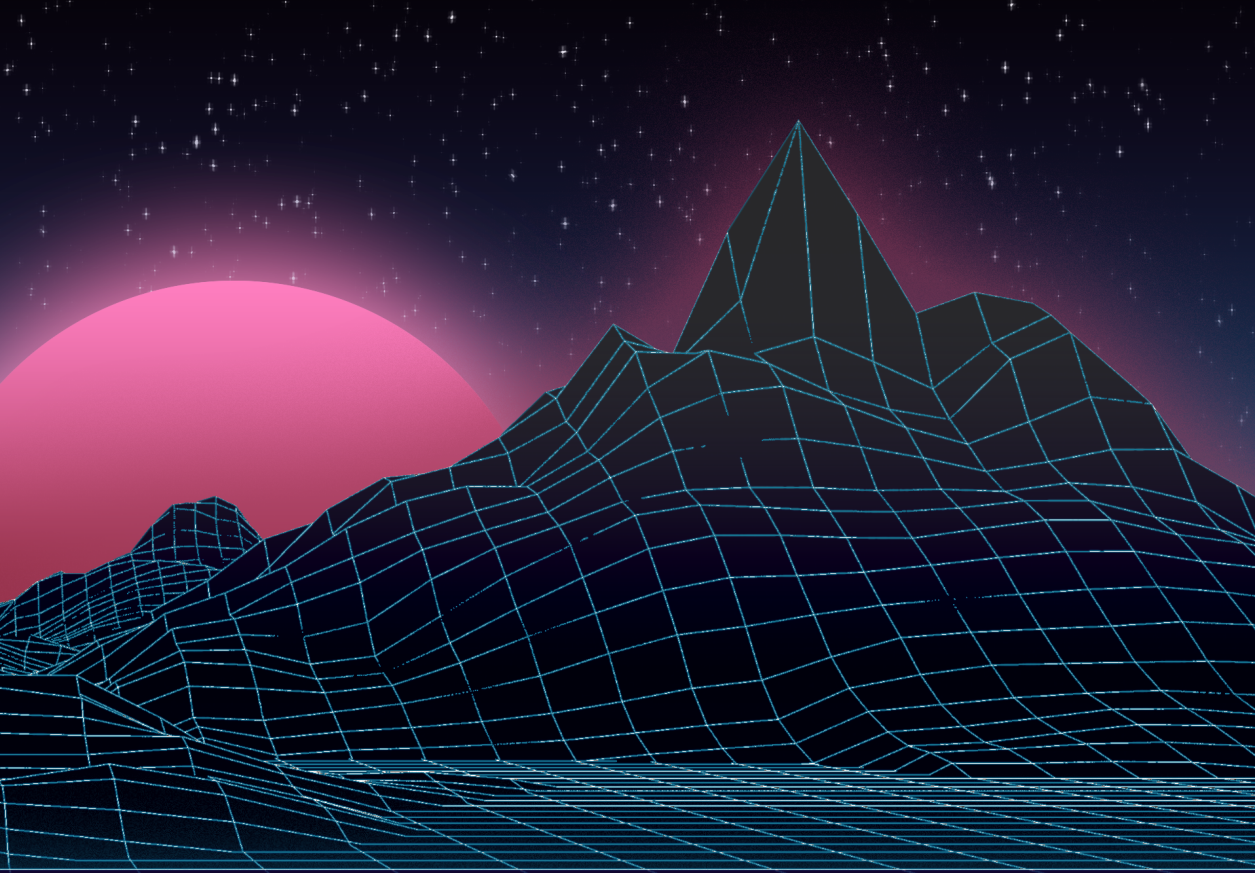


Efficient Algorithms and Data Structures for Compressive Sensing

Dissertation

Sebastian Semper

Efficient Algorithms and Data Structures for Compressive Sensing

Dissertation zur Erlangung des
akademischen Grades Doktor-Ingenieur (Dr.-Ing.)

vorgelegt der *Fakultät Fakultät für Elektrotechnik und Informationstechnik*
der *Technischen Universität Ilmenau*

von M. Sc. *Sebastian Semper*

1. Gutachter: *Univ.-Prof. Dr.-In. Giovanni Del Galdo*
2. Gutachter: *Prof. Dr. Orhan Arikan*
3. Gutachter: *Priv.-Doz. Dr. rer. nat. Martin Spies*

Tag der Einreichung: 16.10.2021
Tag der wissenschaftlichen Aussprache: 06.04.2022

URN: urn:nbn:de:gbv:ilm1-2022000096
DOI: 10.22032/dbt.51729

To all other students who committed themselves to research – past, present and future.

Abstract

Along with the ever increasing number of sensors, which are also generating rapidly growing amounts of data, the traditional paradigm of sampling adhering the Nyquist criterion is facing an equally increasing number of obstacles.

The rather recent theory of Compressed Sensing (CS) promises to alleviate some of these drawbacks by proposing to generalize the sampling and reconstruction schemes such that the acquired samples can contain more complex information about the signal than Nyquist samples. The proposed measurement process is more complex and the reconstruction algorithms necessarily need to be non-linear. Additionally, the hardware design process needs to be revisited as well in order to account for this new acquisition scheme. Hence, one can identify a trade-off between information that is contained in individual samples of a signal and effort during development and operation of the sensing system.

This thesis addresses the necessary steps to shift the mentioned trade-off more to the favor of CS. We do so by providing new results that make CS easier to deploy in practice while also maintaining the performance indicated by theoretical results. The sparsity order of a signal plays a central role in any CS system. Hence, we present a method to estimate this crucial quantity prior to recovery from a single snapshot. As we show, this proposed Sparsity Order Estimation method allows to improve the reconstruction error compared to an unguided reconstruction.

