_\mathrm{s}/c_0$ and a small shell margin $\epsilon_\mathrm{s} \in \mathbb{R}_+$. The a priori knowledge (15.46) allows for constructing the assigning matrix $\mathbf{R} \in \{0, 1\}^{C \times P}$ that consists of $C$ unit vectors in the rows selecting the corresponding frequency elements in $\mathbf{c}$ according to $\mathring{\mathbf{c}} = \mathbf{R}\mathbf{c}$. Thus, the dynamic sampling problem in terms of spectral sampling along the conically shaped subspace of dimension $C$ reads

$$\mathbf{p} = \mathbf{A}_\mathrm{dyn} \boldsymbol{\Psi}^H \mathbf{R}^T \mathbf{R} \boldsymbol{\Psi} \bar{\mathbf{h}} + \boldsymbol{\eta} \tag{15.47}$$

$$= \mathring{\mathbf{B}}_\mathrm{dyn} \, \mathring{\mathbf{c}} + \boldsymbol{\eta}, \tag{15.48}$$

where $\mathbf{R}^T$, in turn, shrinks $\mathbf{B}_\mathrm{dyn} = \mathbf{A}_\mathrm{dyn} \boldsymbol{\Psi}^H$ to the selected columns $\mathring{\mathbf{B}}_\mathrm{dyn} = \mathbf{B}_\mathrm{dyn} \mathbf{R}^T$ that contribute to discrete frequencies satisfying (15.46). Finally, we obtain the recovery problem

$$\underset{\mathring{\mathbf{c}} \in \mathbb{C}^C}{\operatorname{argmin}} \left\| \mathbf{p} - \mathring{\mathbf{B}}_\mathrm{dyn} \, \mathring{\mathbf{c}} \right\|_2^2 \quad \text{s.t.} \quad \left\| \mathring{\mathbf{c}} \right\|_0 \leq K, \tag{15.49}$$

with the sensing matrix $\mathring{\mathbf{B}}_\mathrm{dyn} \in \mathbb{C}^{M \times C}$, sparse frequency parameters along the conical surface in $\mathring{\mathbf{c}} \in \mathbb{C}^C$, and $K < C < P$. Provided that sufficient incoherent measurements are available, (15.49) may be solved by compressed-sensing-based recovery algorithms outlined in Sect. 15.3.1.

### *15.3.4 Perfect Excitation Sequences*

The use of a deterministic $L$-shift cross-orthogonal excitation sequences $s(n) = s(n \bmod L)$ having perfect autocorrelation

$$r_{ss}(m) = \sigma_s^2 \, \delta(m \bmod L), \tag{15.50}$$

with $\sigma_s^2$ being the signal power, leads to an enhanced structure in the dynamic-sensing matrix that can be exploited for reducing the computational complexity of sparse recovery algorithms. Such signals can be constructed, for example, from scaled maximum-length sequences [43] with zero DC offset.

Let $\tilde{s}(n)$ be one period of a repetitive $L$-periodic source sequence $s(n)$ that satisfies (15.50). All circularly shifted versions of $\tilde{s}(n)$ are uncorrelated to each

other (which corresponds to constant-magnitude DFT coefficients). Accordingly, for $U$ periods of excitation in a steady-state situation, we obtain

$$s(n) * \tilde{s}(-n) = \gamma \sum_{m=0}^{U-1} \delta(n - mL), \tag{15.51}$$

where $\gamma$ is the energy of one period with

$$\gamma = L\sigma_s^2 = \sum_{n=0}^{L-1} |\tilde{s}(n)|^2 . \tag{15.52}$$

The relationship (15.51) can be expressed in matrix form by defining the circular convolution matrix $\tilde{\mathbf{S}} \in \mathbb{R}^{L\times L}$ for one period of excitation and the convolution matrix $\mathbf{S}_U \in \mathbb{R}^{LU\times M}$ comprising $U$ periods of excitation, both set up for the steady state with no zero padding at the boundaries. The uncorrelated excitation vectors in $\tilde{\mathbf{S}}$ lead to the orthogonality $\tilde{\mathbf{S}}\tilde{\mathbf{S}}^T = \gamma \mathbf{I}_L$, and the matrix-based formulation being equivalent to (15.51) reads

$$\mathbf{S}_U \tilde{\mathbf{S}}^T = \gamma \underbrace{[\mathbf{I}_L, \ldots, \mathbf{I}_L]}_{U \text{ times } \mathbf{I}_L}^T . \tag{15.53}$$

According to that, the dynamic measurement model (15.35) subject to $U$ excitation periods from a perfect source sequence satisfying (15.50) may be written as

$$\begin{aligned} \mathbf{p} &= \sum_{\mathbf{g}\in\Gamma} w_{\mathbf{g}}^{-1} \mathbf{D}_{\mathbf{g}} \mathbf{S}_U \tilde{\mathbf{S}}^T \tilde{\mathbf{S}} \gamma^{-1} \bar{\mathbf{h}}_{\mathbf{g}} + \boldsymbol{\eta} \\ &= \gamma^{-1} \mathbf{G} \left( \mathbf{W}^{-1} \otimes \tilde{\mathbf{S}} \right) \bar{\mathbf{h}} + \boldsymbol{\eta} \qquad (15.54) \\ &= \tilde{\mathbf{A}}_{\text{dyn}} \bar{\mathbf{h}} + \boldsymbol{\eta}, \qquad (15.55) \end{aligned}$$

where the matrix $\mathbf{G} \in \mathbb{R}^{M\times P}$ has favorable block-wise diagonal structure with

$$\mathbf{G} = \begin{bmatrix} \mathbf{D}_{\mathbf{g}_1}^{[1]} & \mathbf{D}_{\mathbf{g}_2}^{[1]} & \ldots & \mathbf{D}_{\mathbf{g}_G}^{[1]} \\ \mathbf{D}_{\mathbf{g}_1}^{[2]} & \mathbf{D}_{\mathbf{g}_2}^{[2]} & \ldots & \mathbf{D}_{\mathbf{g}_G}^{[2]} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{D}_{\mathbf{g}_1}^{[U]} & \mathbf{D}_{\mathbf{g}_2}^{[U]} & \ldots & \mathbf{D}_{\mathbf{g}_G}^{[U]} \end{bmatrix} \tag{15.56}$$

consisting of $U \times G$ blocks of the diagonal matrix $\mathbf{D}_{\mathbf{g}}^{[u]} \in \mathbb{R}^{L\times L}$ that carries the interpolation coefficients of the RIR at grid point $\mathbf{g}$ for the $u$-th period of excitation,

$$\mathbf{D}_{\mathbf{g}}^{[u]} = \operatorname{diag}\left\{\left[\varphi(\mathcal{G}(\mathbf{x}((u-1)L+1)) - \mathbf{g}), \ldots, \varphi(\mathcal{G}(\mathbf{x}((u-1)L+L)) - \mathbf{g})\right]\right\}. \tag{15.57}$$

Note the well-defined structure in (15.54) due to the perfect excitation sequence. In fact, the part $\left(\mathbf{W}^{-1} \otimes \tilde{\mathbf{S}}\right)\bar{\mathbf{h}}$ simply represents an orthogonal transformation of the sound-field signal in $\bar{\mathbf{h}}$ along the temporal dimension by use of $\tilde{\mathbf{S}}$. The crucial part of the sampling problem is modeled by the highly structured matrix $\mathbf{G}$ that is a direct result of the measurement trajectory. Compared to the original true spatio-temporal sampling problem modeled by (15.41) with (15.43), the temporal dimension de facto becomes separated from the spatial component. This results in a time-decoupled formulation where only the spatial part remains in the inverse problem. Owing to the block-wise diagonal structure of $\mathbf{G}$, the original spatio-temporal sampling problem is actually reduced to $L$ separate spatial sampling problems.

The incorporation of the structured measurement model (15.55) into the sparse recovery procedure described in Sect. 15.3.3 leads to the problem formulation

$$\begin{aligned}
\mathbf{p} &= \tilde{\mathbf{A}}_{\text{dyn}} \boldsymbol{\Psi}^H \mathbf{R}^T \mathbf{R} \boldsymbol{\Psi} \bar{\mathbf{h}} + \boldsymbol{\eta} \\
&= \gamma^{-1} \mathbf{G} \left(\mathbf{W}^{-1} \otimes \tilde{\mathbf{S}}\right) \boldsymbol{\Psi}^H \mathbf{R}^T \mathring{\mathbf{c}} + \boldsymbol{\eta} \\
&= \gamma^{-1} \mathbf{G} \left(\mathbf{W}^{-1} \boldsymbol{\Psi}_{\text{3D}}^H \otimes \tilde{\mathbf{S}} \mathbf{F}_L^H\right) \mathbf{R}^T \mathring{\mathbf{c}} + \boldsymbol{\eta} \\
&= \mathring{\mathbf{B}}_{\text{dyn}} \mathring{\mathbf{c}} + \boldsymbol{\eta},
\end{aligned} \tag{15.58}$$

with the dynamic-sensing matrix $\mathring{\mathbf{B}}_{\text{dyn}} \in \mathbb{C}^{M \times C}$ defined by

$$\mathring{\mathbf{B}}_{\text{dyn}} = \gamma^{-1} \mathbf{G} \left(\mathbf{W}^{-1} \boldsymbol{\Psi}_{\text{3D}}^H \otimes \tilde{\mathbf{S}} \mathbf{F}_L^H\right) \mathbf{R}^T, \tag{15.59}$$

and the $K$-sparse frequency vector $\mathring{\mathbf{c}}$ whose elements point to sampled frequency locations $(\mathbf{k}, l)$ subject to (15.46).

### 15.3.5 Algorithm for Sparse Recovery from Dynamic Measurements

Especially for larger audio bandwidths, the recovery of the four-dimensional sound-field signal poses a large-scale problem that demands for fast computational techniques. An efficient compressed-sensing-based recovery algorithm that works directly on the non-convex cost function (15.49) is the iterative hard-thresholding algorithm (IHT) [5]. By following a greedy strategy, the IHT converges to a local minimum of (15.49) with near-optimal error guarantees. The achievable estimation error depends linearly on the number of samples $M$ provided. The minimum number of required samples grows linearly with the sparsity measure $K$ and logarithmically with the dimension $C$ of the signal space. The number of necessary iterations is

logarithmic in the signal-to-noise ratio. It has been shown in [6] that the IHT has performance properties similar to those achieved by OMP, CoSaMP, and $\ell_1$-based methods.

The IHT approaches the $K$-sparse solution by use of the iterative scheme

$$\hat{\mathring{\mathbf{c}}}_{i+1} = \mathcal{T}_K \left\{ \hat{\mathring{\mathbf{c}}}_i + \nu \, \mathring{\mathbf{B}}_{\text{dyn}}^H \left( \mathbf{p} - \mathring{\mathbf{B}}_{\text{dyn}} \hat{\mathring{\mathbf{c}}}_i \right) \right\}, \tag{15.60}$$

where $i$ denotes the iteration number, $\nu \in [0, 1]$ is a sufficiently small step size [7], and $\mathcal{T}_K\{\cdot\}$ is the nonlinear thresholding operator that sets all but the $K$ largest absolute values in the signal to zero. One iteration involves a simple gradient descent step into the direction of the least-squares solution with step size $\nu$ followed by a hard projection of the signal estimate onto the subspace of its $K$-sparse representation. At this, the bottlenecks for computational complexity and memory demand are the operators $\mathring{\mathbf{B}}_{\text{dyn}}$ and $\mathring{\mathbf{B}}_{\text{dyn}}^H$. For straightforward matrix calculations, the effort is $\mathcal{O}(MC)$. However, as demonstrated previously, the sensing matrix $\mathring{\mathbf{B}}_{\text{dyn}}$ has well-defined structure due to the uniform-grid parameterization model. This structure excellently fits into the scheme (15.60) and can be exploited to express the IHT in terms of simplified and fast update equations that directly operate on the considered signals.

By applying the dynamic sampling model (15.47) to (15.60), the iterative scheme can be reformulated as

$$\hat{\mathring{\mathbf{c}}}_{i+1} = \mathcal{T}_K \left\{ \hat{\mathring{\mathbf{c}}}_i + \nu \, \mathbf{R}\boldsymbol{\Psi}(\mathbf{W}^{-1} \otimes \mathbf{I}_L) \, \mathbf{h}_i^{\text{up}} \right\}, \tag{15.61}$$

where $\mathbf{h}_i^{\text{up}} \in \mathbb{R}^P$ is the negative gradient of the least-squares problem to (15.41) for the simple case $w_{\text{3D}}(\mathbf{g}) = 1$. The vector $\mathbf{h}_i^{\text{up}} \in \mathbb{R}^P$ is the concatenation of the four-dimensional update signal $h_i^{\text{up}}(\mathbf{g}, n)$, $(\mathbf{W}^{-1} \otimes \mathbf{I}_L)$ performs the inverse windowing along its spatial dimension, and, finally, $\mathbf{R}\boldsymbol{\Psi}$ performs its partial four-dimensional DFT considering discrete frequencies according to conical structure (15.46). The partial DFT can be applied by use of the fast Fourier transform with $\mathcal{O}(P \log P)$ operations and a subsequent pruning. The crucial part is the calculation of $\mathbf{h}_i^{\text{up}}$. Efficient formulas for obtaining $\mathbf{h}_i^{\text{up}}$ are presented in Sects. 15.3.5.1 and 15.3.5.2 for the general case and, respectively, for a source signal having perfect autocorrelation according to (15.50). The latter case allows for cost-saving calculations.

Note that the IHT update (15.61) can also be represented in terms of the complete set of DFT coefficients $\bar{H}(\mathbf{k}, l)$ concatenated in $\bar{\mathbf{c}} = \boldsymbol{\Psi}\bar{\mathbf{h}}$. This increases the memory requirement a little bit; however, it is more straightforward to implement. Then, the iterative scheme becomes

$$\hat{\mathbf{c}}_{i+1} = \mathcal{T}_K \left\{ \hat{\mathbf{c}}_i + \nu \, \mathbf{M}\boldsymbol{\Psi}(\mathbf{W}^{-1} \otimes \mathbf{I}_L) \, \mathbf{h}_i^{\text{up}} \right\}, \tag{15.62}$$

with the $P \times P$ diagonal matrix $\mathbf{M} = \mathbf{R}^T\mathbf{R}$ that performs a frequency masking by setting frequencies to zero that do not live along the conical shape.

Subsequently, in Sects. 15.3.5.1 and 15.3.5.2, it is shown how the detailed knowledge about structures in the sensing matrix, which is provided in Sects. 15.3.2–15.3.4, enables us to define simple iterative schemes for sparse parameter recovery. Explicit matrix calculations and operations in the IHT dissolve into a couple of fast update equations.

#### 15.3.5.1 General Case

The block-wise structured sensing matrix of the dynamic sound-field sampling problem allows for calculating the spatio-temporal update signal in

$$\mathbf{h}_i^{\text{up}} = \left[ \mathbf{h}_{i,\mathbf{g}_1}^{\text{up}T}, \mathbf{h}_{i,\mathbf{g}_2}^{\text{up}T}, \ldots, \mathbf{h}_{i,\mathbf{g}_G}^{\text{up}\,T} \right]^T, \tag{15.63}$$

with

$$h_{i,\mathbf{g}}^{\text{up}} = \left[ h_i^{\text{up}}(\mathbf{g}, 0), h_i^{\text{up}}(\mathbf{g}, 1), \ldots, h_i^{\text{up}}(\mathbf{g}, L-1) \right]^T, \tag{15.64}$$

by the convolution of the source sequence with the particularly weighted residual vector $\boldsymbol{\varepsilon}_i \in \mathbb{R}^M$ according to

$$\mathbf{h}_{i,\mathbf{g}}^{\text{up}} = \mathbf{S}^T \mathbf{D}_{\mathbf{g}}\, \boldsymbol{\varepsilon}_i. \tag{15.65}$$

Remember that $\mathbf{D}_{\mathbf{g}} \in \mathbb{R}^{M\times M}$ is the diagonal matrix defined in (15.39). The residual vector can be obtained from (15.35) and is given by

$$\boldsymbol{\varepsilon}_i = \mathbf{p} - \sum_{\mathbf{g}\in\Gamma} \mathbf{D}_{\mathbf{g}} \mathbf{S}\, \hat{\mathbf{h}}_{i,\mathbf{g}}, \tag{15.66}$$

where $\hat{\mathbf{h}}_{i,\mathbf{g}} = [\hat{h}(\mathbf{g}, 0), \ldots, \hat{h}(\mathbf{g}, L-1)]^T$ is the current estimate for the unwindowed RIR at grid position $\mathbf{g}$, which is provided by the inverse transform of the currently estimated $K$-sparse signal representation subject to

$$\left[ \hat{\mathbf{h}}_{i,\mathbf{g}_1}^T, \hat{\mathbf{h}}_{i,\mathbf{g}_2}^T, \ldots, \hat{\mathbf{h}}_{i,\mathbf{g}_G}^T \right]^T = (\mathbf{W}^{-1} \otimes \mathbf{I}_L) \boldsymbol{\Psi}^H \mathbf{R}^T \hat{\tilde{\mathbf{c}}}_i. \tag{15.67}$$

The convolutions along the temporal dimension by $\mathbf{S}^T \in \mathbb{R}^{P\times M}$ and $\mathbf{S} \in \mathbb{R}^{M\times P}$ in (15.65) and (15.66), respectively, can be computed efficiently in Fourier domain. Besides, there are only point-wise multiplications according to the spatial components in $\mathbf{D}_{\mathbf{g}}$ and a summation for calculating the residual vector.

#### 15.3.5.2 Perfect-Excitation Case

By exploiting the time-decoupled structure of the dynamic-sensing matrix (15.59) for excitation with $U$ periods of a perfect autocorrelation sequence, the update vectors related to (15.65) can be calculated by

$$\mathbf{h}_{i,\mathbf{g}}^{\text{up}} = \tilde{\mathbf{S}}^T \sum_{u=1}^{U} \mathbf{D}_{\mathbf{g}}^{[u]} \boldsymbol{\varepsilon}_i^{[u]}, \tag{15.68}$$

where $\boldsymbol{\varepsilon}_i^{[u]} \in \mathbb{R}^L$ denotes the residual part that corresponds to the $u$-th period of excitation and is obtained from

$$\boldsymbol{\varepsilon}_i = \begin{bmatrix} \boldsymbol{\varepsilon}_i^{[1]} \\ \vdots \\ \boldsymbol{\varepsilon}_i^{[U]} \end{bmatrix} = \mathbf{p} - \sum_{\mathbf{g} \in \Gamma} \begin{bmatrix} \mathbf{D}_{\mathbf{g}}^{[1]} \\ \vdots \\ \mathbf{D}_{\mathbf{g}}^{[U]} \end{bmatrix} \tilde{\mathbf{S}} \, \hat{\mathbf{h}}_{i,\mathbf{g}}, \tag{15.69}$$

with $\mathbf{D}_{\mathbf{g}}^{[u]} \in \mathbb{R}^{L \times L}$ being the diagonal matrix defined in (15.57). Compared to (15.65) and (15.66), the computational effort is lowered in (15.68) and (15.69). The circular convolutions performed by matrices $\tilde{\mathbf{S}}^T \in \mathbb{R}^{L \times L}$ and $\tilde{\mathbf{S}} \in \mathbb{R}^{L \times L}$ can be calculated very fast in Fourier domain.

## 15.4 Coherence Analysis

As an a priori indicator for stable and robust sound-field reconstruction, the coherence of the dynamic-sensing matrix can be used (see Sect. 15.3.1). According to the compressed-sensing paradigm, the upper error bounds for the sparse signal recovery decrease with a lower coherence [8, 10, 15, 16, 23]. Note that uniform performance guarantees derived in several references always represent worst-case bounds for the theoretically worst possible scenario. In practice, however, typical signals and measurement setups involve most likely never the worst-case behavior for sparse recovery. Nevertheless, the coherence is in general a good indicator for evaluating the sampling process in terms of compressed sensing.

Defined by the Welch bound, the coherence may range subject to

$$\sqrt{\frac{C - M}{M(C-1)}} \leq \mu(\mathring{\mathbf{B}}_{\text{dyn}}) \leq 1. \tag{15.70}$$

For calculating $\mu(\mathring{\mathbf{B}}_{\text{dyn}})$, the naive approach of testing any normalized scalar product between two different columns in $\mathring{\mathbf{B}}_{\text{dyn}}$ according to (15.30) involves operations of order $\mathcal{O}(C^2)$. In this section, the specific signal structure in the dynamic-sensing

matrix $\mathbf{B}_{\text{dyn}}$ is analyzed with respect to spectrally flat excitation (Sect. 15.4.1), in order to derive a simple trajectory-dependent expression that allows for computing the coherence of $\mathbf{B}_{\text{dyn}}$ at complexity $\mathcal{O}(P)$ (Sect. 15.4.2). Remember that $\mathbf{B}_{\text{dyn}} \in \mathbb{C}^{M\times P}$ comprises all columns referring to the full set of discrete frequency variables $(\mathbf{k}, l)$, and $\mathring{\mathbf{B}}_{\text{dyn}} \in \mathbb{C}^{M\times C}$ contains the subset of columns from $\mathbf{B}_{\text{dyn}}$ according to the hyperconical characteristic (15.46). Thus, $\mu(\mathring{\mathbf{B}}_{\text{dyn}}) \leq \mu(\mathbf{B}_{\text{dyn}})$ can be used as upper bound. Finally, Sect. 15.4.3 presents conditions under which the assumption of spectrally flat excitation in each dimension holds true.