During the development of the theory we notice that the matrix-free view on the involved linear mappings offers a lot of possibilities to render the reconstruction and modeling stage much more efficient. Hence, we present an open source software architecture to construct these matrix-free representations and showcase its ease of use and performance when used for sparse recovery to detect defects from ultrasound data as well as estimating scatterers in a radio channel using ultra-wideband impulse responses.

For the former of these two applications, we present a complete reconstruction pipeline when the ultrasound data is compressed by means of sub-sampling in the frequency domain. Here, we present the algorithms for the forward model, the reconstruction stage and we give asymptotic bounds for the number of measurements and the expected reconstruction error. We show that our proposed system allows significant compression levels without substantially deteriorating the imaging quality. For the second application, we develop a sampling scheme to acquire the chan-

nel Impulse Response (IR) based on a Random Demodulator (RD) that allows to capture enough information in the recorded samples to reliably estimate the IR when exploiting sparsity. Compared to the state of the art, this in turn allows to improve the robustness to the effects of time-variant radar channels while also outperforming state of the art methods based on Nyquist sampling in terms of reconstruction error.

In order to circumvent the inherent model mismatch of early grid-based compressive sensing theory, we make use of the Atomic Norm Minimization (ANM) framework and show how it can be used for the estimation of the signal covariance with R-dimensional parameters from multiple compressive snapshots. To this end, we derive a variant of the Alternating Directions of Multipliers Method (ADMM) that can estimate this covariance in a very general setting and we show how to use this for direction finding with realistic antenna geometries. In this context we also present a method based on a Stochastic Gradient Descent (SGD) iteration scheme to find compression schemes that are well suited for parameter estimation, since the resulting sub-sampling has a uniform effect on the whole parameter space. Finally, we show numerically that the combination of these two approaches yields a well performing grid-free CS pipeline.

Zusammenfassung

Wegen der kontinuierlich anwachsenden Anzahl von Sensoren, und den stetig wachsenden Datenmengen, die jene produzieren, stößt die konventielle Art Signale zu verarbeiten, beruhend auf dem Nyquist-Kriterium, auf immer mehr Hindernisse und Probleme.

Die kürzlich entwickelte Theorie des Compressed Sensing (CS) formuliert das Versprechen einige dieser Hindernisse zu beseitigen, indem hier allgemeinere Signalaufnahme und -rekonstruktionsverfahren zum Einsatz kommen können. Dies erlaubt, dass hierbei einzelne Abtastwerte komplexer strukturierte Informationen über das Signal enthalten können als dies bei konventiellem Nyquistsampling der Fall ist. Gleichzeitig verändert sich die Signalrekonstruktion notwendigerweise zu einem nicht-linearen Vorgang und ebenso müssen viele Hardwarekonzepte für praktische Anwendungen neu überdacht werden. Das heißt, dass man zwischen der Menge an Information, die man über Signale gewinnen kann, und dem Aufwand für das Design und Betreiben eines Signalverarbeitungssystems abwägen kann und muss.

Die hier vorgestellte Arbeit trägt dazu bei, dass bei diesem Abwägen CS mehr begünstigt werden kann, indem neue Resultate vorgestellt werden, die es erlauben, dass CS einfacher in der Praxis Anwendung finden kann, wobei die zu erwartende Leistungsfähigkeit des Systems theoretisch fundiert ist. Beispielsweise spielt das Konzept der Sparsity eine zentrale Rolle, weshalb diese Arbeit eine Methode präsentiert, womit der Grad der Sparsity eines Vektors mittels einer einzelnen Beobachtung geschätzt werden kann. Wir zeigen auf, dass dieser Ansatz für Sparsity Order Estimation (SOE) zu einem niedrigeren Rekonstruktionsfehler führt, wenn man diesen mit einer Rekonstruktion vergleicht, welcher die Sparsity des Vektors unbekannt ist.

Um die Modellierung von Signalen und deren Rekonstruktion effizienter zu gestalten, stellen wir das Konzept von der matrixfreien Darstellung linearer Operatoren vor. Für die einfachere Anwendung dieser Darstellung präsentieren wir eine freie Softwarearchitektur und demonstrieren deren Vorzüge, wenn sie für die Rekonstruktion in einem CS-System genutzt wird. Konkret wird der Nutzen dieser Bibliothek, einerseits für das Ermitteln von Defektpositionen in Prüfkörpern mittels Ultraschall, und andererseits für das Schätzen von Streuern in einem Funkkanal aus Ultrabreitbanddaten, demonstriert.

Darüber hinaus stellen wir für die Verarbeitung der Ultraschalldaten eine Rekonstruktionspipeline vor, welche Daten verarbeitet, die im Fre-

quenzbereich Unterabtastung erfahren haben. Wir beschreiben effiziente Algorithmen, die bei der Modellierung und der Rekonstruktion zum Einsatz kommen und wir leiten asymptotische Resultate für die benötigte Anzahl von Messwerten, sowie die zu erwartenden Lokalisierungsgenauigkeiten der Defekte her. Wir zeigen auf, dass das vorgestellte System starke Kompression zulässt, ohne die Bildgebung und Defektlokalisierung maßgeblich zu beeinträchtigen. Für die Lokalisierung von Streuern mittels Ultrabreitbandradaren stellen wir ein CS-System vor, welches auf einem Random Demodulator (RD) basiert. Im Vergleich zu existierenden Messverfahren ist die hieraus resultierende Schätzung der Kanalimpulsantwort robuster gegen die Effekte von zeitvarianten Funkkanälen.