### 15.4.1 Influence of the Trajectory on the Sensing Matrix

Based on the dynamic measurement model (15.34), it can be seen that the $(n+1)$-th row of the measurement matrix $\mathbf{A}_{\text{dyn}}$ is built up by the discrete spatio-temporal excitation sequence

$$\begin{aligned}\bar{s}_n(\mathbf{g}, m) &= s(n-m)w_{\text{3D}}^{-1}(\mathbf{g})\varphi_{\text{3D}}(\mathcal{G}(\mathbf{x}(n)) - \mathbf{g}) \\ &= s(n-m)\,\bar{\phi}_n(\mathbf{g}) \end{aligned} \tag{15.71}$$

that contains the time-reversed source signal segment of length $L$, spatially weighted on $G$ grid points $\mathbf{g}$ according to the current microphone position at sampling time $n$. Accordingly, transferred into the sparsifying frequency domain, the $(n+1)$-th row of the resulting sensing matrix $\mathbf{B}_{\text{dyn}} = \mathbf{A}_{\text{dyn}}\boldsymbol{\Psi}^H$ carries the four-dimensional discrete Fourier transform

$$\bar{S}_n(\mathbf{k}, l) = \frac{1}{\sqrt{LG}} \sum_{m=0}^{L-1} \sum_{\mathbf{g}\in\Gamma} \bar{s}_n(\mathbf{g}, m)\, \mathrm{e}^{-2\pi\mathrm{i}\frac{l}{L}m} \mathrm{e}^{-2\pi\mathrm{i}\frac{k_x}{G_x}g_x} \mathrm{e}^{-2\pi\mathrm{i}\frac{k_y}{G_y}g_y} \mathrm{e}^{-2\pi\mathrm{i}\frac{k_z}{G_z}g_z}, \tag{15.72}$$

with sampled frequency variables $(\mathbf{k}, l)$ as defined in Sect. 15.3.3. Correspondingly, one column $\mathbf{b}$ of $\mathbf{B}_{\text{dyn}}$ concatenates $M$ sampled values of the DFT spectra (15.72) for one particular frequency combination $(\mathbf{k}', l')$ according to

$$\mathbf{b}_{(\mathbf{k}',l')} = \left[\bar{S}_0(\mathbf{k}', l'),\ \bar{S}_1(\mathbf{k}', l'),\ \ldots,\ \bar{S}_n(\mathbf{k}', l'),\ \ldots,\ \bar{S}_{M-1}(\mathbf{k}', l')\right]^T. \tag{15.73}$$

Let us convert the microphone trajectory into grid-related coordinates

$$\mathbf{r}(n) = \big(r_x(n), r_y(n), r_z(n)\big) = \left(\frac{x(n)}{\Delta_x}, \frac{y(n)}{\Delta_y}, \frac{z(n)}{\Delta_z}\right) \tag{15.74}$$

and assume perfectly flat spectra $\bar{S}_n(\mathbf{k}, l)$ in any dimension, i.e., a spectrally flat source sequence $s(n)$ as well as inversely windowed interpolation filters $\bar{\phi}_n(\mathbf{g}) = w_{\text{3D}}^{-1}(\mathbf{g})\varphi_{\text{3D}}(\mathcal{G}(\mathbf{x}(n)) - \mathbf{g})$ with ideal frequency response

$$\bar{\Phi}_n(\mathrm{e}^{\mathrm{i}\boldsymbol{\kappa}}) = \mathrm{e}^{-\mathrm{i}r_x(n)\kappa_x}\mathrm{e}^{-\mathrm{i}r_y(n)\kappa_y}\mathrm{e}^{-\mathrm{i}r_z(n)\kappa_z} \tag{15.75}$$

for each sampled inter-grid position $\mathbf{r}(n)$. Then, it can be stated that the moving of the microphone from measuring point $\mathbf{r}(n)$ to position $\mathbf{r}(n+d)$ corresponds to phase shifts in the sampled Fourier spectrum according to

$$S_{n+p}(\mathbf{k}, l) = S_n(\mathbf{k}, l)\mathrm{e}^{-2\pi\mathrm{i}p\frac{l}{L}}\mathrm{e}^{-2\pi\mathrm{i}d_x(n,p)\frac{k_x}{G_x}}\mathrm{e}^{-2\pi\mathrm{i}d_y(n,p)\frac{k_y}{G_y}}\mathrm{e}^{-2\pi\mathrm{i}d_z(n,p)\frac{k_z}{G_z}}, \tag{15.76}$$

involving the uniform phase delay $p$ in the temporal dimension and, in general, fractional phase shifts $\mathbf{d}(n, p) = \big(d_x(n, p), d_y(n, p), d_z(n, p)\big)$ in the spatial dimensions that depend on the grid-related trajectory subject to

$$\mathbf{d}(n, p) = \mathbf{r}(p) - \mathbf{r}(n). \tag{15.77}$$

Combining (15.76) and (15.73), each of the $P$ columns in $\mathbf{B}_{\text{dyn}}$ can be defined by structured temporal and spatial phase terms of the form

$$\mathbf{b}_{(\mathbf{k},l)} = \begin{bmatrix} b^0_{(\mathbf{k},l)} \\ b^0_{(\mathbf{k},l)}\,\mathrm{e}^{-2\pi\mathrm{i}1\frac{l}{L}}\,\mathrm{e}^{-2\pi\mathrm{i}\left(d_x(0,1)\frac{k_x}{G_x}+d_y(0,1)\frac{k_y}{G_y}+d_z(0,1)\frac{k_z}{G_z}\right)} \\ b^0_{(\mathbf{k},l)}\,\mathrm{e}^{-2\pi\mathrm{i}2\frac{l}{L}}\,\mathrm{e}^{-2\pi\mathrm{i}\left(d_x(0,2)\frac{k_x}{G_x}+d_y(0,2)\frac{k_y}{G_y}+d_z(0,2)\frac{k_z}{G_z}\right)} \\ \vdots \\ b^0_{(\mathbf{k},l)}\,\mathrm{e}^{-2\pi\mathrm{i}(M-1)\frac{l}{L}}\,\mathrm{e}^{-2\pi\mathrm{i}\left(d_x(0,M-1)\frac{k_x}{G_x}+d_y(0,M-1)\frac{k_y}{G_y}+d_z(0,M-1)\frac{k_z}{G_z}\right)} \end{bmatrix}, \tag{15.78}$$

where the initial phase state

$$b^0_{(\mathbf{k},l)} = \sigma_s\,\mathrm{e}^{\mathrm{i}\theta_0(l)}\mathrm{e}^{-2\pi\mathrm{i}\left(r_x(0)\frac{k_x}{G_x}+r_y(0)\frac{k_y}{G_y}+r_z(0)\frac{k_z}{G_z}\right)} \tag{15.79}$$

is determined by the initial grid delay $\mathbf{r}(0)$ and the initial phase $\theta_0(l)$ of the excitation signal with power $\sigma_s^2$ leading to the first microphone sample at time $n = 0$. Accordingly, all columns in $\mathbf{B}_{\text{dyn}}$ possess consistent norms

$$\left\|\mathbf{b}_{(\mathbf{k},l)}\right\|_2 = \sqrt{M\sigma_s^2}. \tag{15.80}$$

### *15.4.2 Coherence of Measurements*

By using the column representation (15.78) for calculating $\mu(\mathbf{B}_{\text{dyn}})$ in line with (15.30) and defining the distances between discrete frequency variables $(\mathbf{k}', l')$ and $(\mathbf{k}'', l'')$ in $\bar{\mathbf{f}} = (\bar{\mathbf{k}}, \bar{l})$, with $\bar{\mathbf{k}} = (\bar{k}_x, \bar{k}_y, \bar{k}_z)$ and

$$\bar{l} = l' - l'', \quad \bar{l} \in \{-(L-1), \ldots, L-1\},$$

$$\bar{k}_x = k'_x - k''_x\,, \quad \bar{k}_x \in \{-(G_x-1), \ldots, G_x-1\}\,,$$
$$\bar{k}_y = k'_y - k''_y\,, \quad \bar{k}_y \in \{-(G_y-1), \ldots, G_y-1\}\,,$$
$$\bar{k}_z = k'_z - k''_z\,, \quad \bar{k}_z \in \{-(G_z-1), \ldots, G_z-1\}\,,$$

the coherence can be described by

$$\mu(\mathbf{B}_{\text{dyn}}) = \max_{(\mathbf{k}',l')\neq(\mathbf{k}'',l'')} \frac{\left|\langle \mathbf{b}_{(\mathbf{k}',l')}, \mathbf{b}_{(\mathbf{k}'',l'')}\rangle\right|}{\left\|\mathbf{b}_{(\mathbf{k}',l')}\right\|_2 \left\|\mathbf{b}_{(\mathbf{k}'',l'')}\right\|_2} \tag{15.81}$$

$$= \max_{\bar{\mathbf{f}}\neq\mathbf{0}} \frac{1}{M} \left| \sum_{n=0}^{M-1} \mathrm{e}^{-2\pi \mathrm{i} n \frac{\bar{l}}{L}} \mathrm{e}^{-2\pi \mathrm{i}\left(r_x(n)\frac{\bar{k}_x}{G_x} + r_y(n)\frac{\bar{k}_y}{G_y} + r_z(n)\frac{\bar{k}_z}{G_z}\right)} \right|, \tag{15.82}$$

where the initial phase terms $\mathrm{e}^{\mathrm{i}(\theta_0(l')-\theta_0(l''))}$ and $\mathrm{e}^{2\pi \mathrm{i}\left(r_x(0)\frac{\bar{k}_x}{G_x} + r_y(0)\frac{\bar{k}_y}{G_y} + r_z(0)\frac{\bar{k}_z}{G_z}\right)}$ resulting from the scalar product in (15.81) are independent of the sum over $n$, have no effect on the magnitude, and, therefore, dissolve into (15.82). The trajectory-dependent expression (15.82) shows that the calculation of coherence between columns of absolute frequencies can be relativized to the equivalent problem of finding maximum correlation from possible differences in frequencies. Due to this relationship, calculations reduce to linear complexity $\mathcal{O}(P)$. Moreover, as the excitation signals $\bar{s}_n(\mathbf{g}, m)$ are real-valued, the sampled spectra $\bar{S}_n(\mathbf{k}, l)$ are conjugate symmetric. This may be exploited for saving further computational cost by reducing the set of possible frequency differences in two dimensions, for example, according to $\bar{l} \in \{0, \ldots, L-1\}$ and $\bar{k}_x \in \{0, \ldots, G_x - 1\}$.

The expression (15.82) can be used to provide an upper bound for the evaluation of the sparse recovery problem where only frequencies along the spectral hypercone are considered. According to (15.46), the sensing matrix $\mathring{\mathbf{B}}_{\text{dyn}}$ comprises columns that involve discrete spatial frequencies ranging subject to

$$\mathring{k}_\xi \in \{-\mathring{k}_\xi^{\max}, \ldots, \mathring{k}_\xi^{\max}\} \tag{15.83}$$

with maximum frequencies

$$\mathring{k}_\xi^{\max} = \left\lceil \frac{f_\mathrm{s} \Delta_\xi (L-1)(G_\xi - 1)}{2 c_0 L} + \epsilon_\mathrm{s} \right\rceil \tag{15.84}$$

for each dimension $\xi \in \{x, y, z\}$. Consequently, we obtain the upper coherence bound

$$\mu(\mathring{\mathbf{B}}_{\text{dyn}}) \leq \max_{\bar{\mathbf{f}}\neq\mathbf{0}} \frac{1}{M} \left| \sum_{n=0}^{M-1} \mathrm{e}^{-2\pi \mathrm{i} n \frac{\bar{l}}{L}} \mathrm{e}^{-2\pi \mathrm{i}\left(r_x(n)\frac{\bar{\bar{k}}_x}{G_x} + r_y(n)\frac{\bar{\bar{k}}_y}{G_y} + r_z(n)\frac{\bar{\bar{k}}_z}{G_z}\right)} \right|, \tag{15.85}$$

where possible frequency differences $\mathring{\bar{\mathbf{f}}}$ can be selected from a subset of spatial frequencies in accordance with (15.83) and (15.84).

### 15.4.3 Spectrally Flat Spatio-Temporal Excitation

The quantity (15.82) is equivalent to the coherence (15.30) for spectrally flat behavior of the measuring process in any dimension, which can be met by selecting appropriate design parameters. For the temporal dimension, the choice of $L$-shift cross-orthogonal excitation sequences is suitable (see Sect. 15.3.5.2). For the spatial dimension, a dense grid design with more than twofold oversampling and the use of maximally flat higher-order interpolation polynomials are appropriate. Then, the bandlimited signal is located at the lower half-band where the interpolator approaches ideal response, and the measurement model can be reduced to these low spatial frequencies. For a non-ideal design, the measure (15.82) is an efficient approximation of the coherence (cf. [26]).

Let us consider (15.75) for the simplified case $w_{3\mathrm{D}}(\mathbf{g}) = 1$. The filters $\bar{\phi}_n(\mathbf{g})$ fulfill a spatial alignment task. They perform a fractional delay (FD) in space on the sound field $h(\mathbf{g}, n)$, in order to fit encoded samples $h(\mathbf{r}(n), n)$ taken in continuous space into the modeled spatial grid. For one separated dimension, the impulse response of an ideal FD filter is a shifted and sampled sinc function, $\bar{\phi}_n^{\mathrm{id}}(g_x) = \mathrm{sinc}(g_x - r_x(n))$, where the delay $r_x(n)$ consists of the integer part $\lfloor r_x(n) \rfloor$ and the fractional part $q_x(n) = r_x(n) - \lfloor r_x(n) \rfloor$. Thus, the ideal frequency response of a FD filter reads

$$\bar{\Phi}_n^{\mathrm{id}}(\mathrm{e}^{\mathrm{i}\kappa_x}) = \mathrm{e}^{-\mathrm{i}r_x(n)\kappa_x}, \tag{15.86}$$

with constant magnitude response

$$|\bar{\Phi}_n^{\mathrm{id}}(\mathrm{e}^{\mathrm{i}\kappa_x})| = 1, \tag{15.87}$$

linear phase response

$$\arg\left\{\bar{\Phi}_n^{\mathrm{id}}(\mathrm{e}^{\mathrm{i}\kappa_x})\right\} = \theta_n^{\mathrm{id}}(\mathrm{e}^{\mathrm{i}\kappa_x}) = -r_x(n)\kappa_x, \tag{15.88}$$

and constant phase delay $\tau_n^{\mathrm{id}} = r_x(n)$. For $q_x(n) \neq 0$, the ideal FD filter has infinite length and, thus, is not realizable.

In order to design a realizable FD filter, several finite-length approximations for the sinc function have been proposed. A discussion and comparison of common techniques including non-recursive (FIR) and recursive (IIR) filter approximations can be found in [49]. The maximally flat FD FIR filter approximation of length $N+1$ is equivalent to the coefficients of the classical Lagrange interpolation method considering an $N$-th-order polynomial. The maximum order of the interpolating polynomial is limited by confining the support of the Lagrange kernel to local

grid points. Choosing an odd-order Lagrange interpolator and centering the support of $\bar{\phi}_n(g_x)$ around the measurement position $r_x(n)$ yield a maximum of one in the resulting magnitude response and allow for the best performance. The arising approximation error is highly dependent on the fractional part $q_x(n)$. The worst case occurs with a fractional delay of $q_x(n) = 0.5$, which leads to an excessive magnitude error at high frequencies and even to an exact zero at the Nyquist frequency. However, at low frequencies, the magnitude and phase-delay curves coincide with the ideal response for any $q_x(n)$. By designing higher-order polynomials, this almost perfect behavior can be obtained for the entire lower frequency half-band. Thus, for a notional grid with spacing $\Delta_x \leq c_0/(4 f_{\max})$ that leads to at least twofold spatial oversampling, the Lagrange filter achieves nearly optimal bandlimited interpolation with nearly flat spectral characteristic. The additional weighting by an inverse Hamming window according to (15.19) influences the frequency response only marginally.

## 15.5 Trajectory Optimization

It has been shown in [25] that for sufficiently long sampling the minimum mean squared error of estimates from the inverse problem (15.41) with (15.43) becomes smaller for trajectory positions sampled closer to the notional grid points, independently from the actual interpolation accuracy. For the sparse recovery in frequency domain, no such general statement on the trajectory is applicable. According to compressed-sensing theory, a random microphone trajectory would be a good choice for generating incoherent measurements with high probability for a wide range of constellations $K < M < C$ [41, 42]. However, realistic trajectories cannot be totally random, since the current position of the microphone is highly dependent on its previous position. The speed of the microphone is limited in practice, so, usually, it is impossible to reach any location inside $\bar{\Omega}_\mathrm{i}$ instantly. In [26], the use of a Lissajous trajectory covering the entire volume of interest has been experimentally shown to be a good choice, performing even better than totally random dynamic sampling positions.

### *15.5.1 Techniques for Measurement Matrix Optimization*

Several iterative strategies have been proposed [1, 19, 29, 37, 38], in order to reduce the coherence of measurements and obtain well-designed deterministic matrices with optimized performance compared to totally random choices. Most of these methods operate directly on the sensing matrix or, respectively, on the resulting Gram matrix. However, considering the dynamic sound-field sampling problem, entries of the matrix are not arbitrary, but arranged according to certain structures as described in Sects. 15.3 and 15.4. The elements of the sensing matrix are not the

design variables themselves, but rather result from particular relationships within the multidimensional sampling problem. For such cases, where the sensing matrix can be related to the design variables by use of a differentiable, nonlinear function, an iterative optimization scheme has been proposed in [37]. This algorithm performs an alternating minimization procedure and solves augmented Lagrangian subproblems, in order to lower the coherence subject to the model parameters.

The dynamic sampling procedure possesses several design variables. However, most of them are highly restricted according to the measurement setup and, thus, are more or less predefined in advance. For the temporal dimension, there is the source sequence $s(n)$ that is arbitrary; however, it should cover the entire spectrum of the bandlimited sound-field signal to be measured. Thus, white noise and perfect sequences are convenient. In the spatial dimensions, model parameters are the grid location that is essentially defined by the measurement volume, the observation window that should be larger than the volume of interest for accurate interpolation and designed according to the spectral requirements described in Sect. 15.2.3.1, and the spacings $\Delta_\xi$ of the notional grid that should be modeled subject to about twofold spatial oversampling, in order to achieve (nearly) perfectly flat responses by realizable FD filters at the considered frequencies in the lower half-band (see Sect. 15.4.3).

The essential design variables in the dynamic sampling process are the trajectory positions. A predefined microphone trajectory may lead to a sensing matrix with high coherence. However, just slight changes in dynamic positions often lead to a much lower coherence and reduced recovery error. For customizing the trajectory, a cost-effective optimization procedure is highly favorable as the number of spatial measuring points is, in general, as high as the number of provided samples $M$. For all three dimensions in space, this results in $3M$ free design variables that basically determine the coherence of $\mathring{\mathbf{B}}_{\mathrm{dyn}}$. For example, dynamic sampling at 16 kHz for a duration of 20 s involves about $10^6$ position variables.

The low-complexity measure (15.85) is an excellent basis for deriving a simple and fast algorithm that enables us to optimize the coherence of $\mathring{\mathbf{B}}_{\mathrm{dyn}}$ subject to the trajectory positions [27]. By assuming a spatio-temporal excitation with spectral-flatness character, the optimal grid-related trajectory is considered as

$$\mathbf{r}_{\mathrm{opt}}(n) = \underset{\mathbf{r}(n)}{\operatorname{argmin}} \left( \max_{\bar{\bar{\mathbf{f}}} \neq \mathbf{0}} \left| \sum_{n=0}^{M-1} \mathrm{e}^{-2\pi \mathrm{i} n \frac{\bar{l}}{L}} \mathrm{e}^{-2\pi \mathrm{i} \left( r_x(n) \frac{\bar{k}_x}{G_x} + r_y(n) \frac{\bar{k}_y}{G_y} + r_z(n) \frac{\bar{k}_z}{G_z} \right)} \right| \right). \tag{15.89}$$

In order to improve given trajectory positions, an update scheme can be used that, first, efficiently identifies the origin of maximum spectral correlation subject to (15.85) and, then, performs a gradient descent step for that current maximum subject to the free position variables. This greedy-like method is highly efficient for iteratively approaching a local minimum of maximum correlation and, thus, obtaining a sensing matrix better suited to the CS paradigm.

### 15.5.2 Fast Update Scheme for Trajectory Adjustments

A simple procedure allows for updating the trajectory $\mathbf{r}(n)$ subject to the minimization of the maximum correlation between two columns in $\mathring{\mathbf{B}}_{\text{dyn}}$. By exploiting the low-complexity measure (15.85), the objective function

$$J(\mathbf{r}(\cdot)) = \left| \sum_{n=0}^{M-1} \mathrm{e}^{-\mathrm{i}\mathcal{T}_{\bar{l}'}(n)} \mathrm{e}^{-\mathrm{i}\mathcal{X}_{\bar{\mathbf{k}}'}(\mathbf{r}(n))} \right| \tag{15.90}$$

may be defined, with $\mathring{\bar{\mathbf{k}}}' = (\mathring{\bar{k}}'_x, \mathring{\bar{k}}'_y, \mathring{\bar{k}}'_z)$, the temporal relationship

$$\mathcal{T}_{\bar{l}'}(n) = 2\pi n \frac{\bar{l}}{L}, \tag{15.91}$$

and the positional dependency

$$\mathcal{X}_{\bar{\mathbf{k}}'}(\mathbf{r}(n)) = 2\pi \left( r_x(n) \frac{\mathring{\bar{k}}'_x}{G_x} + r_y(n) \frac{\mathring{\bar{k}}'_y}{G_y} + r_z(n) \frac{\mathring{\bar{k}}'_z}{G_z} \right). \tag{15.92}$$

The frequency distances in $\mathring{\bar{\mathbf{f}}}' = (\mathring{\bar{l}}', \mathring{\bar{\mathbf{k}}}')$ are selected in accordance with the highest spectral correlation for the current trajectory,

$$\mathring{\bar{\mathbf{f}}}' = \underset{\mathring{\bar{\mathbf{f}}} \neq \mathbf{0}}{\operatorname{argmax}} \left| \sum_{n=0}^{M-1} \mathrm{e}^{-\mathrm{i}\mathcal{T}_{\bar{l}}(n)} \mathrm{e}^{-\mathrm{i}\mathcal{X}_{\bar{\mathbf{k}}}(\mathbf{r}(n))} \right|. \tag{15.93}$$

In order to minimize (15.90) with respect to the limitations of the measurement process, different scenarios can be considered by adapting either one single position (e.g., to find the optimal direction of future movement), multiple, or even all points on the trajectory simultaneously at iteration $i$. The latter case emphasizes the need for a computationally efficient optimization. Updates for one particular position variable $r_\xi(n^*)$ are performed following the gradient descent scheme

$$r_\xi^{[i+1]}(n^*) = r_\xi^{[i]}(n^*) - \beta \, \frac{\partial J(\mathbf{r}(\cdot))}{\partial r_\xi^{[i]}(n^*)}, \tag{15.94}$$

with $\beta$ being a small step size. For satisfying convergence conditions, a sufficient choice of $\beta$ can be obtained from common step-size rules [36]. Each iteration goes along with a redesign of the objective function in a greedy fashion: if the origin of the maximum correlation, i.e., (15.93), relocates, then $J(\mathbf{r}(\cdot))$ is adapted accordingly.

Using Euler's formula, the objective function can be rewritten as

$$J(\mathbf{r}(\cdot)) = \left( \left( \sum_{n=0}^{M-1} \cos\left( \mathcal{T}_{\bar{\bar{l}}'}(n) + \mathcal{X}_{\bar{\bar{\mathbf{k}}}'}(\mathbf{r}(n)) \right) \right)^2 + \left( \sum_{n=0}^{M-1} \sin\left( \mathcal{T}_{\bar{\bar{l}}'}(n) + \mathcal{X}_{\bar{\bar{\mathbf{k}}}'}(\mathbf{r}(n)) \right) \right)^2 \right)^{\frac{1}{2}}, \tag{15.95}$$

in order to deduce simple expressions for the partial derivatives composing the considered gradient. By applying the chain rule several times and using the trigonometric identity $\sin(a \pm b) = \sin a \cos b \pm \cos a \sin b$, the partial derivative of (15.95) subject to the specific position variable simply reads

$$\frac{\partial J(\mathbf{r}(\cdot))}{\partial r_\xi(n^*)} = \gamma_\xi \sum_{n=0}^{M-1} \sin\left( \mathcal{T}_{\bar{\bar{l}}'}(n - n^*) + \mathcal{X}_{\bar{\bar{\mathbf{k}}}'}(\mathbf{r}(n) - \mathbf{r}(n^*)) \right), \tag{15.96}$$

where the factor

$$\gamma_\xi = \frac{2\pi \bar{\bar{k}}'_\xi}{D_\xi J(\mathbf{r}(\cdot))} \tag{15.97}$$

depends on the particular dimension $\xi \in \{x, y, z\}$ and the corresponding size $D_\xi \in \{G_x, G_y, G_z\}$.

All in all, efficient adjustments of trajectory positions can be performed by iteratively finding the maximum spectral correlation according to (15.93), calculating the gradient of the free variables in terms of (15.96), and updating subject to (15.94), until some predefined exit conditions are reached. These could be, for example, restrictions due to the measurement setup, i.e., boundaries of the spatial grid, maximum distance between positions, or maximum microphone speed.

In Fig. 15.4, outcomes of the update scheme are presented for the adaptation of positions on a Lissajous trajectory as well as positions resulting from an autoregressive moving average (ARMA) process in each dimension. Both trajectories were sampled at $4 \cdot 10^5$ points, which leads to $1.2 \cdot 10^6$ spatial design variables that determine the coherence of the corresponding sensing matrix. For this example, the coherences of the original states are $\mu(\mathring{\mathbf{B}}_{\text{dyn}}) = 0.18$ for the Lissajous trajectory and $\mu(\mathring{\mathbf{B}}_{\text{dyn}}) = 0.65$ for the trajectory based on ARMA processes. Without building up the large matrices, the optimization technique was capable of reducing the coherences by directly manipulating the trajectory positions. At this, no constraints were made on the resulting microphone velocity. The application of the update rule (15.94) with (15.96) on each trajectory point at step size $\alpha = 0.1/(2\pi)$ obtained improved setups where the coherence was significantly lowered by 0.1 after only a couple of iterations, and finally reached a minimum in $\mu(\mathring{\mathbf{B}}_{\text{dyn}}) = 0.014$ and $\mu(\mathring{\mathbf{B}}_{\text{dyn}}) = 0.021$, respectively. For both types of trajectories, just little changes in $r_\xi(n)$ by 0.2 on average led to the coherence improvement by 0.1. Numerical experiments for various acoustic scenarios show that the coherence-based trajectory optimization in turn improves the performance of the sparse sound-field recovery in frequency domain [27]. Here, two main cases can be distinguished, as outlined in the following.

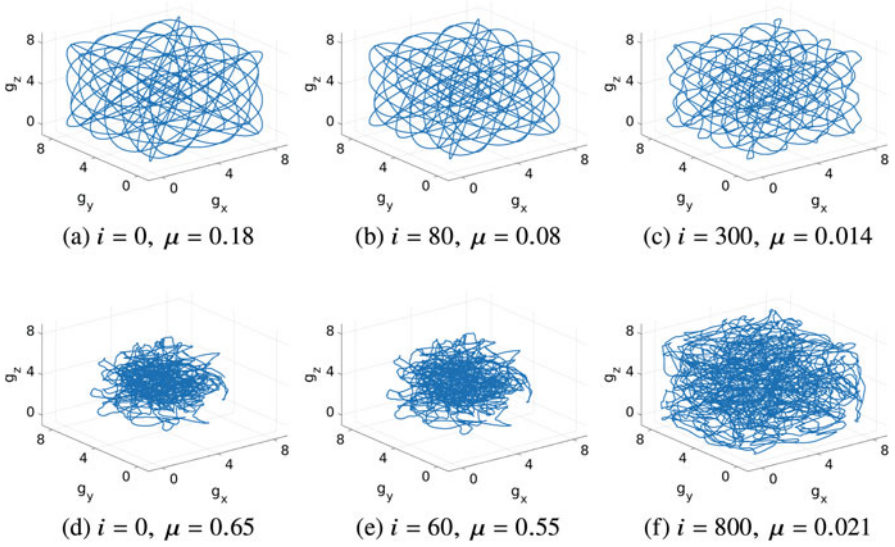
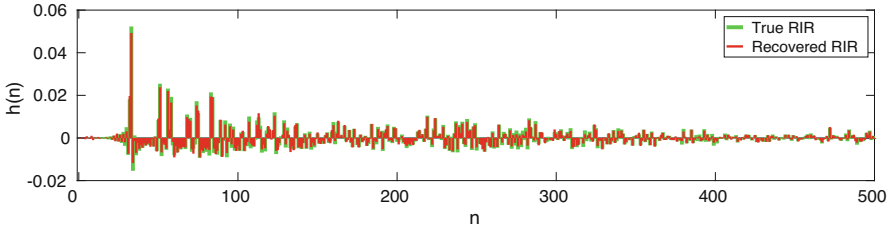


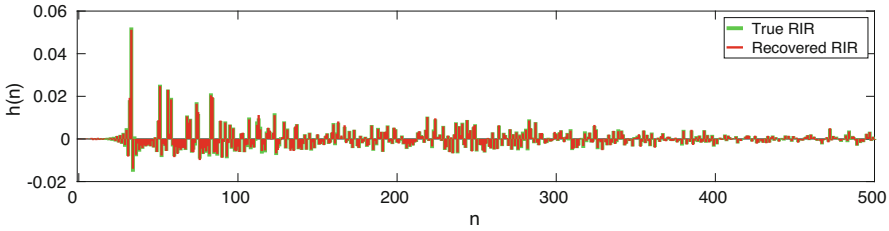

**Fig. 15.4** Optimized microphone trajectories. (**a**)–(**c**) Lissajous trajectory and (**d**)–(**f**) ARMA trajectory in their original states at $i = 0$ and their improved versions after $i$ iterations leading to lower coherence $\mu$

For ARMA-process-based trajectories that originally lead to a lack of sampling positions near the boundaries of the cubical volume $\bar{\Omega}_\mathrm{i}$, the positional optimization converges to a much more expanded trajectory configuration with improved spatial coverage (cf. Fig. 15.4d–f). The wider spread of spatial sampling points lowers the coherence of the sensing matrix drastically and leads to a more homogeneous recovery quality with an overall error reduction. Note that, in practice, the spread of a defined number of trajectory positions will be limited by the maximum speed of the microphone and must be constrained within the optimization procedure.

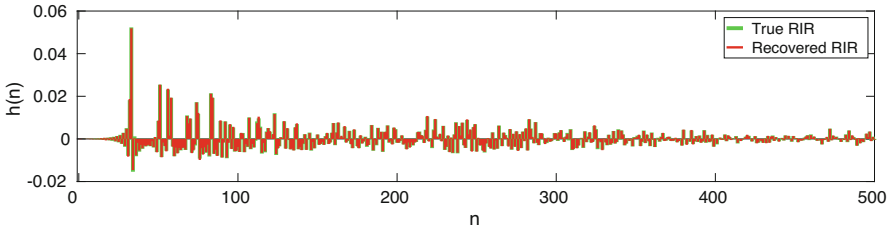
For the case where the original scenario already involves a widespread trajectory, the optimization method performs only slight adjustments of positions. Comparing the Lissajous-like trajectories in Fig. 15.4a and b, the outcome of optimization is hardly visible. Positional differences are in the range of a few millimeters in practice. However, such minor manipulations in the sampling setup have a major effect on the sensing matrix and the CS-based recovery. Especially at positions in areas where only a small number of dynamic samples are available, the optimized trajectory enables more accurate RIR reconstruction. An example for this is depicted in Fig. 15.5, where the RIR recovered at the center position of $\bar{\Omega}_\mathrm{i}$ is compared for the different trajectories in Fig. 15.4a–c given a fixed acoustic scenario (reverberation time: 0.25 s, $L = 2000$, $M/P = 0.25$, $f_\mathrm{s} = 8$ kHz, signal-to-noise ratio: 20 dB). By defining the normalized quality measure




(a) $\mu = 0.18$, Err $= -13$ dB.




(b) $\mu = 0.08$, Err $= -18$ dB.




(c) $\mu = 0.014$, Err $= -23$ dB.

**Fig. 15.5** Early part of the recovered RIR, sampled by using (**a**) a Lissajous trajectory with $\mu = 0.18$, (**b**) its optimized version with $\mu = 0.08$, and (**c**) its optimized version with $\mu = 0.014$

$$\text{Err} = \frac{\|\mathbf{h}_{\text{true}} - \mathbf{h}_{\text{rec}}\|_2^2}{\|\mathbf{h}_{\text{true}}\|_2^2}, \tag{15.98}$$

with $\mathbf{h}_{\text{true}} \in \mathbb{R}^L$ being the ground truth and $\mathbf{h}_{\text{rec}} \in \mathbb{R}^L$ being the recovered RIR, the recovery error based on the original Lissajous trajectory is Err $= -13$ dB. Using the optimized versions from Fig. 15.4b and c results in reduced recovery errors of Err $= -18$ dB and Err $= -23$ dB, respectively.

## 15.6 Summary

In this chapter, basic ideas and strategies for the dynamic sound-field sampling problem have been presented. The parameterization of the particular sound field in terms of Cartesian coordinates and sinc-function-based interpolation filters allowed for the direct incorporation of the dynamic measurement model into a compressed-sensing-based recovery procedure, where sparsity in frequency domain is exploited. Due to the uniform-grid model, both the required spatial interpolation and the sparsifying transformation can be performed efficiently and separately in each dimension. The separability of the dimensions led to a sensing matrix with highly structured block components. The detailed analysis of the specific structure in terms of involved signals and design variables revealed simple mathematical expressions that enable us to avoid large matrix operations and accomplish tasks such as sparse recovery, coherence calculation, and trajectory optimization at low effort in terms of computation and memory.

## References

1. Abolghasemi, V., Ferdowsi, S., Sanei, S.: A gradient-based alternating minimization approach for optimization of the measurement matrix in compressive sensing. Signal Process. **92**, 999–1009 (2012)
2. Ajdler, T., Sbaiz, L., Vetterli, M.: The plenacoustic function and its sampling. IEEE Trans. Signal Process. **54**(10), 3790–3804 (2006)
3. Antonello, N., Sena, E.D., Moonen, M., Naylor, P.A., van Waterschoot, T.: Room impulse response interpolation using a sparse spatio-temporal representation of the sound field. IEEE/ACM Trans. Audio Speech Lang. Process. **25**(10), 1929–1941 (2017)
4. Benichoux, A., Simon, L., Vincent, E., Gribonval, R.: Convex regularizations for the simultaneous recording of room impulse responses. IEEE Trans. Signal Process. **62**(8), 1976–1986 (2014)
5. Blumensath, T., Davies, M.: Iterative thresholding for sparse approximations. J. Fourier Anal. Appl. **14**(5-6), 629–654 (2008)
6. Blumensath, T., Davies, M.: Iterative hard thresholding for compressed sensing. Appl. Comput. Harmon. Anal. **27**(3), 265–274 (2009)
7. Blumensath, T., Davies, M.: Normalized iterative hard thresholding: Guaranteed stability and performance. IEEE J. Sel. Topics Signal Process. **4**(2), 298–309 (2010)
8. Cai, T., Xu, G., Zhang, J.: On recovery of sparse signals via $\ell_1$ minimization. IEEE Trans. Inf. Theory **55**(7), 3388–3397 (2009)
9. Caiafa, C., Cichocki, A.: Computing sparse representations of multidimensional signals using Kronecker bases. Neural Comput. **25**(1), 186–220 (2012)
10. Candès, E.: The restricted isometry property and its implications for compressed sensing. Comptes Rendus Mathematique **346**(9), 589–592 (2008)
11. Candès, E., Tao, T.: The Dantzig selector: Statistical estimation when $p$ is much larger than $n$. Ann. Statist. **35**(6), 2313–2351 (2007)
12. Candès, E., Romberg, J., Tao, T.: Stable signal recovery from incomplete and inaccurate measurements. Commun. Pure Appl. Math. **59**(8), 1207–1223 (2006)
13. Cecchi, S., Carini, A., Spors, S.: Room response equalization – a review. Appl. Sci. **8**(1) (2018)

14. Chen, S.S., Donoho, D.L., Saunders, M.A.: Atomic decomposition by basis pursuit. SIAM J. Sci. Comput. **20**(1), 33–61 (1998)
15. Davenport, M., Duarte, M., Eldar, Y., Kutyniok, G.: Introduction to compressed sensing. In: Y. Eldar, G. Kutyniok (eds.) Compressed Sensing - Theory and Applications, pp. 1–64. Cambridge University Press, New York (2012)
16. Donoho, D.L., Elad, M.: Optimally sparse representation in general (nonorthogonal) dictionaries via $\ell_1$ minimization. Proc. Natl. Acad. Sci., 2197–2202 (2003)
17. Donoho, D.L., Huo, X.: Uncertainty principles and ideal atomic decomposition. IEEE Trans. Inf. Theory **47**(7), 2845–2862 (2001)
18. Duarte, M., Baraniuk, R.: Kronecker compressive sensing. IEEE Trans. Image Process. **21**(2), 494–504 (2012)
19. Elad, M.: Optimized projections for compressed sensing. IEEE Trans. Signal Process. **55**(12), 5695–5702 (2007)
20. Epain, N., Jin, C., van Schaik, A.: The application of compressive sampling to the analysis and synthesis of spatial sound fields. In: 127th Conv. Audio Eng. Soc. (2009)
21. Fernandez-Grande, E., Xenaki, A.: Compressive sensing with a spherical microphone array. J. Acoust. Soc. Am. **139**(2), EL45–EL49 (2016)
22. Hänsler, E., Schmidt, G. (eds.): Topics in Acoustic Echo and Noise Control - Selected Methods for the Cancellation of Acoustical Echoes, the Reduction of Background Noise, and Speech Processing. Signals and Communication Technology. Springer Science (2006)
23. Herman, M., Strohmer, T.: General deviants: An analysis of perturbations in compressed sensing. IEEE J. Sel. Topics Signal Process. **4**(2), 342–349 (2010)
24. Jokar, S., Mehrmann, V.: Sparse solutions to underdetermined Kronecker product systems. Linear Algebra Appl. **431**, 2437–2447 (2009)
25. Katzberg, F., Mazur, R., Maass, M., Koch, P., Mertins, A.: Sound-field measurement with moving microphones. J. Acoust. Soc. Am. **141**(5), 3220–3235 (2017)
26. Katzberg, F., Mazur, R., Maass, M., Koch, P., Mertins, A.: A compressed sensing framework for dynamic sound-field measurements. IEEE/ACM Trans. Audio Speech Lang. Process. **26**(11), 1962–1975 (2018)
27. Katzberg, F., Maass, M., Mertins, A.: Coherence based trajectory optimization for compressive sensing of sound fields. In: Europ. Signal Process. Conf. (2021)
28. Katzberg, F., Maass, M., Mertins, A.: Spherical harmonic representation for dynamic sound-field measurements. In: IEEE Int. Conf. Acoust. Speech, Signal Process. (2021)
29. Lu, C., Li, H., Lin, Z.: Optimized projections for compressed sensing via direct mutual coherence minimization. Signal Process. **151**, 45–55 (2018)
30. Masiero, B., Pollow, M.: A review of the compressive sampling framework in the lights of spherical harmonics: Applications to distributed spherical arrays. In: 2nd Int. Symposium on Ambisonics and Spherical Acoustics (2010)
31. Mignot, R., Daudet, L., Ollivier, F.: Room reverberation reconstruction: interpolation of the early part using compressed sensing. IEEE/ACM Trans. Audio Speech Lang. Process. **21**(11), 2301–2312 (2013)
32. Mignot, R., Chardon, G., Daudet, L.: Low frequency interpolation of room impulse responses using compressed sensing. IEEE/ACM Trans. Audio Speech Lang. Process. **22**(1), 205–216 (2014)
33. Moreau, S., Daniel, J., Bertet, S.: 3D sound field recording with higher order ambisonics - objective measurements and validation of a 4th order spherical microphone. In: 120th Conv. Audio Eng. Soc. (2006)
34. Natarajan, B.: Sparse approximate solutions to linear systems. SIAM J. Comput. **24**(2), 227–234 (1995)
35. Needell, D., Tropp, J.A.: CoSaMP: iterative signal recovery from incomplete and inaccurate samples. Appl. Comput. Harmon. Anal. **26**(3), 301–321 (2008)
36. Nocedal, J., Wright, J.: Numerical Optimization, 2nd edn. Springer (2006)
37. Obermeier, R., Martinez-Lorenzo, J.: Sensing matrix design via mutual coherence minimization for electromagnetic compressive imaging applications. IEEE Trans. Comput. Imag. **3**(2), 217–229 (2017)

38. Pan, J., Qiu, Y.: An orthogonal method for measurement matrix optimization. Circuits Syst. Signal Process. **35**, 837–849 (2015)
39. Rafaely, B.: Analysis and design of spherical microphone arrays. IEEE Trans. Speech Audio Process. **12**(1), 135–143 (2005)
40. Rafaely, B.: Fundamentals of Spherical Array Processing. Springer (2015)
41. Rauhut, H.: Random sampling of sparse trigonometric polynomials. Appl. Comput. Harmon. Anal. **22**(1), 16–42 (2007)
42. Rauhut, H.: Compressive sensing and structured random matrices. In: M. Fornasier (ed.) Theoretical foundations and numerical methods for sparse recovery, Radon Series Comp. Appl. Math. 9, pp. 1–94. de Gruyter, Berlin (2010)
43. Rife, D., Vanderkooy, J.: Transfer-function measurement with maximum-length sequences. J. Audio Eng. Soc. **37**(6), 419–444 (1989)
44. Spors, S., Rabenstein, R., Ahrens, J.: The theory of wave field synthesis revisited. In: 124th Conv. Audio Eng. Soc. (2008)
45. Takida, Y., Koyama, S., Saruwatari, H.: Exterior and interior sound field separation using convex optimization: Comparison of signal models. In: Europ. Signal Process. Conf. (2018)
46. Tibshirani, R.: Regression shrinkage and selection via the LASSO. J. R. Stat. Soc. B **58**(1), 267–288 (1996)
47. Tillmann, A.M., Pfetsch, M.E.: The computational complexity of the restricted isometry property, the nullspace property, and related concepts in compressed sensing. IEEE Trans. Inf. Theory **60**(2), 1248–1259 (2013)
48. Tropp, J.A., Gilbert, A.C.: Signal recovery from random measurements via orthogonal matching pursuit. IEEE Trans. Inf. Theory **53**(12), 4655–4666 (2007)
49. Välimäki, V., Laakso, T.: Principles of fractional delay filters. In: IEEE Int. Conf. Acoust., Speech, Signal Process., pp. 3870–3873 (2000)
50. Wabnitz, A., Epain, N., van Schaik, A., Jin, C.: Time domain reconstruction of spatial sound fields using compressed sensing. In: IEEE Int. Conf. Acoust., Speech, Signal Process., pp. 465–468 (2011)
51. Wang, Y., Chen, K.: Compressive sensing based spherical harmonics decomposition of a low frequency sound field within a cylindrical cavity. J. Acoust. Soc. Am. **141**(1), 1812–1823 (2017)
52. Zea, E.: Compressed sensing of impulse responses in rooms of unknown properties and contents. J. Sound Vib. **459** (2019)

# Chapter 16
# Compressed Sensing in the Spherical Near-Field to Far-Field Transformation


**Cosme Culotta-López, Arya Bangun, Rudolf Mathar, and Dirk Heberling**

## 16.1 Spherical Near-Field Antenna Measurements

In spherical near-field (SNF) measurements, the interaction between an antenna under test (AUT) and a probe antenna in their near field is measured. In near field, the radiation characteristics of both depend on the measurement distance between them, since the radiated waves are spherical. In real applications, both antennas are normally in their so-called far field, which means that a far-field approximation can be used, resulting in assuming the radius of the radiated spherical waves is so large that they are perceived as a plane wave. The implication is that, once a distance that allows for this approximation is reached, it can be assumed that the radiation characteristics of both antennas do not depend on the distance anymore.