Um den inhärenten Modellfehler, den gitterbasiertes CS begehen muss, zu beseitigen, zeigen wir auf wie Atomic Norm Minimization (ANM) es erlaubt ohne die Einschränkung auf ein endliches und diskretes Gitter R-dimensionale spektrale Komponenten aus komprimierten Beobachtungen zu schätzen. Hierzu leiten wir eine R-dimensionale Variante des Alternating Directions of Multipliers Method (ADMM) her, welcher dazu in der Lage ist die Signalkovarianz in diesem allgemeinen Szenario zu schätzen. Weiterhin zeigen wir, wie dieser Ansatz zur Richtungsschätzung mit realistischen Antennenarraygeometrien genutzt werden kann. In diesem Zusammenhang präsentieren wir auch eine Methode, welche mittels Stochastic Gradient Descent (SGD) Messmatrizen ermitteln kann, die sich gut für Parameterschätzung eignen. Die hieraus resultierenden Kompressionsverfahren haben die Eigenschaft, dass die Schätzgenauigkeit über den gesamten Parameterraum ein möglichst uniformes Verhalten zeigt. Zuletzt zeigen wir auf, dass die Kombination des ADMM und des SGD das Design eines CS-Systems ermöglicht, welches in diesem gitterfreien Szenario wünschenswerte Eigenschaften hat.

A Thank You Note

I would like to thank a few notable people who tremendously shaped my professional life in academia and helped me to become the author of this presented thesis.

First, I would like to thank Thomas Hotz for helping me to take my first steps in research and showing me continuous support over all these years we have worked together. Thomas has sparked my initial drive to pursue research and was a role model all along the way.

If I had to pick a single person who is responsible for the existence of this very thesis it would be Florian Römer. It cannot be overestimated how much value our shared discussions, coffee breaks and joint publications helped me to shape my way of thinking, working and publishing. I also have to thank him for his useful feedback on an earlier version of this manuscript.

Giovanni Del Galdo is the head of our research group and as such he is providing all of us with a productive, interesting and especially an enjoyable working environment while also allowing us to draw from his experience in all aspects of research. For all these things I am very grateful and I also thank Giovanni for adopting me – the mathematician – in a group of engineers.

I would also like to thank my former office mates Jan Kirchhof, Fabian Krieg and Christoph Wagner. We had the luck of being dumped into our little Think Tank that allowed us to discuss, brainstorm, quarrel and grow. I think it is just to say that this very constellation of co-workers lead to many great ideas, fun working days, nourishing coffee breaks and hopefully some significant research. Also, Christoph and Jan helped me iron out some issues in an initial draft of this manuscript.

Finally, I have to thank Max Lörtzing for his guiding through the design and layout of this thesis together with the creation of the printed version's front and back cover.

Sebastian Semper
Ilmenau
April 22, 2022

x

Collaborations

Naturally, research is a joint effort and I have not contributed all alone to the topics as they are presented in this thesis.

Thomas Hotz contributed to the development of the results in Section 2.5 and Section 5.2.3 by mentoring the writing process and helping me with the line of though.

Florian Römer provided the initial idea for the theory further developed in Section 2.5. For Section 4.1 he was a constant internal reviewer of our thought process. The results on the R-dimensional in Section 5.2.6 are based on our joint derivation for the single-dimensional case of the algorithm.

Christoph Wagner joined forces with me when developing the software package described in Chapter 3 for which we both had to have many intense discussions and hours of programming together. For Section 4.2 I contributed most to the used mathematical formalism and the embedding of the signal processing ideas into the context of Compressive Sensing.

Jan Kirchhof and Fabian Krieg were very helpful early testers and users of the software described in Chapter 3. Also they both developed the topic in Section 4.1 from an application's side of view and among many other things provided the ultrasound specific codebase which we made extensive use of during the simulations presented in Section 4.1.

In his master thesis Eduardo Pérez developed the asymptotic bounds presented in Section 4.1.5 which we adapted slightly to our treated scenario.

Michael Döbereiner helped significantly when developing the software package for modeling antenna beampatterns as used in Section 5.2.6.

The results in Section 5.3 are heavily based on an original idea presented by Sankalp P. Pawar and he was contributing when developing, testing and publishing the proposed method.

Contents

List of Figures

List of Tables

List of Algorithms

List of Listings

Acronyms

ADC Analog-to-Digital Converter. xviii, 2, 141, 143, 161, 162

ADMM Alternating Directions of Multipliers Method. vi, viii, xix, 182, 193–201, 216–226, 231, 254

ANM Atomic Norm Minimization. vi, viii, 180, 182, 184, 186, 188–190, 193, 200–202

AOI Area of Interest. xvi, 3, 5, 6, 70, 227, 230

API Application Programmable Interface. 99, 100, 103, 107, 132

ARPACK ARnoldi PACKage. 116, 137, 144, 147, 148

BP Basis Pursuit. 19, 57

CF Crest factor. 159–161

CMOS Complementary Metal Oxide Semiconductor. 143

CPU Central Processing Unit. xviii, 116, 133, 145, 147

CR Compression Ratio. 166, 169, 171

CRB Cramér-Rao (lower) Bound. xviii, xix, 46, 111, 114, 116–118, 137, 138, 150, 151, 155, 156, 173, 191–194, 198–201, 210, 212, 213, 215, 225, 252

CS Compressed Sensing. v–viii, xxv, 2–13, 15, 18–20, 24, 27, 33, 34, 38, 42, 44, 46, 48, 51, 69–71, 76, 83, 97, 103–105, 109, 113–115, 118, 127, 142, 143, 147, 155–158, 170, 173, 175, 176, 178, 182, 229–231