The approach used to retrieve the far-field radiation characteristics of an AUT from its interaction with a probe in their near field is modeling the interaction as a multipole expansion, under the assumption that the probe response coefficients be known. The objective is to retrieve the spherical mode coefficients (SMCs) corresponding to the AUT, and once this is done, its far-field characteristics can be calculated by computing the multipole expansion for a distance tending to infinity. In Fig. 16.1, a measurement situation with an AUT and its coordinate system, $(x, y, z)$, and a probe and its coordinate system in primed letter, $(x', y', z')$, is shown, together with the angles that relate the AUT and the probe. The general expression of the multipole expansion relating the AUT and the probe in this situation, frequently called the transmission formula, is given by [37]

C. Culotta-López (✉) · D. Heberling
Institute of High Frequency Technology, RWTH Aachen University, Aachen, Germany
e-mail: culotta@ihf.rwth-aachen.de; heberling@ihf.rwth-aachen.de

A. Bangun · R. Mathar
Institute for Theoretical Information Technology, RWTH Aachen University, Aachen, Germany
e-mail: arya.bangun@rwth-aachen.de; mathar@ti.rwth-aachen.de



G. Kutyniok et al. (eds.), *Compressed Sensing in Information Processing*, Applied and Numerical Harmonic Analysis, https://doi.org/10.1007/978-3-031-09745-4_16

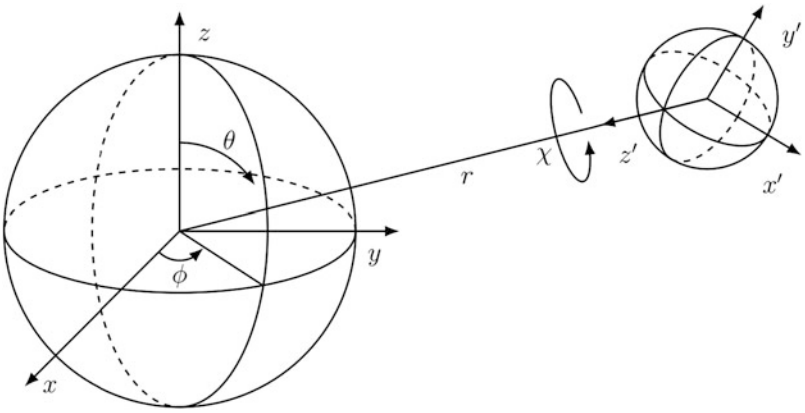



**Fig. 16.1** Relationship between the coordinate systems of the AUT and the probe

$$y(r, \chi, \theta, \phi) = \sum_{h=1}^{2} \sum_{n=-\nu_{\max}}^{\nu_{\max}} \sum_{l=1}^{\infty} \sum_{k=-l}^{l} T_{hlk} \mathrm{D}_l^{k,n}(\theta, \phi, \chi) P_{hnl}(k_w r), \tag{16.1}$$

where $y(r, \chi, \theta, \phi)$ is the measurement signal dependent on the distance $r$, polarization $\chi$, elevation $\theta$, and azimuth $\phi$ angles. The SMCs are given by $T_{hlk}$, and the function $\mathrm{D}_l^{k,n}(\theta, \phi, \chi)$ represents the Euler rotation of spherical waves, also called Wigner D-functions. In addition, $P_{hnl}(k_w r)$ are the probe response coefficients of the probe that is used to acquire near-field samples, with $n$ being the equivalent $l$ order of the probe and $n = \nu_{\max}$ the maximum order considered for it. The probe response coefficients $P_{hnl}(k_w r)$ are, in turn, derived from the SMCs of the probe measured at the center of the coordinate system of the measurement, rotated and translated to the measurement distance to reflect the measurement situation at hand, and can be expressed by

$$P_{hnl}(k_w r) = \sum_{\eta\nu} C_{\eta\nu l}^{hn(3)}(k_w r) R_{\eta\nu l}, \tag{16.2}$$

where the Greek indices represent their Latin counterpart for the original SMCs of the probe, $R_{\eta\nu l}$ are the original SMCs of the probe, and $C_{\eta\nu l}^{hn(3)}(k_w r)$ are the translation coefficients [37] that translate these to the measurement distance $r$, and the superscript (3) represents outward travelling waves.

At the same time, $h$, as well as its Greek counterpart $\eta$, are limited to the values 1 and 2, which represent the propagation of transverse electric and magnetic waves, respectively. Since antennas are band-limited, the summation over $l$ can be truncated to the band-limit constant $B$, so that we have a degree $1 \leq l \leq B$. The band-limit constant is defined according to

$$B = \lfloor k_w r_0 \rfloor + L_0, \tag{16.3}$$

where $k_w$ is, again, the wavenumber, $r_0$ is the radius of the minimum sphere containing the AUT, the floor brackets signify the largest integer smaller than or equal to the product $k_w r_0$, and $L_0$ is a constant used for stability and accuracy. In the literature, the choice of $L_0 = 10$ is frequently used. We defer the construction of a linear system of equations of the transmission formula to Sect. 16.3.2.

### *16.1.1 Notation*

Throughout this chapter, we denote the vectors and matrices by lowercase and uppercase letters. The elevation, azimuth, and polarization angle are denoted by $\theta$, $\phi$, and $\chi$, respectively. The set $\{1, ..., m\}$ is denoted by $[m]$. $\bar{x}$ is the conjugate of $x$, and it is defined element-wise if applied to a vector. The inner product of two vectors $\mathbf{a}, \mathbf{b} \in \mathbb{C}^N$ is given by $\langle \mathbf{a}, \mathbf{b} \rangle := \sum_{i=1}^{N} a_i \bar{b}_i$. The notation $a \gtrsim b$ means that there is a universal constant $C$ such that $a \geq Cb$. The $\ell_p$-norm of a vector $\mathbf{x} \in \mathbb{C}^N$ is given by

$$\|\mathbf{x}\|_p := \begin{cases} \left( \sum_{i=1}^{N} |x_i|^p \right)^{1/p}, & 1 \leq p < \infty \\ \max\limits_{i \in [N]} |x_i|, & p = \infty. \end{cases} \tag{16.4}$$

The set $\{1, 2, \ldots\}$ of natural numbers is given by $\mathbb{N}$, whereas $\mathbb{N}_0$ includes $\{0\}$ as well.

## 16.2 Compressed Sensing

The classical concept of a system of equations is that of a set of $m$ linearly independent equations, i.e., equations that are not a linear combination of the others, involving the same set of $N$ variables. If the number of linearly independent equations matches the number of variables, i.e., $m = N$, the system has a unique solution. However, for the case where the equations are fewer than the number of variables, i.e., $m < N$, the system is underdetermined, and it may have infinitely many solutions, or no solution. Nevertheless, if many of the $N$ variables are 0 or have a neglectable influence on the system, it may be argued that the number of significant variables is $\tilde{s} < N$, and thus, $m = \tilde{s}$ equations should formally suffice to solve the system. With this knowledge, solving the system still has the obstacle of identifying which of the $\tilde{s}$ elements within the $N$ variables are significant. Correctly retrieving the solution in this case is called sparse retrieval. Let us assume we have a linear system of equations expressed as a matrix equation as follows:

$$\mathbf{y} = \mathbf{A}\mathbf{x}, \tag{16.5}$$

where $\mathbf{A} \in \mathbb{C}^{m \times N}$ is the matrix with $m$ row and $N$ column dimension over the complex fields. The linear inverse problem is the estimation of vector $\mathbf{x} \in \mathbb{C}^N$ given vector $\mathbf{y} \in \mathbb{C}^m$. It is well-known that for $m \geq N$ and a full-rank matrix $\mathbf{A}$, the recovery of vector $\mathbf{x}$ is unique. Nevertheless and as introduced earlier, this does not hold true for the case $m < N$, where this problem becomes ill-posed, which means the problem does not have a unique and stable solution. This situation is dramatically different once we know that vector $\mathbf{x}$ has at most $\tilde{s}$ non-zero elements. The $\ell_0$-norm [1] of a vector $\mathbf{x}$, $\|\mathbf{x}\|_0$, is the count of its non-zero elements. Formulating this as an optimization problem, we can write

$$\underset{x}{\text{minimize}} \quad \|\mathbf{x}\|_0 \quad \text{subject to} \quad \mathbf{y} = \mathbf{A}\mathbf{x}. \tag{P0}$$

The combinatorial nature of the problem makes it computationally intractable, since we have to consider all possible $\tilde{s}$-sparse vectors. Nevertheless, a convex relaxation of the problem can be used, which changes the objective functions into the $\ell_1$-norm, so that it becomes

$$\underset{x}{\text{minimize}} \quad \|\mathbf{x}\|_1 \quad \text{subject to} \quad \mathbf{y} = \mathbf{A}\mathbf{x}, \tag{P1}$$

which is a Basis Pursuit (BP) program [30]. Although this optimization problem has been used in the late 1970s [41, 45], stable and robust recovery guarantees for this algorithm have been derived, for the first time, in the seminal works [16, 30] in 2006. The book [32] provides a thorough treatment of these concepts. A prominent sufficient condition for recovery is the so-called restricted isometry property (RIP).

**Definition 16.1** A matrix $\mathbf{A}$ satisfies the RIP of order $\tilde{s}$ with constant $\delta \in (0, 1)$, if the following inequalities hold for all $\tilde{s}$-sparse vectors $\mathbf{x}$

$$(1 - \delta)\,\|\mathbf{x}\|_2^2 \leq \|\mathbf{A}\mathbf{x}\|_2^2 \leq (1 + \delta)\,\|\mathbf{x}\|_2^2 .$$

The smallest number $\delta$, denoted by $\delta_s$, is called the restricted isometry constant of $\mathbf{A}$.

Although the RIP would guarantee that solving (P1) is essentially akin to solving (P0), certifying the RIP proves to be NP-hard [5]. For this reason, another metric,

[1] The $\ell_0$-norm of a vector $\mathbf{x} \in \mathbb{C}^N$ is defined by

$$\|\mathbf{x}\|_0 := \sum_{i=1}^{n} 1(x_i \neq 0), \tag{16.6}$$

where $1(\cdot)$ is the identity function. The $\ell_0$-norm is called a norm just by convention, as it is not a norm in a classical sense.

the mutual coherence of the sampling matrix $\mathbf{A}$, can be used as measure for reconstructability.

**Definition 16.2** The mutual coherence of a matrix $\mathbf{A} = [\mathbf{a}_1 \dots \mathbf{a}_N] \in \mathbb{C}^{m\times N}$ is defined as the maximum of the normalized inner product of columns of the matrix, i.e.,

$$\mu(\mathbf{A}) := \max_{1\le i<j\le N} \frac{\left|\langle \mathbf{a}_i, \mathbf{a}_j\rangle\right|}{\|\mathbf{a}_i\|_2 \, \|\mathbf{a}_j\|_2}.$$

The mutual coherence is commonly used to assess the conditioning of deterministic matrices, and it is fundamentally related to the restricted isometry constants [32]. The mutual coherence is upper bounded by one and lower bounded by the Welch bound [52]: in practice, if the coherence of a sensing matrix is closer to the Welch bound, it will provide a better recovery guarantee. Even if the mutual coherence provides weaker reconstruction guarantees than the RIP, it is easily computable and, thus, practical for deterministic matrices.

## 16.3 Definition and Backgrounds

In this section, we provide a compact definition of signal processing on the sphere and the rotation group. For a complete and detailed introduction in this area, we refer the interested reader to [40].

### *16.3.1 Wigner D-Functions and Spherical Harmonics*

The rotation group SO(3) is a set of all possible rotations on the three-dimensional Euclidean space $\mathbb{R}^3$. This space can be parametrized by three rotation angles $\phi, \chi \in [0, 2\pi)$ and $\theta \in [0, \pi]$. In this chapter, the convention of elevation $\theta \in [0, \pi]$, azimuth, and polarization angles $\phi, \chi \in [0, 2\pi)$, as introduced in Fig. 16.1, is used. Suppose we have a square-integrable function on this space denoted by $f, g \in L^2(\mathrm{SO}(3))$. This space is indeed a Hilbert space with inner product given by

$$\langle f, g\rangle := \int_{\mathrm{SO}(3)} f(\theta, \phi, \chi)\overline{g(\theta, \phi, \chi)}\mathrm{d}\nu(\theta, \phi, \chi), \tag{16.7}$$

where $\mathrm{d}\nu(\theta, \phi, \chi) := \sin\theta \mathrm{d}\theta\mathrm{d}\phi\mathrm{d}\chi$. Wigner D-functions are the orthonormal basis functions for the Hilbert space $L^2(\mathrm{SO}(3))$, denoted by

$$\mathrm{D}_l^{k,n}(\theta, \phi, \chi) = N_l e^{-\mathrm{i}k\phi}\mathrm{d}_l^{k,n}(\cos\theta)e^{-\mathrm{i}n\chi}, \tag{16.8}$$

where $N_l = \sqrt{\frac{2l+1}{8\pi^2}}$ is the normalization factor, and $\mathrm{d}_l^{k,n}(\cos\theta)$ are the Wigner D-functions [2] of band-limit degree $0 \leq l \leq B-1$ and orders $-l \leq k, n \leq -l$. It should be noted that in SNF, formally, the trivial, continuous signal related to $l = 0$, i.e., $d_0^{k,n}$, is not considered to be part of the bandwidth, so $1 \leq l \leq B$, as defined in Sect. 16.1. However, for a purely mathematical derivation, the range $0 \leq l < B$ is considered.

Besides definition of the rotation group SO(3), we also provide the definition of unit sphere $\mathbb{S}^2$. Similar to the rotation group, we can also define a square-integrable function on this space. The only caveat is that, instead of having a parametrization of three angles, this space only has two parameter angles, namely $\theta \in [0, \pi]$ and $\phi \in [0, 2\pi)$. The orthonormal basis functions in this space are called spherical harmonics and can be generated by considering their relation between Wigner D-functions, which is

$$\mathrm{D}_l^{-k,0}(\theta, \phi, 0) = (-1)^k \sqrt{\frac{1}{2\pi}} \mathrm{Y}_l^k(\theta, \phi), \tag{16.9}$$

where the complex spherical harmonics can be written as

$$\mathrm{Y}_l^k(\theta, \phi) := N_l^k P_l^k(\cos\theta) e^{\mathrm{i}k\phi}.$$

Additionally, the definition of real spherical harmonics is given by

$$\mathrm{Y}_l^k(\theta, \phi) = \begin{cases} (-1)^k \sqrt{2} N_l^k P_l^k(\cos\theta) \sin(|k|\,\phi) & \text{if } k < 0 \\ N_l^0 P_l(\cos\theta) & \text{if } k = 0 \\ (-1)^k \sqrt{2} N_l^k P_l^k(\cos\theta) \cos(k\phi) & \text{if } k > 0 \end{cases}. \tag{16.10}$$

In both definitions, $P_l^k(\cos\theta)$ are the associated Legendre polynomials, and the term $N_l^k := \sqrt{\frac{2l+1}{4\pi} \frac{(l-k)!}{(l+k)!}}$ is a normalization factor.

### *16.3.2 Sparse Expansions of Band-Limited Functions*

As discussed in Sect. 16.1, since we can assume most antennas are band-limited, we can expand the electromagnetic fields radiated from an AUT with a band-limited expansion of Wigner D-functions. This can be written as

[2] With a lowercase or small d, to differentiate them from the Wigner D-functions $\mathrm{D}_l^{k,n}(\theta, \phi, \chi)$ and called, colloquially, "Wigner small-d functions."

$$g(\theta, \phi, \chi) = \sum_{l=0}^{B-1} \sum_{k=-l}^{l} \sum_{n=-l}^{l} \hat{g}_l^{k,n} \, \mathrm{D}_l^{k,n}(\theta, \phi, \chi). \tag{16.11}$$

As a consequence, a finite-dimensional vector of Fourier coefficients $\mathbf{g} = (\hat{g}_l^{k,n})_{0 \le l < B}$ with ambient dimension $N = \frac{B(2B-1)(2B+1)}{3}$ is obtained. A similar concept for band-limited functions on the unit sphere can be defined by setting the order $n = 0$ and the polarization angle $\chi = 0$. As a result, we also have finite-dimensional Fourier coefficients on the sphere with ambient dimension $N = B^2$.

The matrix $\mathbf{A}_{\mathrm{W}} \in \mathbb{C}^{m \times N}$ is the measurement or sensing matrix constructed from Wigner D-functions, which are the basis functions introduced in Eq. (16.1), and it is given by

$$\mathbf{A}_{\mathrm{W}} = \begin{pmatrix} \mathrm{D}_0^{0,0}(\theta_1, \phi_1, \chi_1) & \dots & \mathrm{D}_{B-1}^{B-1,B-1}(\theta_1, \phi_1, \chi_1) \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{D}_0^{0,0}(\theta_m, \phi_m, \chi_m) & \dots & \mathrm{D}_{B-1}^{B-1,B-1}(\theta_m, \phi_m, \chi_m) \end{pmatrix}, \tag{16.12}$$

with orders $\mathrm{D}_0^{0,0}, \mathrm{D}_1^{-1,-1}, \mathrm{D}_1^{-1,0}, \mathrm{D}_1^{-1,1}, \mathrm{D}_1^{0,-1}, \dots, \mathrm{D}_{B-1}^{B-1,B-3}, \mathrm{D}_{B-1}^{B-1,B-2}, \mathrm{D}_{B-1}^{B-1,B-1}$ in each row. This matrix is a collection of $m$ different samples of Wigner D-functions, where for each sample there exist Wigner D-functions related to its degree $l$ and order $|k|, |n| < B$. The column dimension can be calculated as $N = \frac{B(2B-1)(2B+1)}{3}$, which means that, for a single column $q \in [N]$, we have degree and orders dependent on $q$, i.e., $l(q)$, $k(q)$, and $n(q)$. This is different from the case portrayed by Eq. (16.1) and the column dimension $N$ calculated by Eq. (16.16), since, there, $n$ is limited to $n \in [-\nu_{\max}, \nu_{\max}]$, with $\nu_{\max} = 1$ for the vast majority of applications.

The construction of the sensing matrix from spherical harmonics $\mathbf{A}_{\mathrm{SH}}$ with band-limit degree $0 \le l \le B - 1$ and order $-l \le k \le l$ is expressed as follows:

$$\mathbf{A}_{\mathrm{SH}} = \begin{pmatrix} \mathrm{Y}_0^0(\theta_1, \phi_1) & \dots & \mathrm{Y}_{B-1}^{B-1}(\theta_1, \phi_1) \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{Y}_0^0(\theta_m, \phi_m) & \dots & \mathrm{Y}_{B-1}^{B-1}(\theta_m, \phi_m) \end{pmatrix}, \tag{16.13}$$

where each row follows the order $\mathrm{Y}_0^0, \mathrm{Y}_1^{-1}, \mathrm{Y}_1^0, \mathrm{Y}_1^1, \dots, \mathrm{Y}_{B-1}^{B-2}, \mathrm{Y}_{B-1}^{B-1}$. The structure of a sampling matrix $\mathbf{A} \in \mathbb{C}^{m \times N}$, be it $\mathbf{A}_{\mathrm{W}}$ or $\mathbf{A}_{\mathrm{SH}}$, is highly dependent on the sampling of the Wigner D-functions and spherical harmonics. In this chapter, the condition of RIP for both matrices and the construction of low-coherence deterministic matrices are investigated.

The sampled version of band-limited expansion can be formulated in terms of a system of linear equations

$$\mathbf{y} = \mathbf{A}\mathbf{x}, \tag{16.14}$$

where the matrix $\mathbf{A} \in \mathbb{C}^{m \times N}$ is constructed from sampled Wigner D-functions, and the vector $\mathbf{y}$ consists of the acquired electromagnetic fields, i.e.,

$$\mathbf{y} = \begin{pmatrix} g\left(\theta_1, \phi_1, \chi_1\right) \\ \vdots \\ g\left(\theta_m, \phi_m, \chi_m\right) \end{pmatrix}. \tag{16.15}$$

The vector of coefficients $\mathbf{x} \in \mathbb{C}^N$ is constructed from Wigner coefficients $\hat{g}_{l(q)}^{k(q),n(q)}$. For SNF as described in the transmission formula (16.1), it is usually assumed that the first-order modes of the probe are the ones with the most power, whereas the rest are negligible. This translates into $\nu_{\max} = 1$, which, in the classical SNF literature and following its usual notation, is referred to as "$\mu = \pm 1$ probes" [37]. The column dimension $N$ of this matrix depends on the band-limit constant $B$ and writes

$$N = 2B(B+2) = 2B^2 + 4B, \tag{16.16}$$

slightly different defined as general expansion of Wigner D-function, where we have $N = \frac{B(2B-1)(2B+1)}{3}$ by taking the whole combination of degree $l$. The row dimension $m$ is defined by the number of sampled points. The vectors $\mathbf{y} \in \mathbb{C}^m$ and $\mathbf{x} \in \mathbb{C}^N$ represent finite samples of near field and the SMCs. From this property, a total of $m = N$ measurements suffice to solve the linear equation system assuming the matrix $\mathbf{A}$ is well-conditioned. However, for the classically used equiangular sampling pattern, the total number of measurements required is larger than twice the ambient dimension $N$:

$$m = 2(B+1)(2B+1) > 2N. \tag{16.17}$$

However, the SMCs are compressible, which means only a small part of these coefficients have a high intensity, as given in Fig. 16.2. This phenomenon leads to the fundamental question of whether it is possible to reduce the number of measurements scaling only with the number of significant coefficients. This problem

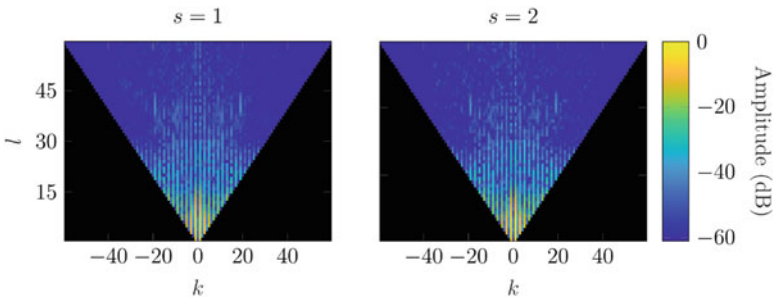



**Fig. 16.2** SMCs (left: transverse electric, right: transverse magnetic)

is crucial to be addressed in this chapter by using tools from compressed sensing (CS), as discussed in the next section.

### *16.3.3 Construction of the Sensing Matrix*

#### 16.3.3.1 RIP Condition for Sensing Matrices

Uniformly distributed random samples on the sphere are highly connected to the uniform probability measure on the sphere $\mathrm{d}\nu := \sin\theta \mathrm{d}\theta\mathrm{d}\phi$. As discussed in [15], taking this random sampling to construct a sensing matrix from spherical harmonics is proven to satisfy the RIP condition, with the number of measurements scaling as $m \gtrsim N^{1/2}\tilde{s}\log^4(N)$. However, this bound is not optimal because of the scaling factor $N^{1/2}$. The scaling with $N$ can be further improved to $N^{1/4}$ by changing the probability measure to $\mathrm{d}\theta\mathrm{d}\phi$ and including $\sin^{1/2}(\theta)$ as preconditioning [43]. An additional improvement to $N^{1/6}$ is possible by sampling according to $|\tan\theta|^{1/3}\mathrm{d}\theta\mathrm{d}\phi$ and using $(\sin^2\theta\cos^2\theta)^{1/6}$ as preconditioned, see also [15]. Considering these different measures will affect the distribution of sampling points on the spherical surface as depicted in Fig. 16.3. It can be seen that, to have uniformly distributed random samples on the sphere, a distribution with respect to the measure $\sin(\theta)\mathrm{d}\theta\mathrm{d}\phi$ shall be considered. On the contrary, taking uniformly random samples directly from $\theta \in [0, \pi]$ and $\phi \in [0, 2\pi)$ leads to a concentration of points at the spherical poles, as well as to a concentration on the equator if random samples with respect to $|\tan(\theta)|^{\frac{1}{3}}\,\mathrm{d}\theta\mathrm{d}\phi$ are taken. This strategy can be tailored to construct a matrix from sampled Wigner D-functions. This result is given in the following theorem [6, Theorem 10], [10, Theorem 3].

**Theorem 16.1** *Consider the problem of finding sparse Fourier coefficients* $\mathbf{g}$ *of a band-limited function* $g \in L^2(\mathrm{SO}(3))$ *from noisy linear measurements* $\mathbf{y} = \mathbf{A}\mathbf{g} + \boldsymbol{\eta}$ *with* $\|\boldsymbol{\eta}\|_\infty \leq \epsilon$. *Suppose that the sensing matrix* $\mathbf{A}$ *is constructed as* (16.12) *using* $m$ *i.i.d. samples* $(\theta_p, \phi_p, \chi_p)$, $p \in [m]$ *drawn uniformly from* $[0, \pi] \times [0, 2\pi] \times [0, 2\pi]$. *Let* $\mathbf{P} \in \mathbb{R}^{m\times m}$ *be a diagonal matrix with* $P_{ii} = \sin(\theta_i)^{1/2}$. *The number of measurements* $m$ *is assumed to satisfy the following inequality:*

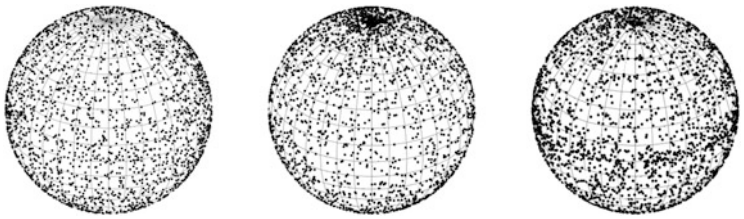


**Fig. 16.3** Distribution of random sampling (left: $\sin(\theta)\mathrm{d}\theta\mathrm{d}\phi$, middle: $\mathrm{d}\theta\mathrm{d}\phi$, right: $|\tan(\theta)|^{1/3}\,\mathrm{d}\theta\mathrm{d}\phi$)

$$m \gtrsim N^{1/6}\, \tilde{s}\, \log^3(\tilde{s})\, \log(N).$$

*Then with probability of at least* $1 - N^{-\gamma \log^3(\tilde{s})}$*, for universal constant* $\gamma \geq 0$*, the following holds. If* $\mathbf{g}^{\#}$ *is the solution to the following problem:*

$$\mathbf{g}^{\#} = \arg\min \|\mathbf{z}\|_1 \text{ subject to } \|\mathbf{PAz} - \mathbf{Py}\|_2 \leq \sqrt{m}\epsilon.$$

*then*

$$\left\|\mathbf{g} - \mathbf{g}^{\#}\right\|_2 \lesssim \frac{\sigma_{\tilde{s}}(\mathbf{g})_1}{\sqrt{\tilde{s}}} + \epsilon,$$

*where the best* $\tilde{s}$*-sparse approximation of* $\mathbf{g}$ *is expressed as*

$$\sigma_{\tilde{s}}(\mathbf{g})_p = \min_{\hat{\mathbf{g}} \in \mathbb{C}^N : \|\hat{\mathbf{g}}\|_0 \leq \tilde{s}} \left\|\hat{\mathbf{g}} - \mathbf{g}\right\|_p.$$

*In particular, when the measurements are not noisy, the recovery is unique for an* $\tilde{s}$*-sparse signal, namely* $\mathbf{g} = \mathbf{g}^{\#}$*.*

Although using random samples to prove the RIP condition is interesting from a theoretical perspective, the design of deterministic sampling patterns is more interesting for real application, especially for SNF. Due to the mechanical nature of the problem, i.e., the need of mechanically rotating an AUT to acquire all defined points, a random sampling scheme may reduce the number of samples in comparison to other schemes, but still require a long measurement time. However and as mentioned previously, certifying the RIP condition for deterministic matrices is hard [5, 49], so the (low) coherence is used instead as a measure for reconstructability. In the next section, a sampling pattern that yields a low-coherence matrix with the inclusion of a constraint to enable fast acquisition is proposed.

#### 16.3.3.2 Construction of Low-Coherence Sensing Matrices

Suppose a matrix is constructed from spherical harmonics and Wigner D-functions as given in (16.13) and (16.12), respectively. The mutual coherence expressions for spherical harmonics, $\mu(\mathbf{A}_{\mathrm{SH}})$, and Wigner D-functions, $\mu(\mathbf{A}_{\mathrm{W}})$, are given by [10]

$$\mu(\mathbf{A}_{\mathrm{SH}}) := \max_{1 \leq r < q \leq N} \left| \sum_{p=1}^{m} \frac{\mathrm{Y}_{l(q)}^{k(q)}(\theta_p, \phi_p)\, \overline{\mathrm{Y}_{l(r)}^{k(r)}(\theta_p, \phi_p)}}{\left\|\mathrm{Y}_{l(q)}^{k(q)}(\boldsymbol{\theta}, \boldsymbol{\phi})\right\|_2 \left\|\mathrm{Y}_{l(r)}^{k(r)}(\boldsymbol{\theta}, \boldsymbol{\phi})\right\|_2} \right| \tag{16.18}$$

$$\mu(\mathbf{A}_{\mathrm{W}}) := \max_{1\le r<q\le N} \left| \sum_{p=1}^{m} \frac{\mathrm{D}_{l(q)}^{k(q),n(q)}(\theta_p,\phi_p,\chi_p)\overline{\mathrm{D}_{l(r)}^{k(r),n(r)}(\theta_p,\phi_p,\chi_p)}}{\left\|\mathrm{D}_{l(q)}^{k(q),n(q)}(\boldsymbol{\theta},\boldsymbol{\phi},\boldsymbol{\chi})\right\|_2 \left\|\mathrm{D}_{l(r)}^{k(r),n(r)}(\boldsymbol{\theta},\boldsymbol{\phi},\boldsymbol{\chi})\right\|_2} \right|, \tag{16.19}$$

where the following convention is adopted:

$$\mathrm{Y}_l^k(\boldsymbol{\theta},\boldsymbol{\phi}) := \begin{pmatrix} \mathrm{Y}_l^k(\theta_1,\phi_1) \\ \vdots \\ \mathrm{Y}_l^k(\theta_m,\phi_m). \end{pmatrix} \quad \mathrm{D}_l^{k,n}(\boldsymbol{\theta},\boldsymbol{\phi},\boldsymbol{\chi}) := \begin{pmatrix} \mathrm{D}_l^{k,n}(\theta_1,\phi_1,\chi_1) \\ \vdots \\ \mathrm{D}_l^{k,n}(\theta_m,\phi_m,\chi_m) \end{pmatrix}. \tag{16.20}$$

In most cases, equiangular sampling patterns are used for their ease of implementation. For example, the classical SNF methods utilize this type of sampling patterns to estimate the SMCs. Their convenience is not limited to the implementation of the sampling itself but also has practical reasons with regard to the solver to estimate the SMCs, which then allows the implementation of fast Fourier transforms for processing. Regardless of their popularity, using these sampling patterns to construct sensing matrices from spherical harmonics and Wigner D-functions produces maximum coherence, i.e., it has a bad reconstruction guarantee, as discussed in [7, Theorem 3], [10, Theorem 4].

**Theorem 16.2** *Let the matrix* $\mathbf{A} \in \mathbb{C}^{m\times N}$ *be constructed from either samples of spherical harmonics* $\mathrm{Y}_l^k(\theta,\phi)$ *or Wigner D-functions* $\mathrm{D}_l^{k,n}(\theta,\phi,\chi)$ *for a signal with bandwidth B using a sampling pattern that satisfies*

$$\begin{aligned} 2k\phi_i &\equiv 2k\phi_j \mod 2\pi, && \forall i,j \in [m] \\ 2n\chi_i + 2k\phi_i &\equiv 2n\chi_j + 2k\phi_j \mod 2\pi, && \forall i,j \in [m] \end{aligned}$$

*for some* $-(B-1) \le k,n \le B-1$. *Then the mutual coherence of this matrix attains its maximum, i.e.,* $\mu(\mathbf{A}) = 1$.

The equiangular sampling pattern, shown in Fig. 16.4a, belongs to the class mentioned in the previous theorem. In the numerical evaluation, it is shown that the coherence of the spherical harmonics sensing matrix as well as Wigner D-functions is equal to one when applying this sampling pattern. Thus, the design of low-coherence sampling matrices is necessary for compressed-sensing applications. This problem is equal to finding the sequence of angles $\theta_p \in [0,\pi]$, $\phi_p \in [0,2\pi]$, $\chi_p \in [0,2\pi]$ for $p \in [m]$ that minimizes the coherence. Minimizing the coherence of matrices from spherical harmonics and Wigner D-functions is a non-trivial problem because of the non-convexity of associated Legendre, Jacobi, and also trigonometric polynomials. However, and since certifying the RIP is hard, a minimal coherence is a practical guarantee for reconstructability. It is possible to describe a coherence bound considering equispaced sampling of $\theta$ as defined in [7, Theorem 4], [10, Proposition 2], [11, Theorem 1], as follows:

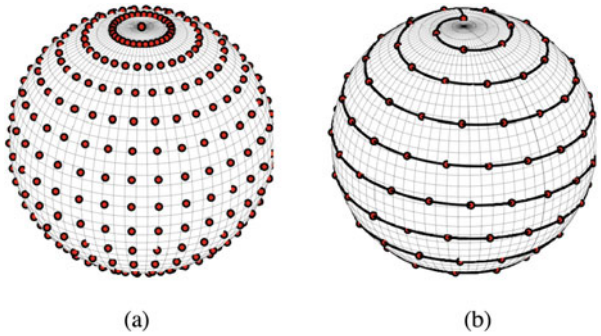



**Fig. 16.4** (**a**) Equiangular and (**b**) spiral sampling schemes

**Theorem 16.3** *For symmetric and equispaced sampling pattern with* $\cos\theta_p = \frac{2p-m-1}{m-1}$, $p \in [m]$, *the coherence of corresponding sensing matrices from spherical harmonics and Wigner D-functions are lower bounded by*

$$\begin{aligned}\mu(\mathrm{A}) &= \max_{q\neq r} \left| \sum_{p=1}^{m} \frac{\mathrm{Y}_{l^{(q)}}^{k^{(q)}}(\theta_p,\phi_p)\overline{\mathrm{Y}_{l^{(r)}}^{k^{(r)}}(\theta_p,\phi_p)}}{\left\|\mathrm{Y}_{l^{(q)}}^{k^{(q)}}(\theta,\phi)\right\|_2 \left\|\mathrm{Y}_{l^{(r)}}^{k^{(r)}}(\theta,\phi)\right\|_2} \right|, \\ &\geq \left| \sum_{p=1}^{M} \frac{P_{B-1}(\cos\theta_p)\,P_{B-3}(\cos\theta_p)}{\|P_{B-1}(\cos\boldsymbol{\theta})\|_2\,\|P_{B-3}(\cos\boldsymbol{\theta})\|_2} \right|\end{aligned} \tag{16.21}$$

*where* $P_l(\cos\theta)$ *is the Legendre polynomial of degree* $l \in \{0,\ldots,B-1\}$.[3]

As observed in [7, 10], by using a simple heuristic approach, there exists a sequence of $\phi_p \in [0, 2\pi)$ for $p \in [m]$ that achieves the lower bound in Theorem 16.3, which implies that Eq. (16.21) becomes an equality. Pseudocode for constructing such a sequence is presented in Algorithm 1. In the next section, we will draw numerical results to show the performance of this proposed sampling pattern.

### *16.3.4 Numerical Evaluation*

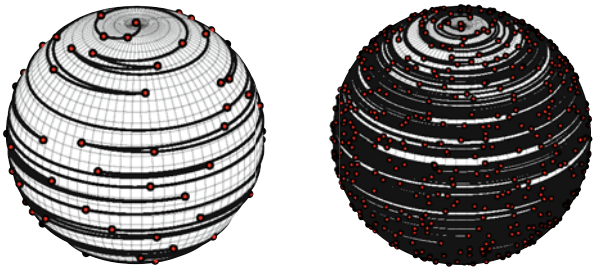
In this section, the coherence of matrices from spherical harmonics and Wigner D-functions is evaluated, as discussed in [10]. Besides the proposed sampling pattern,

[3] As explained previously, for the precise case of SNF, the range $1 \leq l \leq B$ is taken, since the trivial continuous signal $P_0(\cos\theta)$ is not considered for the bandwidth. Under this assumption, the degrees of the associated Legendre polynomial for the bound become $B$ and $B-2$, respectively.

**Algorithm 6:** Pattern search

**input** : $\boldsymbol{\theta}$ given, $\boldsymbol{\phi}_0 \in \mathbb{R}^m$ as initial points, $\Delta_0 > 0$ as initial update step,
: standard basis $\boldsymbol{e}_i$ for $i \in [m]$, $\lambda \in (0, 1)$.

**for** $k = 0, 1, \ldots, k_{\max}$ *until* $\left|\mu(\boldsymbol{\theta}, \boldsymbol{\phi}_k) - \mu_{\mathrm{LB}}\right| \leq \epsilon$ **do**
  **if** $\mu(\boldsymbol{\theta}, \boldsymbol{x}) < \mu(\boldsymbol{\theta}, \boldsymbol{\phi}_k)$ *for* $\boldsymbol{x} \in S_k := \{\boldsymbol{\phi}_k \pm \Delta_k \boldsymbol{e}_i\}$ **then**
    $\boldsymbol{\phi}_{k+1} = \boldsymbol{x} \mod 2\pi$
    $\Delta_{k+1} = \Delta_k$
  **else**
    $\boldsymbol{\phi}_{k+1} = \boldsymbol{\phi}_k \mod 2\pi$
    $\Delta_{k+1} = \lambda \Delta_k$
  **end**
**end**



(a) (b)

**Fig. 16.5** Proposed sampling path for the proposed scheme. (**a**) $m = 97$. (**b**) $m = 800$

several well-known sampling patterns on the sphere and the rotation group, as discussed in [38], will also be provided. Additionally, the recovery results in terms of phase transition diagrams as well as the implementation in spherical near-field antenna measurements are presented, where the BP programs (P1) YALL1 [54] and SPGL1 [50] will be used. The distribution of the proposed sampling pattern and sampling path for near-field measurements, as given in [25, 26], is shown in Fig. 16.5.

#### 16.3.4.1 Coherence Evaluation

In this setting, the sensing matrix from spherical harmonics with a column dimension $N = 100$ will be considered. In Fig. 16.6a, the mutual coherence of sampling matrices from Wigner D-functions is evaluated for different sampling strategies: the equiangular sampling, discussed previously; the proposed sampling scheme, and the spiral sampling scheme [12, 44], shown in Fig. 16.4b. The spiral sampling scheme is included for being often researched for accelerating SNF measurements [14], sometimes in combination with the application of compressed sensing [39]. It can be seen that, among all three sampling patterns, the equiangular sampling

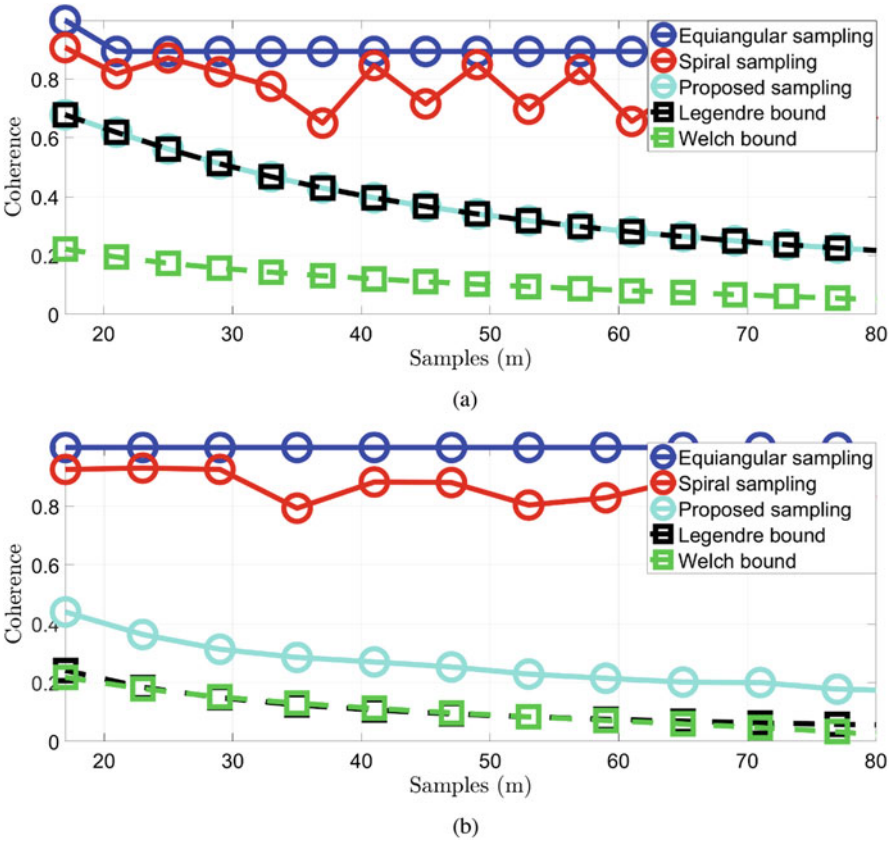



**Fig. 16.6** Coherence evaluation of sensing matrices from both analyzed basis functions. (**a**) For spherical harmonics. (**b**) For Wigner D-functions

pattern delivers the worst performance as discussed in Theorem 16.2. From looking at Fig. 16.6, it is clear that for spherical harmonics, given an equispaced sampling on $\theta$, the construction of sampling pattern from Algorithm 1 yields sequences on $\phi$ that can reach the coherence bound in Theorem 16.3, labeled in the figure as Legendre bound. Along the same line, we can evaluate the coherence of the sensing matrix from Wigner D-functions with $N = 84$. Apart from the similar performance for equiangular sampling pattern, it is shown in Fig. 16.6b that the bound in Theorem 16.3 cannot be achieved for Wigner D-functions. Nevertheless, among all sampling patterns, the proposed sampling pattern still delivers a low-coherence sensing matrix.