CVX Convex Optimization toolboX. xix, 190, 200, 201, 225

DAS Delay-and-Sum. 111, 112, 114, 122

DCT Discrete Cosine Transform. 94

DFT Discrete Fourier Transform. 38, 75, 76, 84, 85, 101, 121–125, 132, 133

DoA Direction of Arrival. xviii, xix, 44, 176–181, 184–186, 192, 194, 198, 200–204, 207, 212, 213, 219, 222, 224, 225, 230

EADF Effective Aperture Distribution Function. 184, 191, 208, 213, 223

EFT Exponential Fitting Test. 56, 61

ESPRIT Estimation of Signal Parameters via Rotational Invariance Techniques. 181, 188, 197–199, 225

ETT Eigenvalue Threshold Test. 56, 61

FFT Fast Fourier Transform. 75, 76, 86, 87, 118, 132, 142–144, 146, 153, 247, 248

FHT Fast Hadamard Transform. 94, 167

FIM Fisher Information Matrix. 46, 137, 138, 251, 252

FISTA Fast Iterative Shrinkage-Thresholding Algorithm. xiv, xvii, xviii, xxiii, 31–33, 76, 80, 82, 103, 116, 118, 129, 133–137, 142, 144, 146–154, 156, 165, 184

FMCW Frequency-Modulated Continuous-Wave. 158, 160

FRVM Fast Relevance Vector Machine. 181

GPU Graphical Processing Unit. xviii, xxv, 105, 107, 114, 116, 132, 133, 145, 147, 156, 231

HRPE High Resolution Parameter Estimation. 179

iid independent and identically distributed. 35, 191, 198, 205

IR Impulse Response. vi, 157–160, 162–166, 168–170, 223

IRF Impulse Response Function. 162

ISTA Iterative Shrinkage-Thresholding Algorithm. 32, 103

LFSR Linear Feedback Shift Register. 106, 157, 161, 164, 167

LOS Line of Sight. 166, 169, 171

MIMO multiple input multiple output. 194

MLBS Maximum Length Binary Sequence. 161, 162, 165

MMV multiple measurement vectors. 48–50, 187

MRI Magnetic resonance imaging. 2, 3

MSE mean squared error. 216–222

MSM M-Sequence Method. 158, 160–167, 169, 171

MUSIC Multiple Signal Classification. 181, 188, 197

MWC Modulated Wideband Converter. 158

NDT Nondestructive Testing. 112, 114, 137, 155, 156

OMP Orthogonal Matching Pursuit. xvii, xviii, xxiii, 19, 28–30, 33, 34, 49, 57, 62, 63, 65, 67, 69, 76, 77, 79–81, 103, 105, 106, 151, 152, 157, 165, 168, 178, 179

OOP Object Oriented Programming. 99, 103

PN Pseudo-Noise. 158, 169

PURA Patch Uniform Rectangular Array. 208

RD Random Demodulator. vi, viii, xviii, 7, 157, 158, 160, 161, 164, 170, 171, 228

REM Reconstruction Error Metric. 169, 171

RIC Restricted Isometry Constant. 20, 22, 25, 34

RIMAX Richter Maximization Approach. 180

RIP Restricted Isometry Property. xiv, xvii, 20–22, 24, 26, 30, 34, 35, 37–39, 41–44, 70, 113, 125, 126

RMS root mean squared. 159

ROI Region of Interest. 116

RVM Relevance Vector Machine. 181

SAFT Synthetic Aperture Focusing Technique. 111, 112, 118, 129, 140–142, 144, 146, 151, 152, 154, 155

SCF spatial correlation function. 203–205, 208, 210, 212

SGD Stochastic Gradient Descent. vi, viii, xix, 202, 206–213, 215–225

SHM Structural Health Monitoring. 115

SISO Single Input Single Output. xviii, 166

SNR Signal-to-Noise Ratio. xix, 46, 162, 200, 201, 216–219, 222–224

SOE Sparsity Order Estimation. v, vii, xvii, xxiii, 52, 53, 55, 57, 60–64, 68, 69

SPUCPA Stacked Polarimetric Uniform Circular Patch Array. 190, 191

SSR Sparse Signal Recovery. xvii, xxv, 16, 21, 33, 55, 56, 63, 68, 69, 71, 76, 77, 97, 105, 113, 116, 129, 157, 165, 229