#### 16.3.4.2 Phase Transition Diagram: Random vs. Deterministic

In this section, the sparse recovery performances of different sensing matrices, including the proposed sampling pattern, are evaluated. Phase transition diagrams visualize the performance of successful and failed recoveries, which are represented by white and black colors. The abscissa, $m/N$, represents the ratio between the number of samples considered $m$ and the total number of variables $N$, while the ordinate axis, $\tilde{s}/m$, represents the ratio between the number of non-zero variables and the number of samples considered. The sparse coefficients are generated according to a zero-mean and unit-variance complex Gaussian distribution. For each parameter, we consider Monte Carlo (MC) simulations with 50 trials, for which any random quantities are redrawn independently. The recovery is classified as successful if the following holds:

$$\frac{\left\|\mathbf{x}-\hat{\mathbf{x}}\right\|_2}{\|\mathbf{x}\|_2} \leq \varepsilon, \tag{16.22}$$

where $\mathbf{x}, \hat{\mathbf{x}} \in \mathbb{C}^N$ are the original and its estimated signal, respectively. The threshold is given by $\varepsilon = 10^{-3}$, so that it relates to an error of $\varepsilon_{\mathrm{dB}} = -60\mathrm{dB}$. The performance of several sampling schemes is assessed: the classical equiangular sampling, the spiral sampling, the proposed sampling, and two definitions of the random sampling scheme, discussed in Sect. 16.3.3.1. Here, the random sampling schemes related to $\mathrm{d}\theta\mathrm{d}\phi$ and $|\tan\theta|^{1/3}\,\mathrm{d}\theta\mathrm{d}\phi\mathrm{d}\chi$ are denoted as Random 1 and Random 2, respectively. The background reflects the probability of success of the best case, while the transition bounds for each case are drawn for a probability of success of 0.5, i.e., 50%. For both cases, it can be seen from Fig. 16.7a and b that the worst performance, as expected, occurs when constructing a matrix from equiangular sampling patterns. Thus, this type of sampling pattern is inappropriate to be used for sparse recovery. Numerically, we show that the construction of a low-coherence deterministic sensing matrix has a performance comparable to the one of a sensing matrix from random samples. Although, in theory, the construction of a sensing matrix with low coherence presents a pessimistic recovery guarantee compared to satisfying the RIP condition, in this specific case for spherical harmonics and Wigner D-functions we can have a similar performance [29, 39].

### *16.3.5 Implementation in Spherical Near-Field Antenna Measurements*

In this section, the implementation of the proposed sampling pattern to real measurement data is discussed. As described in [25, 26], the double-ridged guide-horn antenna SAS-571 by AH System's [46] is used as AUT. The polarization angle $\chi$ is alternated between $\chi = 0°$ and $\chi = 90°$ for consecutive points in

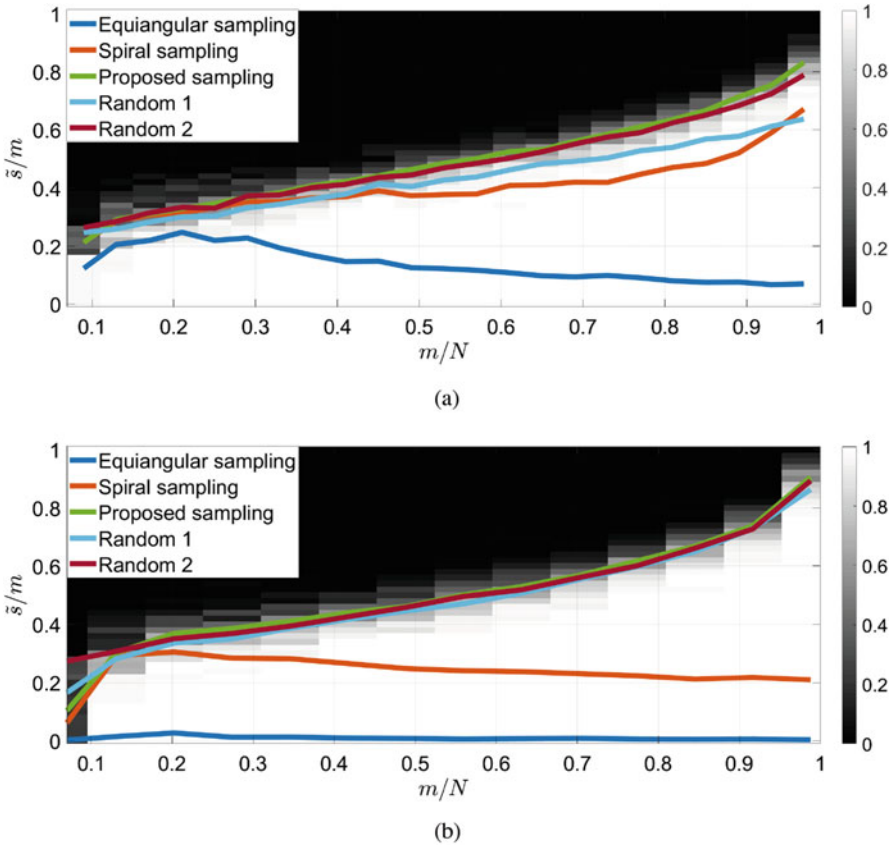



**Fig. 16.7** Phase transition diagrams of sensing matrices from both analyzed basis functions. (**a**) For spherical harmonics. (**b**) For Wigner D-functions

elevation $\theta$ [23, 25]. The operation frequency for all experiments is $f = 10$ GHz, and the original measurements are performed with a number of measurements $m = 14{,}280$, while the reconstruction by using CS is accomplished with $m = 5038$. In comparison, only 35.28% of the originally acquired measurement points are required for the shown reconstruction. The measurement situation, with the coordinate system of the AUT, the primed coordinate system of the probe, and the measurement angles, can be seen in Fig. 16.8.

The reconstruction of SMCs is performed by solving the BP program (P1) with the MOSEK solver [1–3]. Afterward, these coefficients are used to estimate the far-field radiation pattern, as shown in Fig. 16.9. The reference and reconstructed coefficients are shown in Fig. 16.10.

The evaluation of the far field is performed by comparing the original, calculated from a classical measurement, with the reconstruction obtained with the proposed strategy. This is done for the $\phi = 0°$ and $\phi = 45°$ cuts and both the co-polar (Co-

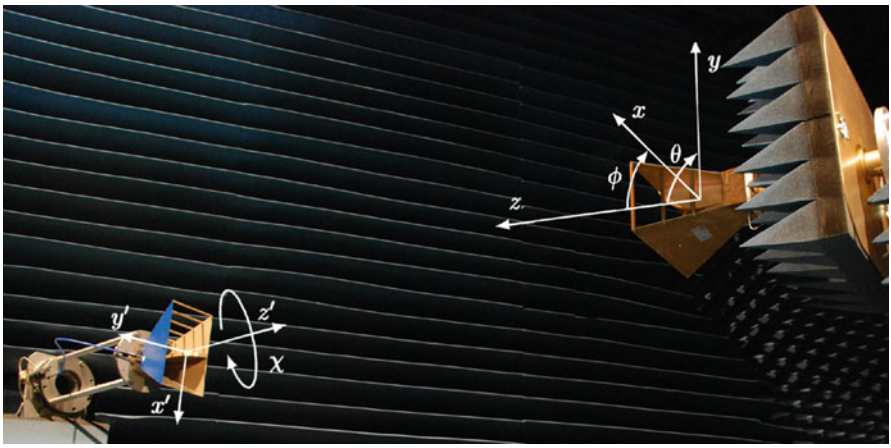



**Fig. 16.8** Measurement situation. The primed coordinate system corresponds to the probe

Pol) and the cross-polar (Cx-Pol) components of the AUT's radiation pattern. The equivalent error signal (EES) is shown as well, calculated by [23]

$$\mathrm{EES} = 20 \log_{10} \big| |E_{\mathrm{ref,\,norm}}| - |E_{\mathrm{rec,\,norm}}| \big|, \tag{16.23}$$

where $E_{\mathrm{ref,\,norm}}$ and $E_{\mathrm{rec,\,norm}}$ are the reference and reconstructed patterns normalized to their maximum, respectively. The reconstruction shows good agreement, considering that the simulations are carried out with a number of measurement points of around 3 times lower than for equiangular sampling.

#### 16.3.5.1 Modifying the Scheme for Time Efficiency

Acquiring fewer points does not directly relate to a reduction of the measurement time, since this depends on the number of mechanical movements required [26]. The presented minimum-coherence sampling scheme, calculated to be equidistant in elevation, can be easily acquired with a conventional roll-over-azimuth positioner. As is the case for equiangular measurements, the roll axis, corresponding to $\phi$, can be set to work in continuous mode, i.e., without stop between consecutive samples. Likewise, the azimuth axis, corresponding to $\theta$, is set to work in step mode, i.e., stopping between consecutive samples. For the proposed sampling, more points on the step axis, $\theta$, are generally required, since there exist as many $\theta$ positions as points $m$. This causes the acquisition to be slower for schemes with comparatively large values of $m$. Taking advantage of the rotations of the continuous axis, $\phi$, it is possible to increase the information content with no time cost by adding sampling points to the sampling path [23, 26]. To keep the equiangular sampling as comparison standard, the newly introduced sampling points are taken at the same

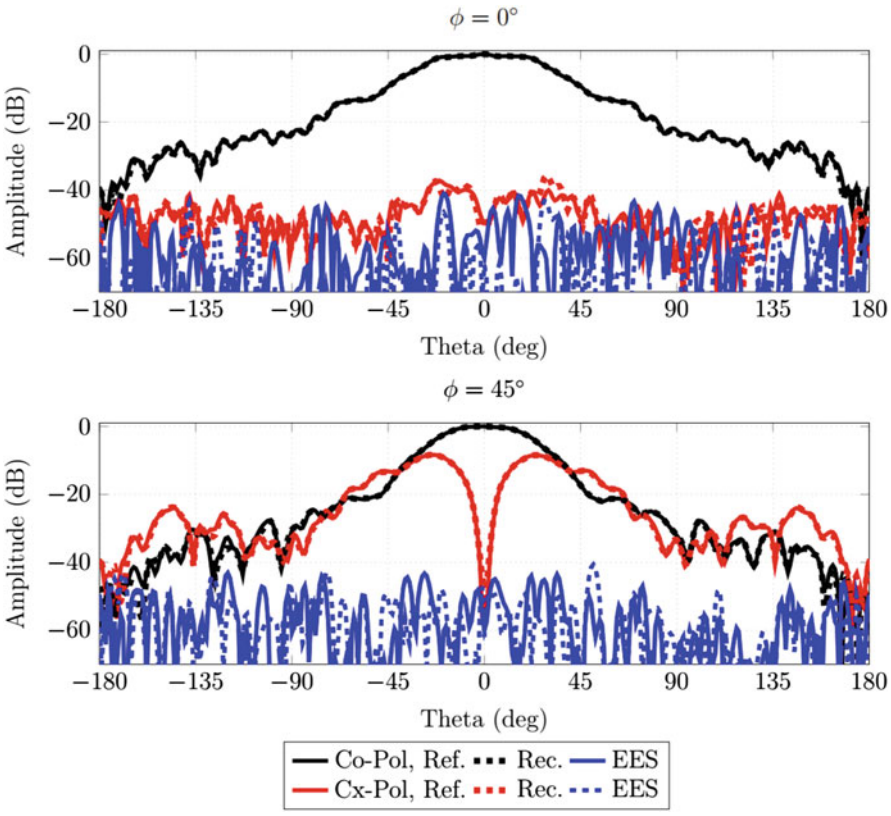



**Fig. 16.9** SAS-571's original normalized far-field radiation pattern and the pattern reconstructed with the proposed method for $\phi = 0°$ and $\phi = 45°$ for a number of samples that amounts to 35.28% of the original

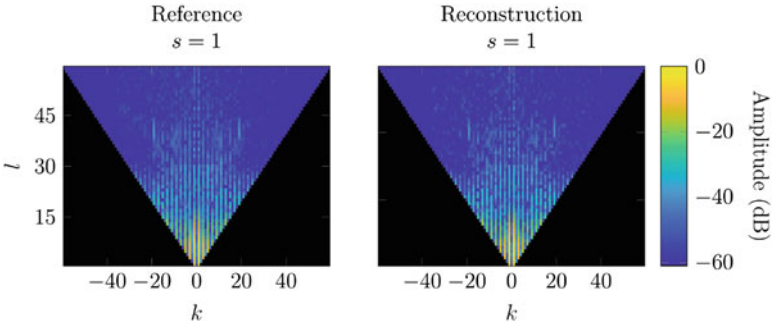



**Fig. 16.10** Normalized SMCs of the SAS-571 double-ridged guide-horn antenna

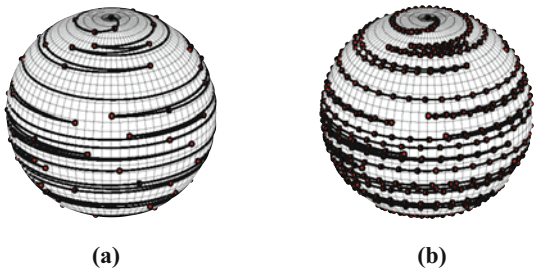



**Fig. 16.11** Proposed scheme and its modification for $m = 97$ and $m_g = 97$, respectively. (**a**) Minimum coherence. (**b**) Modified sampling

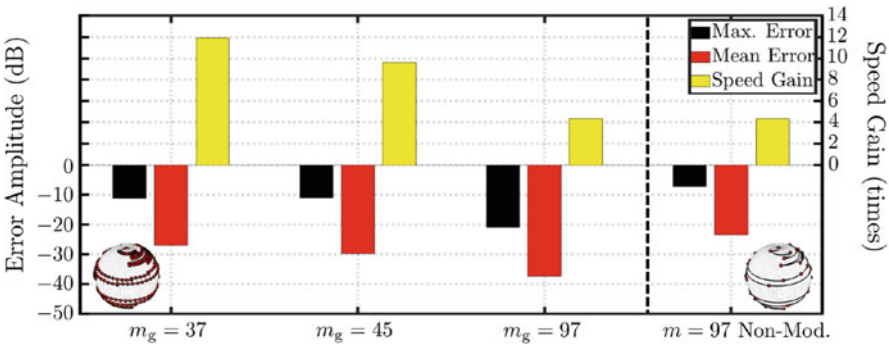



**Fig. 16.12** Maximum and mean errors and speed gain in terms of times the speed of equiangular

angular distance the equiangular sampling is acquired with. A comparison between the proposed sampling, used in the previous section, and the modified sampling with additional points on the path is shown in Fig. 16.11.

The modified scheme based on a minimum-coherence compressed sampling with $m$ number of sampling points is said to have a base number of *ground* sampling points $m_g$. It is shown that this scheme does reduce the reconstruction error without increasing the measurement time. This modification to the sampling scheme allows measurements with lower $m_g$ schemes to deliver a reconstruction error normally only achievable by choosing larger values of $m$. Provided fixed reconstruction error requirements, the proposed modification allows for faster measurements.

The same antenna used for previous simulations, AH Systems' SAS-571 [46], is used to test this concept. The measurement frequency is $f = 10$ GHz. The measurement step for a measurement with an equiangular measurement scheme is $\Delta\theta = \Delta\phi = 3\,^\circ$. The numerical experiments are performed for modified sampling schemes with $m_g = 37$, $m_g = 45$, and $m_g = 97$. In Fig. 16.12, the maximum and average error retrieved from these reconstructions is shown and compared to a reconstruction with $m = 97$, i.e., without additional points acquired on the sampling path. The estimated speed gain is also shown, calculated as *times faster than equiangular sampling*, where a value $C$ represents a measurement time of $t_{CS} = {}^{t_{eq}}/_{C}$.

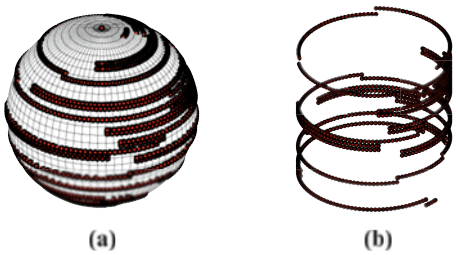



**Fig. 16.13** Projection of a compressed sampling scheme onto a cylindrical surface. (**a**) Sphere. (**b**) Cylinder

#### 16.3.5.2 Extension to Arbitrary Surfaces

The theory introduced until now has the constraint of only being valid for spherical surfaces, since the basis functions it is based on are defined for a spherical geometry. However, extending it to other surfaces, especially surfaces that are currently used in near-field facilities, would be advantageous for its application.

Equation (16.1) can be modified to accommodate a variable radius, $\rho > r_0$, $\rho \in \mathbb{R}^m$, which forces a different set of probe response coefficients $P_{hnl}(k_w \rho_j)$, $j \in [m]$ per measured point [21, 22], which is called pointwise probe correction. By radial projection of the $(r, \theta, \phi)$ points of a compressed sampling scheme onto an arbitrary surface defined in the spherical space $(\rho, \theta, \phi)$, pointwise probe correction enables the retrieval of the SMCs and, thus, the application of the investigated concepts on non-spherical geometries [23, 24]. To this end, a function $\rho(\theta, \phi)$ is required for the description of the chosen surface.

For the mentioned case of a cylindrical surface, results are promising despite truncation when the top and bottom are not measured, which is the typical application case of cylindrical near-field measurements. In Fig. 16.13, the projection of a compressed sampling scheme onto a cylinder is shown, whereas the reconstruction results for the $\phi = 0°$ and $\phi = 45°$ cuts, considering the same conditions as in the previous paragraph and a scheme with $m_g = 3B = 177$, are shown in Fig. 16.14, together with the EES, calculated as in Eq. (16.23). The vertical dashed lines define the truncation angle for the explored experiment.

#### 16.3.5.3 Implementation Considerations: Basis Mismatch

Compressed-sensing applications often suffer from specific problems, such as basis mismatch (BM) [20]. BM occurs when the measurement signal's basis functions matrix is not exactly $\mathbf{A}$ but $\hat{\mathbf{A}}$, though $\mathbf{A}$ is assumed for calculations. Since the measured signal is sparse only in $\mathbf{A}$, if the disagreement between both bases is high, the measured signal loses its compressibility and, thus, its reconstructability from compressed sampling schemes. Due to the finite nature of digital systems, an argument is made for BM always being present to some extent [20]. Formalizing the problem, let $\hat{\mathbf{y}}$ be a measurement signal sparse over the transformation matrix $\hat{\mathbf{A}}$ [23, 27], so that $\hat{\mathbf{y}} = \hat{\mathbf{A}}\mathbf{x}$. Assuming the basis function $\hat{\mathbf{A}}$ is unknown and $\mathbf{A}$

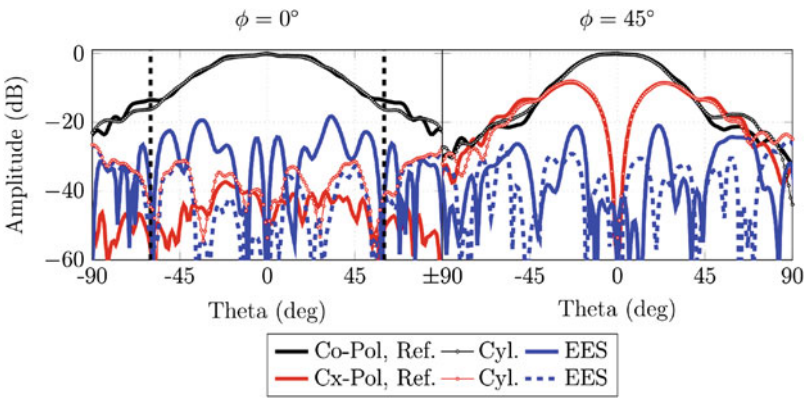



**Fig. 16.14** Simulated reconstruction for a compressed sampling projected onto a cylinder

is assumed instead delivers $\hat{\mathbf{y}} = \mathbf{A}\hat{\mathbf{x}}$, where the SMCs vector is not the original anymore, but a BM-distorted vector $\hat{\mathbf{x}}$. Solving this equation and inserting it in the previous equation yield

$$\hat{\mathbf{x}} = \mathbf{A}^{+}\hat{\mathbf{y}}, \quad \hat{\mathbf{x}} = \left(\mathbf{A}^{+}\hat{\mathbf{A}}\right)\mathbf{x} = \boldsymbol{\Delta}\mathbf{x}, \tag{16.24}$$

where $\mathbf{A}^{+}$ is the Moore–Penrose pseudoinverse of $\mathbf{A}$ and the matrix $\boldsymbol{\Delta} = (\mathbf{A}^{+}\hat{\mathbf{A}})$ is the BM factor between the true SMCs vector $\mathbf{x}$ and the reconstructed vector $\hat{\mathbf{x}}$. Ideally, $\hat{\mathbf{A}} = \mathbf{A}$ and $\boldsymbol{\Delta} = \mathbb{I}_N$. Several non-idealities can be modeled as BM, such as errors introduced by the characterization of the measurement probe, positioning errors, and aliasing introduced by the lack of redundancy due to undersampling. In [27], a compressed measurement is performed using NSI-MI Technologies' Robotic Antenna Measurement System (RAMS) in their facility. Although numerical simulations using the positioning uncertainties of RAMS show the impact thereof is limited, a measurement in an unshielded environment proves that the impact of aliasing in compressed measurements is high. Thus, a shielded environment is advisable for compressed measurements. The mismatch caused by the finite precision of the mechanical setup or by a (reasonable) error introduced by the operator setting up the measurement is proved to have a more limited impact [23].