STELA Soft-Thresholding with Exact Line Search Algorithm. 33, 103

SUCA Stacked Uniform Circular Array. 210

SVD singular value decomposition. 44, 203

SVM Support Vector Machine. 181

T&H Track and Hold. xviii, 161–164, 228

TWISTA Two-step Iterative Shrinkage-Thresholding Algorithm. 33

ULA Uniform Linear Array. 177, 180

URA Uniform Rectangular Array. 219, 223

US-NDT Ultrasonic Non-destructive Testing. 111, 128, 173

UWB Ultra-Wideband. xviii, 110, 157, 158, 160, 165, 171, 173

Mathematical Notation and Quantites

Symbol	Definition
$\mathbb{N}$	natural numbers $\mathbb{N} = \{1, 2, \dots\}$
$\mathbb{N}_0$	$\mathbb{N} \cup 0$
$\mathbb{R}$	field of real numbers
$\mathbb{C}$	field of complex numbers
$\boldsymbol{y} \in \mathbb{K}^N$	usually a sampled version of a signal
$N \in \mathbb{N}$	size of the sampled signal $\boldsymbol{y} \in \mathbb{C}^N$
$\boldsymbol{x} \in \mathbb{C}^M$	usually a sparse representation vector $\boldsymbol{x}$ of a sampled signal $\boldsymbol{y}$
$M \in \mathbb{N}$	size of the sparse vector $\boldsymbol{x} \in \mathbb{C}^M$
$\boldsymbol{A} \in \mathbb{C}^{N \times M}$	sparsifying basis in case $N = M$, or dictionary in case of $N < M$
$\boldsymbol{\Phi}$	usually the compression matrix $\boldsymbol{\Phi} \in \mathbb{C}^{m \times N}$
$\boldsymbol{e}_i$	the i-th canonical basis vector of $\mathbb{C}^N$
$\boldsymbol{I}_N$	identity matrix in $\mathbb{C}^N$
$\operatorname{span}(A)$	linear hull of the vectors in the set $A \subset \mathbb{C}^n$
$\operatorname{supp}(\boldsymbol{x})$	support of a vector, i.e. the index set of non-zero entries
$\lVert \cdot \rVert_0$	the ℓ_0-quasi-norm, i.e. the number of non-zero entries of a vector
ℓ_p	sequence space of sequences that satisfy $\sum_i \lvert x_i \rvert^p < \infty$.
$\boldsymbol{x}^{\mathrm{H}}$, $\boldsymbol{A}^{\mathrm{H}}$	Hermitian transpose of a vector or matrix
spark	See Remark 2.2
k-rank	See Remark 2.2
$\jmath$	imaginary unit, i.e. $\mathbb{C} = \mathbb{R} + \jmath\mathbb{R}$
$[M]$	set integers n with $1 \leqslant n \leqslant M$
$\boldsymbol{\Psi}_{\cdot, \mathcal{S}}$	restrict the columns of $\boldsymbol{\Psi}$ to those indexed by the elements of $\mathcal{S}$

Symbol	Definition
$\boldsymbol{x}_S$	Restrict the entris of $\boldsymbol{x}$ to those indexed by the elements of $\mathcal{S}$
$\mathcal{N}(\mathbf{0}, \boldsymbol{I}_M)$	vector valued standard normal distribution in $\mathbb{R}^M$
$\Re, \Im$	real an imaginary part
$\overline{\boldsymbol{x}}$	elementwise complex conjugate of the vector $\boldsymbol{x}$
∇	first order differential operator
$\mathbf{0}_M$	zero vector in $\mathbb{C}^M$
$\sigma_{\max}(\boldsymbol{A})$	largest singular value of the matrix $\boldsymbol{A}$
$\mathcal{N}(\mu, \sigma)$	normal distribution with mean μ
$\boldsymbol{F}_n$	Discrete Fourier Transform matrix on $\mathbb{C}^n$
$\mathfrak{S}_C^R \boldsymbol{M}$	sub-selection of the matrix $\boldsymbol{M}$ where the colums are indexed by the index set C and the rows by R, respectivel
$\operatorname{diag}(\boldsymbol{\xi})$	diagonal matrix with $\boldsymbol{\xi}$ as main diagonal
$\boldsymbol{A} \otimes \boldsymbol{B}$	Kronecker matrix-matrix product
$\boldsymbol{A} \diamond \boldsymbol{B}$	columnwise Kronecker matrix-matrix product, i.e. Khatri-Rao product
$\|\boldsymbol{A}\|_\infty$	maximum absolute value of all elements of the matrix $\boldsymbol{A}$
$\operatorname{tr}(\boldsymbol{A})$	trace of the square matrix $\boldsymbol{A}$, i.e. the sum of all diagonal elements; equals the sum of all eigenvalues
$\|\boldsymbol{A}\|_F$	Frobenius norm of $\boldsymbol{A}$, i.e. $\|\boldsymbol{A}\|_F = \sqrt{\operatorname{tr} \boldsymbol{A}\boldsymbol{A}^{\mathrm{H}}}$
$f \circ g$	concatenation of the two functions g and f
$\operatorname{vec}(\boldsymbol{A})$	align the columns of $\boldsymbol{A} \in \mathbb{C}^{n \times m}$ into a vector $\boldsymbol{a} \in \mathbb{C}^{n \cdot m}$
$\mathbf{id}$	identity mapping, i.e. $\mathbf{id}(\boldsymbol{x}) = \boldsymbol{x}$

Symbol	Definition
$\mathbf{blkdiag}(\boldsymbol{A}_1, \ldots, \boldsymbol{A}_n)$	align the square matrices $\boldsymbol{A}_1$ to $\boldsymbol{A}_n$ in a block matrix along the diagonal
$\mathcal{CN}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$	vector valued circularly symmetric complex normal distribution with expectation $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$
$*$	non-cyclic convolution
$\circledast$	cyclic convolution
$\boldsymbol{A} \succcurlyeq \boldsymbol{B}$	$\boldsymbol{A} - \boldsymbol{B}$ is positive semi-definite
$\mathrm{conv}(A)$	convex hull of the vectors in the set A
$\mathbf{E}(X)$	Expectation of a random variable X
$\mathrm{Cov}(X)$	Covariance operator of a random variable X
$\det(\boldsymbol{A})$	determinant of the matrix $\boldsymbol{A}$, i.e. the product of its eigenvalues
$\boldsymbol{x} \odot \boldsymbol{y}$	pointwise product of the vectors $\boldsymbol{x}$ and $\boldsymbol{y}$, i.e. the Hadamard product
$\delta : \mathbb{R} \to \mathbb{R}$	Dirac δ-function
$\mathcal{S}_{\tau_i}$	shift operator on $\{f : \mathbb{R} \to \mathbb{C}\}$
$CF(x)$	Crest Factor of the signal x
$\mathbf{1}_n$	vector in $\mathbb{R}^n$ with all entries equal to 1
$\mathcal{O}(\cdot)$	big O notation to describe memory and runtime complexity

Chapter 1

Introduction

> "To live is to risk it all; otherwise you're just an inert chunk of randomly assembled molecules drifting wherever the universe blows you."
>
> *Rick Sanchez*

1.1 The Rise of Compressed Sensing

In recent years the number of sensing devices the we use to monitor a plethora of quantities in virtually all areas of human life has grown rapidly. We use sensors to monitor manufacturing processes in industrial applications [1], there are devices that capture a constant stream of images with ever increasing resolution and frame rates [2], the number of antenna ports in communications is steadily increasing [3] and our cars get equipped with numerous capabilities to monitor their environment [4].

Often, these sensors are naturally scattered in the environment in a distributed manner. Hence, the mainly digital post-processing of the acquired data is moving further away both in location as well as time from the point of acquisition. This means that the raw sensor data has to be transferred first before it is available for evaluation and further processing. In applications like structural health monitoring [5] the sensors are installed in disconnected locations like the inside of a tunnel, or on crucial points of a bridge. In these environments wired data transfer is not an option and often the available radio channels only have a limited capacity

for data transfer. This sparks the need for the reduction of data before the costly and slow transmission.

Other applications pose a high demand on the operating speed of the used Analog-to-Digital Converters (ADCs) that are employed during digitization of the measured signal. In these high-bandwidth and -frequency scenarios one has to pose restrictive assumptions on the signals in order to guarantee successful representation without aliasing effects or adapt the measurement process such that the sampling can be carried out at a lower rate. This means that in these cases the requirements posed by traditional Nyquist sampling lead to either unrealistic requirements on the ADC or unpractical assumptions about the signal.

Summarizing, with the mentioned rise in sensors, frame rates and resolutions the amounts of data that have to be transferred to the processing stage and converted by the ADCs are steadily increasing. The theory of Compressed Sensing (CS) formulates the promise that some cheap and most importantly data agnostic way of compression which operates in analog domain can be moved very close to the sensors resulting in less data to be transferred and converted. In fact the compression happens so close that the sensor itself is perceived as the entity conducting the compression – it is a *compressive sensor*. This offers the possibility to decrease the demands on data conversion and transfer substantially.

At the same time, this promise also comes at a price. First, these Compressed Sensing schemes have to be designed from scratch both theoretically and practically. One has to redesign new and custom electronics – often from the ground up – which have to implement this novel and more complex sampling scheme in affordable and feasible hardware. Second, due to the nature of this new sampling paradigm, the reconstruction of the signal becomes a non-trivial task to which end new algorithms have to be employed. That promise and these two interesting obstacles, namely the compressed measurements and non-linear reconstruction have led to a surge in research carried out by engineers and mathematicians alike, since it involves development of new hardware concepts, new signal processing schemes and also the discovery of interesting mathematical tools in information theory, geometry, numerics and optimization. For a Compressed Sensing based application to succeed all these tools have to be developed in unison and balanced against each other.

Soon, a large collection of publications shaped the new theory of CS by formulating it as an extension of the traditional sampling theory based on the Shannon-Nyquist sampling theorem. The proposed and possible applications today are numerous including imaging in the form of Magnetic

resonance imaging (MRI) [6], radar [7], or parameter estimation [8] for source localization [9] as well as spectral estimation [10]. Recently, one of the first proposed applications of CS for MRI[11] was incorporated into most major MRI platforms by now[1]. However, for several other applications the additional overhead to develop the innovative signal acquisition and reconstruction methods still outweighs the benefits.

Often, the reason for this additional overhead is the gap between requirements posed by the developed theory and practical restrictions, like hardware cost, calibration efforts, energy requirements or limits in computational power. This is due to the fact that although the theory of CS can be formulated in a very general fashion and as such seems to be readily applicable to many existing scenarios which have traditionally struggled with the limitations by Shannon-Nyquist sampling, the mentioned side constraints in a specific application are usually unique to the desired use case. Ultimately, every single application requires a lot of new and advanced engineering effort applying the concepts of CS.

In other words: the aforementioned gap between theory and implementation has to be closed by carefully studying specific applications in terms of their use case, side constraints and algorithmic aspects. The results presented in this thesis aim at narrowing this gap for two applications in imaging as well as a very general spectrum sensing setup.

1.2 Motivation and Contributions

Since the demands on a CS system heavily depend on the application, it is by no means straightforward to incorporate it into an arbitrary target system. This is supported by the fact that although the total number of publications on the field seems very promising, actual products is still lacking behind this initial drive. The obstacles during implementation are numerous and during the development of this thesis we summarized these into the following four distinct Areas of Interest (AOI). However, as we argue, the problems in these categories cannot be considered indepen-

[1] See for instance the press releases in:

→ https://www.radiologie-muenchen.de/ueber-uns/aktuelles/compressed-sensing

→ https://www.corporate.siemens-healthineers.com/de/press/releases/pr-2016110086hcen.html

→ https://www.philips.de/healthcare/resources/landing/the-next-mr-wave/compressed-sense

dently from each other. This thesis demonstrates how these categories' inter-relation ultimately makes up the characteristics and performance of a CS system.