## 16.4 Phaseless Spherical Near-Field Antenna Measurements

It has already been observed in [42, 47, 48, 53] that, when the phase measure is unreliable because of hardware defects, one can only rely on the magnitude measurements. For near-field measurements or more specifically SNF, the lack of

phase information creates ambiguities in the reconstruction of the SMCs. Besides the complete characterization of ambiguities that exist in phaseless measurements, several issues related to the number of measurements and suitable sampling patterns can be considered as object of investigation.

### *16.4.1 Phaseless Measurements*

Suppose we only have access to the phaseless measurements of near-field radiation. This problem can be expressed as follows:

$$\mathbf{y} = |\mathbf{A}\mathbf{x}| \,. \tag{16.25}$$

As for the classical problem in spherical near-field measurements, the goal is to estimate the coefficients $\mathbf{x} \in \mathbf{C}^N$ given phaseless measurements $\mathbf{y} \in \mathbf{R}^m$. The system is no longer linear, and phase ambiguities are introduced. The possible ambiguities for spherical harmonics are characterized and explained in the following section.

#### 16.4.1.1 Ambiguities in Phaseless Spherical Harmonics Expansion

Let us construct $\mathbf{A}_{\mathrm{SH}} = \{\mathbf{a_p}\}_{p\in[m]} \in \mathbb{C}^N$ from band-limited spherical harmonics as given in Eq. (16.13) with vector $\mathbf{x} \in \mathbb{C}^N$ containing spherical harmonic coefficients $\hat{f}_l^k$, i.e., $\mathbf{x} = \left(\hat{f}_0^0, \hat{f}_1^{-1}, \hat{f}_1^0 \ldots \hat{f}_{B-1}^{B-1}\right)^T$. The rotated coefficients $\mathbf{z} = \mathbf{x}e^{j\alpha} \in \mathbb{C}^N$ for $\alpha \in [0, 2\pi)$ and the reflected conjugate coefficients $\mathbf{z} = \overline{\mathbf{x}}$ deliver the same intensity measurement. The first property is a trivial implication of phaseless measurements. The second part follows from the property of conjugate spherical harmonics.

$$\overline{\mathrm{Y}_l^k(\theta, \phi)} = (-1)^k \mathrm{Y}_l^{-k}(\theta, \phi). \tag{16.26}$$

These coefficients $\mathbf{z}$ and $\overline{\mathbf{x}}$ are different in general for complex signals. Since the degree and order of the spherical harmonics are defined as $0 \leq l \leq B-1$ and $-l \leq k \leq l$, this conjugate symmetry produces an ambiguity between positive and negative orders. The last property does not exist when considering real spherical harmonics. Apart from phase ambiguities, the property of real spherical harmonics yields another type of ambiguity. For instance, if an inappropriate sampling pattern is considered, as discussed in the following result [8].

**Proposition 16.1 (Ambiguity-Incurring Sampling Patterns)** *Consider real spherical harmonics expansions of bandwidth B. Let the sampling points* $(\theta_p, \phi_p)$ *be chosen as* $(\theta_p, (B-2)\theta_p)$ *for* $p \in [m]$ *to construct a matrix of spherical harmonics* $\mathbf{A}_{SH} = [\mathbf{a}_1, \ldots, \mathbf{a}_m]^T \in \mathbb{R}^{m\times N}$ *as in* (16.13)*. Suppose that the elements of a vector of coefficients* $\mathbf{x} \in \mathbb{R}^N$ *are constructed from the following:*

$$\hat{f}_l^k = \begin{cases} c_l & k = 0,\ l+B \text{ is an odd number} \\ 0 & \text{otherwise} \end{cases}.$$

*Then there is a vector of coefficients* $\mathbf{y} \in \mathbb{C}^N$ *with single non-zero elements at degree and order* $l = k = 1$*, i.e.,* $\hat{g}_1^1 = d_1^1$ *such that for all* $p \in [m]$*:*

$$\mathbf{A}_{SH}\mathbf{x} = \mathbf{A}_{SH}\mathbf{y}$$
$$\sum_{l=0}^{B-1} N_l^0 P_l(\cos\theta_p) c_l = N_1^1 P_1^1(\cos\theta_p)\sin\big((B-2)\theta_p\big) d_1^1 .$$

*In other words, the products* $\mathbf{A}_{SH}\mathbf{x}$ *and* $\mathbf{A}_{SH}\mathbf{y}$ *cannot be distinguished using neither complete nor phaseless measurements.*

Since Proposition 16.1 shows that the solution for phase retrieval based on spherical harmonics is not unique, thus the problem seems ill-posed. Considering $\theta_p = \frac{(p-1)\pi}{m-1}$ for $p \in [m]$ and $B \geq 4$, it is interesting to see that the previous proposition is equal to equiangular sampling. In the numerical evaluation, we will consider this type of sampling patterns to represent Proposition 16.1.

### 16.4.2 Numerical Evaluation

Recently, many algorithms to recover coefficients $\mathbf{x}$ from phaseless measurements have been developed. In this chapter, the numerical experiments are performed by using algorithms in the PhasePack library [19] and with the semidefinite program using CVX [35, 36]. At its core, the semidefinite program considers squared measurements instead of absolute measurements as follows:

$$\mathrm{b}_p = \left|\langle \mathbf{a}_p, \mathbf{x}\rangle\right|^2 = \mathrm{trace}\left(\mathbf{a}_p\mathbf{a}_p^*\mathbf{X}\right) \quad \text{for} \quad p \in [m], \tag{16.27}$$

where $\mathbf{a}_p = [\mathrm{Y}_0^0(\theta_p, \phi_p), \ldots, \mathrm{Y}_{B-1}^{B-1}(\theta_p, \phi_p)] \in \mathbb{C}^N$ and the operator trace(.) takes the summation of diagonal square matrices. Note that the matrix $\mathbf{X} = \mathbf{x}\mathbf{x}^* \in \mathbb{C}^{N\times N}$ is a rank-1 and positive semidefinite matrix. Therefore, the optimization problem can be written as

$$\begin{aligned} &\text{find } \mathbf{X} \\ &\text{subject to } \mathrm{rank}\,(\mathbf{X}) = 1,\ \mathbf{X} \succeq 0,\ \mathrm{b}_p = \mathrm{trace}\left(\mathbf{a}_p\mathbf{a}_p^T\mathbf{X}\right)\ \forall p \in [m]. \end{aligned} \tag{16.28}$$

This optimization problem is non-convex; therefore, a convex relaxation based on semidefinite programming is used to solve it as suggested in [17]. This relaxation is called PhaseLift and is described by

$$\begin{aligned} &\text{find } \mathbf{X} \text{ that minimize trace } (\mathbf{X}) \\ &\text{subject to } \ \mathbf{X} \succeq 0, \ \ \mathrm{b}_p = \text{trace}\left(\mathbf{a}_p \mathbf{a}_p^T \mathbf{X}\right) \ \ \forall p \in [m]. \end{aligned} \tag{16.29}$$

The application of this and other methods to SNF measurements is investigated in [9]. Since a relaxed version is considered, it is necessary to analyze when and in which conditions the solutions of both problems are equivalent, i.e., when PhaseLift indeed solves Eq. (16.28). For sensing matrices constructed from a random normal distribution, the condition has been explained in [17]. However, the condition is totally different when considering structured matrices from spherical harmonics. Although this problem is still open in general, a number of algorithms for recovery are tested in the next section.

#### 16.4.2.1 Phase Transition Diagram

In this setting, the same well-known sampling patterns as we discussed in Sect. 16.4.2 from [38] and algorithms in [19] will be used. The Gerchberg–Saxton [33] and Fienup [31] algorithms use the alternating projection method [13]. Similar to the alternating projection method, the Kaczmarz [51] method is an iterative method that consists of projecting the estimation to a hyperplane determined by each row of a sensing matrix from spherical harmonics. In contrast to the alternating projection methods, PhaseLift [17], PhaseMax [34], and PhaseLamp [28] are convex optimization methods to solve the phase retrieval problem. While PhaseLift [17] works on the squared or intensity measurement, PhaseMax [34] and its optimization, PhaseLamp [28], rely only on the magnitude measurement directly. Wirtinger flow [18] is a gradient-based method, which consists of minimizing the loss function in terms of the mean squared error from the intensity measurements.

In Fig. 16.15, all algorithms fail to recover the correct signal for equiangular sampling, which confirms Proposition 16.1 [8]. Nevertheless, it can be seen that PhaseLift delivers a successful recovery from a smaller number of samples than the other algorithms. For this reason, a complete phase transition diagram for PhaseLift is provided, as shown in Fig. 16.16. Band-limit constant $B = \{4, 5, \ldots, 10\}$ and $N = B^2$ are assumed, and MC with 10 trials is performed in this setting. As derived in [4], the measurement bound for real measurements $m \geq 2N - 1$ seems sufficient to recover the coefficients by using PhaseLift.

#### 16.4.2.2 Implementation in Spherical Near-Field Antenna Measurements

The application of these concepts for phaseless SNF measurements is evaluated. An array of dipole antennas with ambient dimension of coefficients $N = 96$ is used as AUT. It can be seen from Fig. 16.17 that $m = 2.5N$ measurements are enough to recover SMCs by using PhaseLift, which is lower than the works in [42, 48], i.e.,

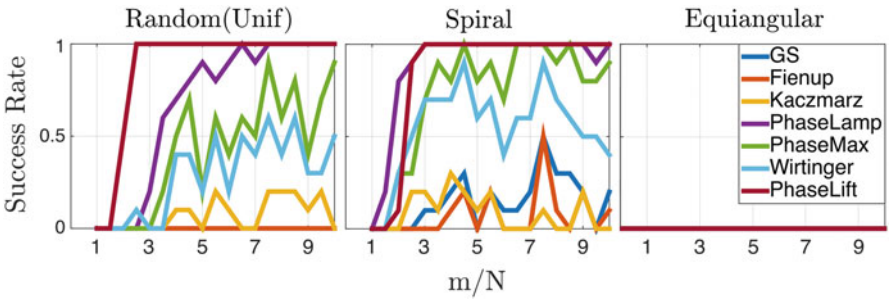



**Fig. 16.15** Phase transition of different algorithms

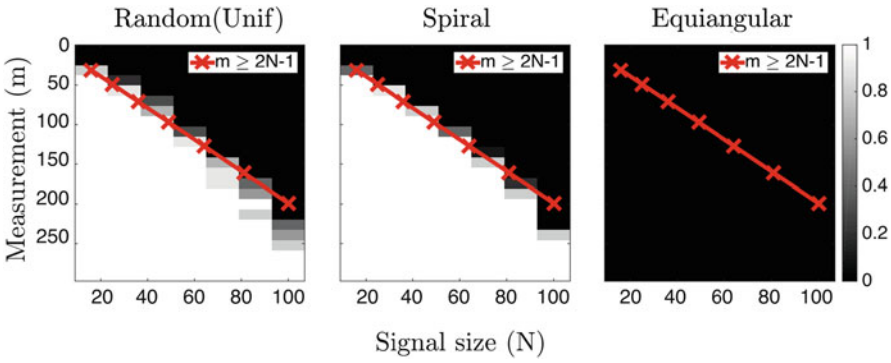



**Fig. 16.16** Phase transition of different sampling patterns with PhaseLift

$m > 4N$. Moreover, recovering the SMCs from phaseless near-field measurements is possible by considering the spiral and uniformly random samplings.

## 16.5 Summary

This chapter introduces a method to reduce the measurement time of spherical near-field antenna measurements by reducing the number of sampling points using compressed-sensing techniques. This is done starting from the theoretical conditions that allow for reconstruction from an undersampled set of random measurement points, followed by the practical approach of introducing a constraint in the sampling scheme to promote quick mechanical acquisition. Besides, it is studied whether compressed-sensing techniques can be used to the same end for phaseless spherical near-field antenna measurements.

A sampling strategy to construct a low-coherence sensing matrix from spherical harmonics and Wigner D-functions has been proposed. It is numerically shown that these sampling patterns deliver a better recovery performance than other well-

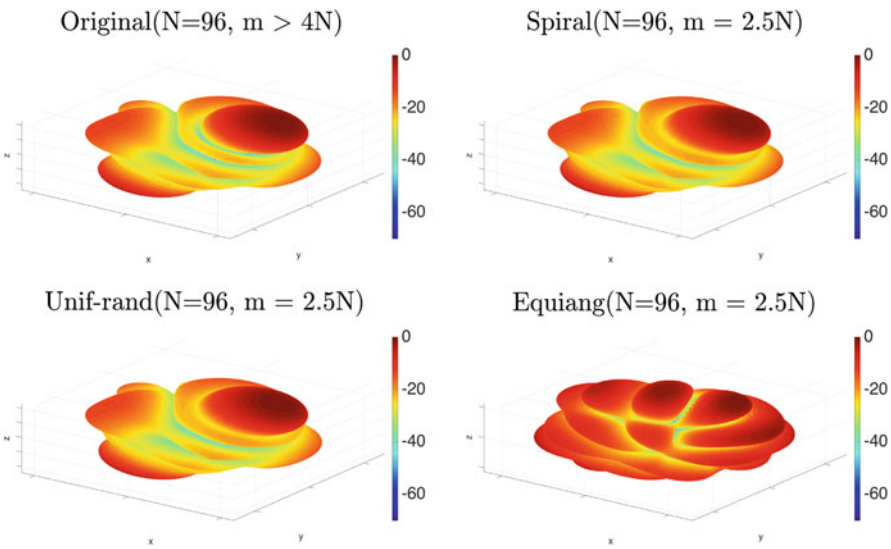



**Fig. 16.17** Far-field reconstruction from phaseless measurements

known sampling patterns. This sampling pattern also outperforms random sampling schemes in terms of the reconstruction of the spherical mode coefficients and far-field radiation pattern. The implementation of the proposed sampling pattern is verified, showing that the measurement time can be significantly reduced. Furthermore, by application of the technique known as pointwise probe correction, the derived theory can be extended to other arbitrary geometries. This enables, e.g., the application of the proposed sampling to the acquisition with systems optimized for geometries other than spherical, such as the cylindrical one.

However, compressed measurements suffer from additional problems due to the lack of redundancy of the measurements. Most additional effects can be modeled as contributing to basis mismatch, with the effect of aliasing being critical due to environmental reflections occurring outside of the measurement sphere. For adequate performance, measuring in a shielded environment is required.

A reduction in the number of measurement points for phaseless spherical near-field measurements, compared to the classical method, is shown feasible by measuring on a single surface. This is tested with classical, but undersampled, sampling schemes and existing algorithms for sparse phase retrieval from other fields. A lot of open questions related to this problem remain, such as the condition of the sensing matrix, the number of measurements to have recovery guarantee, as well as sampling strategies to construct the sensing matrices. Nevertheless, further research in the direction of phaseless spherical near-field antenna measurements seems promising.

**Acknowledgments** This work has been funded by the German Research Foundation (DFG, Deutsche Forschungsgemeinschaft) as project CoSSTra (HE 6073|8-1 and MA 1184|31-1).

## References


1. Andersen, E., Ye, Y.: A computational study of the homogeneous algorithm for large-scale convex optimization. Comput. Optim. Appl. **10**(3), 243–269 (1997). https://doi.org/10.1023/A:1018369223322
2. Andersen, E., Roos, C., Terlaky, T.: On implementing a primal-dual interior-point method for conic quadratic programming. Mathematical Programming **95**, 249–277 (2003). https://doi.org/10.1007/s10107-002-0349-3
3. ApS, M.: The MOSEK optimization toolbox for MATLAB manual. Version 9.0. (2019). http://docs.mosek.com/9.0/toolbox/index.html
4. Balan, R., Casazza, P., Edidin, D.: On signal reconstruction without phase. Appl. Comput. Harmon. Anal. **20**(3), 345–356 (2006)
5. Bandeira, A.S., Dobriban, E., Mixon, D.G., Sawin, W.F.: Certifying the restricted isometry property is hard. IEEE Trans. Inf. Theory **59**(6), 3448–3450 (2013)
6. Bangun, A., Behboodi, A., Mathar, R.: Sparse recovery in Wigner-D basis expansion. In: 2016 IEEE Global Conference on Signal and Information Processing (GlobalSIP), pp. 287–291. IEEE (2016)
7. Bangun, A., Behboodi, A., Mathar, R.: Coherence bounds for sensing matrices in spherical harmonics expansion. In: IEEE ICASSP'18. IEEE, Calgary, Canada (2018)
8. Bangun, A., Behboodi, A., Mathar, R.: Signal recovery from phaseless measurements of spherical harmonics expansion. In: 2019 27th European Signal Processing Conference (EUSIPCO) (EUSIPCO 2019). IEEE, A Coruña, Spain (2019)
9. Bangun, A., Culotta-López, C., Behboodi, A., Mathar, R., Heberling, D.: On phaseless spherical Near-Field antenna measurements. In: 13th European Conference on Antennas and Propagation (EUCAP 2019) (EuCAP 2019). IEEE, Krakow, Poland (2019)
10. Bangun, A., Behboodi, A., Mathar, R.: Sensing matrix design and sparse recovery on the sphere and the rotation group. IEEE Trans. Signal Process. **68**, 1439–1454 (2020). https://doi.org/10.1109/TSP.2020.2973545
11. Bangun, A., Behboodi, A., Mathar, R.: Tight bounds on the mutual coherence of sensing matrices for Wigner D-functions on regular grids. Sampling Theory Signal Process. Data Anal. **19** (2021). https://doi.org/10.1007/s43670-021-00006-2
12. Bauer, R.: Distribution of points on a sphere with application to star catalogs. J. Guid. Control Dynam. **23**, 130–137 (2000). https://doi.org/10.2514/2.4497
13. Bauschke, H.H., Combettes, P.L., Luke, D.R.: Phase retrieval, error reduction algorithm, and Fienup variants: a view from convex optimization. JOSA A **19**(7), 1334–1345 (2002)
14. Bucci, O.M., Gennarelli, C., D'Agostino, F.: A new and efficient NF-FF transformation with spherical spiral scanning. In: IEEE Antennas and Propagation Society International Symposium, pp. 629–632. IEEE (2001)
15. Burq, N., Dyatlov, S., Ward, R., Zworski, M.: Weighted eigenfunction estimates with applications to compressed sensing. SIAM J. Math. Anal. **44**(5), 3481–3501 (2012)
16. Candès, E.J., Romberg, J., Tao, T.: Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. IEEE Trans. Inf. Theory **52**(2), 489–509 (2006)
17. Candès, E.J., Strohmer, T., Voroninski, V.: PhaseLift: Exact and stable signal recovery from magnitude measurements via convex programming. Commun. Pure Appl. Math. **66**(8), 1241–1274 (2013)
18. Candès, E.J., Eldar, Y.C., Strohmer, T., Voroninski, V.: Phase retrieval via matrix completion. SIAM Review **57**(2), 225–251 (2015)
19. Chandra, R., Zhong, Z., Hontz, J., McCulloch, V., Studer, C., Goldstein, T.: PhasePack: A phase retrieval library. arXiv:1711.10175 (2017)
20. Chi, Y., Scharf, L.L., Pezeshki, A., Calderbank, A.R.: Sensitivity to basis mismatch in compressed sensing. IEEE Trans. Signal Process., 2182–2195 (2011)
21. Cornelius, R.: Fast Spherical Near-Field Antenna Measurement Methods. Dissertation, RWTH Aachen University (2018)

22. Cornelius, R., Heberling, D.: Spherical near-field scanning with pointwise probe correction. IEEE Trans. Antennas Propag. **65**(2), 995–997 (2017)
23. Culotta-López, C.: Fast Near-Field Measurements by Application of Compressed Sensing. Dissertation, RWTH Aachen University (2021)
24. Culotta-López, C., Heberling, D.: Fast spherical near-field measurements on arbitrary surfaces by application of pointwise probe correction to compressed sampling schemes. In: 2019 41st Antenna Measurement Techniques Association Symposium (AMTA). San Diego, California, USA (2019)
25. Culotta-López, C., Heberling, D., Bangun, A., Behboodi, A., Mathar, R.: A compressed sampling for spherical near-field measurements. In: 2018 40th (AMTA). Williamsburg Virginia, USA (2018)
26. Culotta-López, C., Bangun, A., Behboodi, A., Mathar, R., Heberling, D.: A modified Minimum-coherence sampling for fast spherical near-field measurements. In: 13th European Conference on Antennas and Propagation (EUCAP 2019) (EuCAP 2019). IEEE, Krakow, Poland (2019)
27. Culotta-López, C., Walkenhorst, B., Ton, Q., Heberling, D.: Practical considerations in compressed spherical near-field measurements. In: Antenna Measurement Techniques Association Symposium (AMTA), pp. 14–19. IEEE (2019)
28. Dhifallah, O., Thrampoulidis, C., Lu, Y.M.: Phase retrieval via linear programming: Fundamental limits and algorithmic improvements. In: 2017 55th Annual Allerton Conference on Communication, Control, and Computing (Allerton), pp. 1071–1077. IEEE (2017)
29. Dirksen, S., Lecué, G., Rauhut, H.: On the gap between restricted isometry properties and sparse recovery conditions. IEEE Trans. Inf. Theory (2016)
30. Donoho, D.L.: Compressed sensing. IEEE Trans. Inf. Theory **52**(4), 1289–1306 (2006). https://doi.org/10.1109/TIT.2006.871582
31. Fienup, J.R.: Phase retrieval algorithms: a comparison. Applied Optics **21**(15), 2758–2769 (1982)
32. Foucart, S., Rauhut, H.: A Mathematical Introduction to Compressive Sensing. Springer (2013)
33. Gerchberg, R.W.: A practical algorithm for the determination of phase from image and diffraction plane pictures. Optik **35**, 237–246 (1972)
34. Goldstein, T., Studer, C.: PhaseMax: Convex phase retrieval via basis pursuit. IEEE Trans. Inf. Theory (2018)
35. Grant, M., Boyd, S.: Graph implementations for nonsmooth convex programs. In: V. Blondel, S. Boyd, H. Kimura (eds.) Recent Advances in Learning and Control, Lecture Notes in Control and Information Sciences, pp. 95–110. Springer (2008)
36. Grant, M., Boyd, S.: CVX: Matlab software for disciplined convex programming, version 2.1. http://cvxr.com/cvx (2014)
37. Hansen, J.E.: Spherical near-field antenna measurements, vol. 26. IET (1988)
38. Hardin, D.P., Michaels, T., Saff, E.B.: A comparison of popular point configurations on sˆ 2. Dolomites Res. Note Approx. **9**(1) (2016)
39. Hofmann, B., Neitz, O., Eibert, T.F.: On the minimum number of samples for sparse recovery in spherical antenna near-field measurements. IEEE Trans. Antennas Propag. **67**(12), 7597–7610 (2019)
40. Kennedy, R.A., Sadeghi, P.: Hilbert Space Methods in Signal Processing. Cambridge University Press, Cambridge, UK (2013). OCLC: ocn835955494
41. Logan, B.F.: Properties of high-pass signals. Ph.D. thesis, Columbia University (1965)
42. Paulus, A., Knapp, J., Eibert, T.F.: Phaseless near-field far-field transformation utilizing combinations of probe signals. IEEE Trans. Antennas Propag. **65**(10), 5492–5502 (2017)
43. Rauhut, H., Ward, R.: Sparse recovery for spherical harmonic expansions. arXiv:1102.4097 (2011)
44. Saff, E.B., Kuijlaars, A.B.: Distributing many points on a sphere. Math. Intell. **19**(1), 5–11 (1997)
45. Santosa, F., Symes, W.W.: Linear inversion of band-limited reflection seismograms. SIAM J. Sci. Stat. Comput. **7**(4), 1307–1330 (1986)

46. SAS-571: Double ridge guide horn antenna, datasheet (2020). https://www.ahsystems.com/datasheets/SAS-571_Horn_Antenna_Datasheet.pdf
47. Schmidt, C.H., Rahmat-Samii, Y.: Phaseless spherical near-field antenna measurements: Concept, algorithm and simulation. In: 2009 IEEE Antennas and Propagation Society International Symposium, pp. 1–4 (2009). https://doi.org/10.1109/APS.2009.5171897
48. Schmidt, C.H., Razavi, S.F., Eibert, T.F., Rahmat-Samii, Y.: Phaseless spherical near-field antenna measurements for low and medium gain antennas. Adv. Radio Sci. **8**(B. 2-1/2-2), 43–48 (2010)
49. Tillmann, A.M., Pfetsch, M.E.: The computational complexity of the restricted isometry property, the nullspace property, and related concepts in compressed sensing. IEEE Trans. Inf. Theory **60**(2), 1248–1259 (2014)
50. van den Berg, E., Friedlander, M.P.: SPGL1: A solver for large-scale sparse reconstruction (2007). http://www.cs.ubc.ca/labs/scl/spgl1
51. Wei, K.: Solving systems of phaseless equations via Kaczmarz methods: A proof of concept study. Inverse Problems **31**(12), 125008 (2015)
52. Welch, L.: Lower bounds on the maximum cross correlation of signals (corresp.). IEEE Trans. Inf. theory **20**(3), 397–399 (1974)
53. Yaccarino, R.G., Rahmat-Samii, Y.: Phaseless bi-polar planar near-field measurements and diagnostics of array antennas. IEEE Trans. Antennas Propag. **47**(3), 574–583 (1999). https://doi.org/10.1109/8.768794
54. Zhang, Y., Yang, J., Yin, W.: YALL1: Your algorithms for L1. MATLAB software, http://www.caam.rice.edu/~optimizationL1 **1** (2010)

# Applied and Numerical Harmonic Analysis

**(104 volumes)**

1. A. I. Saichev and W. A. Woyczyński: *Distributions in the Physical and Engineering Sciences* (ISBN: 978-0-8176-3924-2)
2. C. E. D'Attellis and E. M. Fernandez-Berdaguer: *Wavelet Theory and Harmonic Analysis in Applied Sciences* (ISBN: 978-0-8176-3953-2)
3. H. G. Feichtinger and T. Strohmer: *Gabor Analysis and Algorithms* (ISBN: 978-0-8176-3959-4)
4. R. Tolimieri and M. An: *Time-Frequency Representations* (ISBN: 978-0-8176-3918-1)
5. T. M. Peters and J. C. Williams: *The Fourier Transform in Biomedical Engineering* (ISBN: 978-0-8176-3941-9)
6. G. T. Herman: *Geometry of Digital Spaces* (ISBN: 978-0-8176-3897-9)
7. A. Teolis: *Computational Signal Processing with Wavelets* (ISBN: 978-0-8176-3909-9)
8. J. Ramanathan: *Methods of Applied Fourier Analysis* (ISBN: 978-0-8176-3963-1)
9. J. M. Cooper: *Introduction to Partial Differential Equations with MATLAB* (ISBN: 978-0-8176-3967-9)
10. Procházka, N. G. Kingsbury, P. J. Payner, and J. Uhlir: *Signal Analysis and Prediction* (ISBN: 978-0-8176-4042-2)
11. W. Bray and C. Stanojevic: *Analysis of Divergence* (ISBN: 978-1-4612-7467-4)
12. G. T. Herman and A. Kuba: *Discrete Tomography* (ISBN: 978-0-8176-4101-6)
13. K. Gröchenig: *Foundations of Time-Frequency Analysis* (ISBN: 978-0-8176-4022-4)
14. L. Debnath: *Wavelet Transforms and Time-Frequency Signal Analysis* (ISBN: 978-0-8176-4104-7)



G. Kutyniok et al. (eds.), *Compressed Sensing in Information Processing*, Applied and Numerical Harmonic Analysis, https://doi.org/10.1007/978-3-031-09745-4

15. J. J. Benedetto and P. J. S. G. Ferreira: *Modern Sampling Theory* (ISBN: 978-0-8176-4023-1)
16. D. F. Walnut: *An Introduction to Wavelet Analysis* (ISBN: 978-0-8176-3962-4)
17. A. Abbate, C. DeCusatis, and P. K. Das: *Wavelets and Subbands* (ISBN: 978-0-8176-4136-8)
18. O. Bratteli, P. Jorgensen, and B. Treadway: *Wavelets Through a Looking Glass* (ISBN: 978-0-8176-4280-80)
19. H. G. Feichtinger and T. Strohmer: *Advances in Gabor Analysis* (ISBN: 978-0-8176-4239-6)
20. O. Christensen: *An Introduction to Frames and Riesz Bases* (ISBN: 978-0-8176-4295-2)
21. L. Debnath: *Wavelets and Signal Processing* (ISBN: 978-0-8176-4235-8)
22. G. Bi and Y. Zeng: *Transforms and Fast Algorithms for Signal Analysis and Representations* (ISBN: 978-0-8176-4279-2)
23. J. H. Davis: *Methods of Applied Mathematics with a MATLAB Overview* (ISBN: 978-0-8176-4331-7)
24. J. J. Benedetto and A. I. Zayed: *Sampling, Wavelets, and Tomography* (ISBN: 978-0-8176-4304-1)
25. E. Prestini: *The Evolution of Applied Harmonic Analysis* (ISBN: 978-0-8176-4125-2)
26. L. Brandolini, L. Colzani, A. Iosevich, and G. Travaglini: *Fourier Analysis and Convexity* (ISBN: 978-0-8176-3263-2)
27. W. Freeden and V. Michel: *Multiscale Potential Theory* (ISBN: 978-0-8176-4105-4)
28. O. Christensen and K. L. Christensen: *Approximation Theory* (ISBN: 978-0-8176-3600-5)
29. O. Calin and D.-C. Chang: *Geometric Mechanics on Riemannian Manifolds* (ISBN: 978-0-8176-4354-6)
30. J. A. Hogan: *Time-Frequency and Time-Scale Methods* (ISBN: 978-0-8176-4276-1)
31. C. Heil: *Harmonic Analysis and Applications* (ISBN: 978-0-8176-3778-1)
32. K. Borre, D. M. Akos, N. Bertelsen, P. Rinder, and S. H. Jensen: *A Software-Defined GPS and Galileo Receiver* (ISBN: 978-0-8176-4390-4)
33. T. Qian, M. I. Vai, and Y. Xu: *Wavelet Analysis and Applications* (ISBN: 978-3-7643-7777-9)
34. G. T. Herman and A. Kuba: *Advances in Discrete Tomography and Its Applications* (ISBN: 978-0-8176-3614-2)
35. M. C. Fu, R. A. Jarrow, J.-Y. Yen, and R. J. Elliott: *Advances in Mathematical Finance* (ISBN: 978-0-8176-4544-1)
36. O. Christensen: *Frames and Bases* (ISBN: 978-0-8176-4677-6)
37. P. E. T. Jorgensen, J. D. Merrill, and J. A. Packer: *Representations, Wavelets, and Frames* (ISBN: 978-0-8176-4682-0)
38. M. An, A. K. Brodzik, and R. Tolimieri: *Ideal Sequence Design in Time-Frequency Space* (ISBN: 978-0-8176-4737-7)
39. S. G. Krantz: *Explorations in Harmonic Analysis* (ISBN: 978-0-8176-4668-4)

40. B. Luong: *Fourier Analysis on Finite Abelian Groups* (ISBN: 978-0-8176-4915-9)
41. G. S. Chirikjian: *Stochastic Models, Information Theory, and Lie Groups, Volume 1* (ISBN: 978-0-8176-4802-2)
42. C. Cabrelli and J. L. Torrea: *Recent Developments in Real and Harmonic Analysis* (ISBN: 978-0-8176-4531-1)
43. M. V. Wickerhauser: *Mathematics for Multimedia* (ISBN: 978-0-8176-4879-4)
44. B. Forster, P. Massopust, O. Christensen, K. Gröchenig, D. Labate, P. Vandergheynst, G. Weiss, and Y. Wiaux: *Four Short Courses on Harmonic Analysis* (ISBN: 978-0-8176-4890-9)
45. O. Christensen: *Functions, Spaces, and Expansions* (ISBN: 978-0-8176-4979-1)
46. J. Barral and S. Seuret: *Recent Developments in Fractals and Related Fields* (ISBN: 978-0-8176-4887-9)
47. O. Calin, D.-C. Chang, and K. Furutani, and C. Iwasaki: *Heat Kernels for Elliptic and Sub-elliptic Operators* (ISBN: 978-0-8176-4994-4)
48. C. Heil: *A Basis Theory Primer* (ISBN: 978-0-8176-4686-8)
49. J. R. Klauder: *A Modern Approach to Functional Integration* (ISBN: 978-0-8176-4790-2)
50. J. Cohen and A. I. Zayed: *Wavelets and Multiscale Analysis* (ISBN: 978-0-8176-8094-7)
51. D. Joyner and J.-L. Kim: *Selected Unsolved Problems in Coding Theory* (ISBN: 978-0-8176-8255-2)
52. G. S. Chirikjian: *Stochastic Models, Information Theory, and Lie Groups, Volume 2* (ISBN: 978-0-8176-4943-2)
53. J. A. Hogan and J. D. Lakey: *Duration and Bandwidth Limiting* (ISBN: 978-0-8176-8306-1)
54. G. Kutyniok and D. Labate: *Shearlets* (ISBN: 978-0-8176-8315-3)
55. P. G. Casazza and P. Kutyniok: *Finite Frames* (ISBN: 978-0-8176-8372-6)
56. V. Michel: *Lectures on Constructive Approximation* (ISBN : 978-0-8176-8402-0)
57. D. Mitrea, I. Mitrea, M. Mitrea, and S. Monniaux: *Groupoid Metrization Theory* (ISBN: 978-0-8176-8396-2)
58. T. D. Andrews, R. Balan, J. J. Benedetto, W. Czaja, and K. A. Okoudjou: *Excursions in Harmonic Analysis, Volume 1* (ISBN: 978-0-8176-8375-7)
59. T. D. Andrews, R. Balan, J. J. Benedetto, W. Czaja, and K. A. Okoudjou: *Excursions in Harmonic Analysis, Volume 2* (ISBN: 978-0-8176-8378-8)
60. D. V. Cruz-Uribe and A. Fiorenza: *Variable Lebesgue Spaces* (ISBN: 978-3-0348-0547-6)
61. W. Freeden and M. Gutting: *Special Functions of Mathematical (Geo-)Physics* (ISBN: 978-3-0348-0562-9)
62. A. I. Saichev and W. A. Woyczyński: *Distributions in the Physical and Engineering Sciences, Volume 2: Linear and Nonlinear Dynamics of Continuous Media* (ISBN: 978-0-8176-3942-6)

63. S. Foucart and H. Rauhut: *A Mathematical Introduction to Compressive Sensing* (ISBN: 978-0-8176-4947-0)
64. G. T. Herman and J. Frank: *Computational Methods for Three-Dimensional Microscopy Reconstruction* (ISBN: 978-1-4614-9520-8)
65. A. Paprotny and M. Thess: *Realtime Data Mining: Self-Learning Techniques for Recommendation Engines* (ISBN: 978-3-319-01320-6)
66. A. I. Zayed and G. Schmeisser: *New Perspectives on Approximation and Sampling Theory: Festschrift in Honor of Paul Butzer's $85^{th}$ Birthday* (ISBN: 978-3-319-08800-6)
67. R. Balan, M. Begue, J. Benedetto, W. Czaja, and K. A. Okoudjou: *Excursions in Harmonic Analysis, Volume 3* (ISBN: 978-3-319-13229-7)
68. H. Boche, R. Calderbank, G. Kutyniok, and J. Vybiral: *Compressed Sensing and its Applications* (ISBN: 978-3-319-16041-2)
69. S. Dahlke, F. De Mari, P. Grohs, and D. Labate: *Harmonic and Applied Analysis: From Groups to Signals* (ISBN: 978-3-319-18862-1)
70. A. Aldroubi: *New Trends in Applied Harmonic Analysis* (ISBN: 978-3-319-27871-1)
71. M. Ruzhansky: *Methods of Fourier Analysis and Approximation Theory* (ISBN: 978-3-319-27465-2)
72. G. Pfander: *Sampling Theory, a Renaissance* (ISBN: 978-3-319-19748-7)
73. R. Balan, M. Begue, J. Benedetto, W. Czaja, and K. A. Okoudjou: *Excursions in Harmonic Analysis, Volume 4* (ISBN: 978-3-319-20187-0)
74. O. Christensen: *An Introduction to Frames and Riesz Bases, Second Edition* (ISBN: 978-3-319-25611-5)
75. E. Prestini: *The Evolution of Applied Harmonic Analysis: Models of the Real World, Second Edition* (ISBN: 978-1-4899-7987-2)
76. J. H. Davis: *Methods of Applied Mathematics with a Software Overview, Second Edition* (ISBN: 978-3-319-43369-1)
77. M. Gilman, E. M. Smith, and S. M. Tsynkov: *Transionospheric Synthetic Aperture Imaging* (ISBN: 978-3-319-52125-1)
78. S. Chanillo, B. Franchi, G. Lu, C. Perez, and E. T. Sawyer: *Harmonic Analysis, Partial Differential Equations and Applications* (ISBN: 978-3-319-52741-3)
79. R. Balan, J. Benedetto, W. Czaja, M. Dellatorre, and K. A. Okoudjou: *Excursions in Harmonic Analysis, Volume 5* (ISBN: 978-3-319-54710-7)
80. I. Pesenson, Q. T. Le Gia, A. Mayeli, H. Mhaskar, and D. X. Zhou: *Frames and Other Bases in Abstract and Function Spaces: Novel Methods in Harmonic Analysis, Volume 1* (ISBN: 978-3-319-55549-2)
81. I. Pesenson, Q. T. Le Gia, A. Mayeli, H. Mhaskar, and D. X. Zhou: *Recent Applications of Harmonic Analysis to Function Spaces, Differential Equations, and Data Science: Novel Methods in Harmonic Analysis, Volume 2* (ISBN: 978-3-319-55555-3)
82. F. Weisz: *Convergence and Summability of Fourier Transforms and Hardy Spaces* (ISBN: 978-3-319-56813-3)
83. C. Heil: *Metrics, Norms, Inner Products, and Operator Theory* (ISBN: 978-3-319-65321-1)

84. S. Waldron: *An Introduction to Finite Tight Frames: Theory and Applications.* (ISBN: 978-0-8176-4814-5)
85. D. Joyner and C. G. Melles: *Adventures in Graph Theory: A Bridge to Advanced Mathematics.* (ISBN: 978-3-319-68381-2)
86. B. Han: *Framelets and Wavelets: Algorithms, Analysis, and Applications* (ISBN: 978-3-319-68529-8)
87. H. Boche, G. Caire, R. Calderbank, M. März, G. Kutyniok, and R. Mathar: *Compressed Sensing and Its Applications* (ISBN: 978-3-319-69801-4)
88. A. I. Saichev and W. A. Woyczyński: *Distributions in the Physical and Engineering Sciences, Volume 3: Random and Fractal Signals and Fields* (ISBN: 978-3-319-92584-4)
89. G. Plonka, D. Potts, G. Steidl, and M. Tasche: *Numerical Fourier Analysis* (978-3-030-04305-6)
90. K. Bredies and D. Lorenz: *Mathematical Image Processing* (ISBN: 978-3-030-01457-5)
91. H. G. Feichtinger, P. Boggiatto, E. Cordero, M. de Gosson, F. Nicola, A. Oliaro, and A. Tabacco: *Landscapes of Time-Frequency Analysis* (ISBN: 978-3-030-05209-6)
92. E. Liflyand: *Functions of Bounded Variation and Their Fourier Transforms* (ISBN: 978-3-030-04428-2)
93. R. Campos: *The XFT Quadrature in Discrete Fourier Analysis* (ISBN: 978-3-030-13422-8)
94. M. Abell, E. Iacob, A. Stokolos, S. Taylor, S. Tikhonov, J. Zhu: *Topics in Classical and Modern Analysis: In Memory of Yingkang Hu* (ISBN: 978-3-030-12276-8)
95. H. Boche, G. Caire, R. Calderbank, G. Kutyniok, R. Mathar, P. Petersen: *Compressed Sensing and its Applications: Third International MATHEON Conference 2017* (ISBN: 978-3-319-73073-8)
96. A. Aldroubi, C. Cabrelli, S. Jaffard, U. Molter: *New Trends in Applied Harmonic Analysis, Volume II: Harmonic Analysis, Geometric Measure Theory, and Applications* (ISBN: 978-3-030-32352-3)
97. S. Dos Santos, M. Maslouhi, K. Okoudjou: *Recent Advances in Mathematics and Technology: Proceedings of the First International Conference on Technology, Engineering, and Mathematics, Kenitra, Morocco, March 26-27, 2018* (ISBN: 978-3-030-35201-1)
98. Á. Bényi, K. Okoudjou: *Modulation Spaces: With Applications to Pseudodifferential Operators and Nonlinear Schrödinger Equations* (ISBN: 978-1-0716-0330-7)
99. P. Boggiato, M. Cappiello, E. Cordero, S. Coriasco, G. Garello, A. Oliaro, J. Seiler: *Advances in Microlocal and Time-Frequency Analysis* (ISBN: 978-3-030-36137-2)
100. S. Casey, K. Okoudjou, M. Robinson, B. Sadler: *Sampling: Theory and Applications* (ISBN: 978-3-030-36290-4)

101. P. Boggiatto, T. Bruno, E. Cordero, H. G. Feichtinger, F. Nicola, A. Oliaro, A. Tabacco, M. Vallarino: *Landscapes of Time-Frequency Analysis: ATFA 2019* (ISBN: 978-3-030-56004-1)
102. M. Hirn, S. Li, K. Okoudjou, S. Saliana, Ö. Yilmaz: *Excursions in Harmonic Analysis, Volume 6: In Honor of John Benedetto's $80^{th}$ Birthday* (ISBN: 978-3-030-69636-8)
103. F. De Mari, E. De Vito: *Harmonic and Applied Analysis: From Radon Transforms to Machine Learning* (ISBN: 978-3-030-86663-1)
104. G. Kutyniok, H. Rauhut, R. J. Kunsch, *Compressed Sensing in Information Processing* (ISBN: 978-3-031-09744-7)

*For an up-to-date list of ANHA titles, please visit* http://www.springer.com/series/4968