Compression (AOI-C)

The theory of CS introduces a very different data acquisition scheme than the traditional Nyquist sampling allowing a more complex sampling process. These new degrees of freedom allow us to tailor the data capturing and compression process to the requirements one might have with respect to the complete CS system. So, these new degrees of freedom have to be used well and carefully in order to account for structural assumptions about the signal, limitations imposed by the hardware implementations or requirements on the system with respect to stability and robustness. ■

Hardware (AOI-H)

Possibly the largest obstacle is the design of suitable hardware that carries out the compression step with satisfying accuracy, speed and energy consumption. Initially, the proposed measurement schemes, which are optimal with respect to theoretical metrics, have turned out to be impractical as soon as one aims at designing the necessary hardware – often due to size, cost or calibration reasons. This sparks the need for alternative compression schemes, which approximate the behavior of theoretically optimal ones, while still being deployable in practice. Additionally, economies of scale might hinder a CS-based solution. So, although an innovative sensor would work but it might still be cheaper to employ the massively produced classical solution with traditional Nyquist sampling. ■

Reconstruction (AOI-R)

As already mentioned, special care is in order during the signal reconstruction stage. This is due to the fact that one generally has to solve a non-linear optimization problem to recover the information from the previously compressed data. Depending on the application, one either still has large amounts of data, desires a high accuracy during this recovery stage or one aims at very fast runtime of this stage in the pipeline. In any case these requirements spark the need for a careful analysis and fine tuning of all algorithmic aspects related to signal recovery. ■

Models (AOI-M)

When using CS in a certain system one usually formulates a signal model, which has direct influence on the design of rest of the CS system. The more accurate the signal model, the more refined the compression and reconstruction step can be. The feasibility of the whole CS pipeline in terms of stability, robustness, runtime of its solvers etc. inevitably depends on the specific model employed. So, this modeling stage needs special attention in order to design a well performing CS system. ■

The contribution of this thesis is to demonstrate that these four distinct AOI have to be considered as highly intertwined aspects of CS. We exemplify this union specifically for ultrasonic non-destructive testing, ultra-wideband radar and direction finding.

For instance, we will demonstrate that the compression setup has to obey the restrictions imposed on it by the available hardware. At the same time the sampling process has to be tailored to the specific signal model we are dealing with. Also, both the compression scheme and the signal model directly influence the necessary efforts and obtainable accuracy during the reconstruction stage.

To fill the above statement with more substance we outline the contributions of this thesis explicitly below, where we also connect them to the relevant publications and the previously mentioned four distinct Area of Interest (AOI).

1.2.1 Application Agnostic Contributions

Some results in this thesis are not tailored to a single application, but can be applied to a broader range of compressed sensing scenarios and we briefly put them into context below.

As we will see, the so-called sparsity order plays a central role both in practical considerations as well as in theoretical stability and robustness results. Since it usually is unknown prior to reconstruction, but serves as an indicator for the complexity of the signal, which in turn influences the reconstruction and compression stage, in [O1] we propose a model order selection scheme that does not involve reconstructing the signal and the results are presented in Section 2.5 thus addressing AOI-M and AOI-C.

During the rapid prototyping stage when designing a compressed sensing system one usually has to carry out large scale simulations, which

have to mimic and thus implement the whole processing chain. In order to facilitate this process we developed a free library for lazily evaluated linear transforms [O2, O3] that make use of matrix-free algorithms. This allows to conveniently compile algorithms and whole pipelines running close to optimality in terms of memory and runtime, while retaining much of the convenience when working with regular matrices. This approach is explained in more detail in Chapter 3 and as such it addresses AOI-R and to some degree AOI-M as well.

When using CS for parameter estimation, the compression scheme has to be adapted to the parametric model in order to achieve a measurement setup that allows low variance estimates across the whole parameter space of interest. To this end we propose a low complexity gradient algorithm [O4] inspired by methods used in machine learning, which is independent of the specific application at hand. We explain this approach in Section 5.3 thus tackling AOI-C.

Since multi-dimensional spectrum estimation has many different applications in array processing, radar etc. we still consider this task as a very generic problem. In order to employ CS in a practically relevant manner one has to first alleviate the requirement for a parameter grid, which turns the reconstruction method into a much more challenging problem. Thus, we propose a very general approximate algorithm [O5] in order to allow high resolution spectrum estimation from multiple compressed measurements. This method is described in Section 5.2.3 and contributes to AOI-R.

Some additional contributions are not presented in this thesis but still address to the mentioned AOI. We studied compression schemes that follow a Khatri-Rao-structure in [O6] addressing AOI-C. Additionally, to allow efficient direction finding from compressive measurements realistic antenna models are crucial and we deployed free software [O7] to provide convenient methods to deal with these antenna data, thus tackling AOI-M.

1.2.2 Application Specific Contributions

The previously mentioned algorithm to design well-behaved compressive measurements for parameter estimation is specified for the task of direction finding with compressive antennas, where we study the proposed algorithm in terms of expected estimation accuracy for arbitrary antenna geometries. The results of [O4] are presented in Section 5.3.3, hence there we address AOI-C.

As another specific application, we present the problem of designing a signal processing pipeline [O8], [O9], [O10], [O11] to find defects in an specimen excited by ultrasound waves. We show how to define a suitable measurement setup that uses samples in frequency domain and we compose a parametric model that also allows efficient reconstruction from very few samples. These concepts are explained in Section 4.1 where we address AOI-C, AOI-H, AOI-M and AOI-R. Hence, in this case we in fact cover the whole journey, down from a theoretic analysis, up until a hardware-aware compression scheme that allows efficient reconstruction afterwards.

In order to improve the signal energy collected by an ultra-wideband system, we propose to employ a Random Demodulator (RD) [O12, P1] to implement a CS system that uses linear feedback shift registers in order to realize a practically feasible compressive sensor for wideband channel impulse responses. As we show, the Doppler range in non-static scenarios is increased substantially and reconstruction of the impulse response can be carried out efficiently by using the proposed software for matrix-free algorithm. So, we address AOI-C, AOI-H and AOI-R as described in Section 4.2

As a last topic, we present an approach how to use the derived high resolution algorithm for spectral sensing for direction finding with arbitrary antenna arrays. We combine the results in Section 5.3.3 and Section 5.2.3 to show how one can find optimized compression schemes and an efficient algorithm for parameter estimation. These results tackling AOI-C and AOI-R are presented in Section 5.3.4.

Additional contributions not detailed in this thesis touch compressive channel sounding [O13] extending the concepts in [O4] to the much more challenging and higher dimensional setting. Finally, the concept of efficient matrix-free linear transforms was employed in ghost imaging [O14] to turn the reconstruction process into a feasible method by substantially decreasing the algorithmic complexity and also in [O15] where we employed a sparse recovery step to recover a sparse wavefield from subsampled tomography data.

1.3 Overview

We would like to note that due to the range of different topics covered in this thesis we do not present a state of the art for the whole manuscript beforehand, but introduce the topics of interest in their respective sections. This will make the presentations of independent topics more self contained, which allows to skip certain chapters entirely.

At a very high level, this thesis is split in to two main parts. The Chapters 2 to 4 contain results concerning *grid-bound* CS, which deal with the theory of CS, as it was initially developed. The later chapters extend this theory to a much more general and as we call it *grid-free* setting.

Chapter 2 introduces basic notions, like the general data model, the measurement setup and presents some key results concerning compression, recovery guarantees and reconstruction algorithms. Moreover, we present a general method to estimate the unknown sparsity order at the end of this chapter.

Motivated by these basic concepts, in Chapter 3 we introduce matrix-free linear mappings in order to improve upon the runtime of the previously presented algorithms. We also describe the software package that allows to easily work with these matrix-free linear mappings for modeling and reconstruction for CS systems and beyond.

Next, Section 4.1 and Section 4.2 depict how to use the theory presented beforehand applied to ultrasonic non-destructive testing and estimating channel impulse responses from ultra-wideband data.

Afterwards, in Chapter 5 we introduce the concepts involved in grid-free CS to allow high resolution parameter estimation. The results presented there derive a high-dimensional and efficient reconstruction algorithm. Also we present a method to optimize the used measurement kernels for the task of parameter estimation. This we apply to spectrum estimation and direction finding by presenting a pipeline that encompasses these two mentioned algorithms.

Finally, in Chapter 6 we iterate back on the motivations and the stated goal of this thesis in Section 6.1. Finally, in Section 6.3 we give an outlook on further research hence concluding this work.

Chapter 2

Grid-Bound Compressed Sensing

> "Would it save you a lot of time if I just gave up and went mad now?"
>
> *Douglas Adams*

This chapter first lays the foundation for notation, definitions and general assumptions we make during the whole thesis. We start by modeling our signals as elements in an appropriate function space and deduce a finite dimensional representation of the considered signals. By doing so, we provide a very short recap of conventional sampling based on the Nyquist criterion, which later on allows us to compare the theory of CS to it more easily.

Additionally, we outline a short way through the theory of grid-based CS and we derive that CS can be considered as another sampling theory besides Nyquist sampling. During the presentation of this alternative signal processing paradigm, we describe measurement setups via random projections, we present reconstruction guarantees for various approaches and we study two recovery algorithms more closely. Further, we present how measurement matrices can be evaluated in terms of the attainable stability during reconstruction and how initially random guesses can be optimized further.

Finally, we present a processing scheme that incorporates a novel method for sparsity order estimation from a single snapshot into the

reconstruction framework. This inference of the signal complexity prior to reconstruction allows to decrease the model mismatch during signal estimation and by virtue of that we provide new results for AOI-M.

2.1 Introduction and Basic Concepts

The following section gives a rapid and concise introduction to band-limited signals, which are the basis for sampling based on the Nyquist criterion. We shortly explain this theory in order to compare it to the statements that can be inferred in the context of CS later on. The current and the following chapters draw heavily from the comprehensive overview of CS that can be found in [12] and [13].

2.1.1 Bandlimited Signals

We start by introducing a suitable space for the signals we want to study in this thesis. First, we assume that the signals are in the separable Hilbert space $L^2(\mathbb{R})$ of square integrable functions with sesquilinear inner product $\langle \cdot, \cdot \rangle_{L^2}$ and induced norm $\|\cdot\|_{L^2}$ [14, Ch. V]. This means that for each $f \in L^2(\mathbb{R})$ with $t \mapsto f(t) \in \mathbb{C}$ it holds that

$$\|f\|_{L^2}^2 = \langle f, f \rangle_{L^2} = \int_{\mathbb{R}} \overline{f(t)} f(t) \, \mathrm{d}t < \infty,$$

where $\overline{x}$ denotes the complex conjugate of $x \in \mathbb{C}$. In a signal processing context, one usually denotes the signals contained in this space as signals of finite energy.

Additionally, we assume that the signals, or functions respectively, we consider are band-limited. In order to coin this term, we define the Fourier transform $\mathcal{F} : L^2(\mathbb{R}) \to L^2(\mathbb{R})$ via

$$f \mapsto \mathcal{F}f = \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} f(t) \exp(-it\cdot) \, \mathrm{d}t.$$

At first, it is not directly clear why this transformation is well behaved on the space L^2, which in fact it is not. However, for functions $f \in L^2$, which are also absolutely integrable, which means that

$$\int_R |f(t)| \, \mathrm{d}t < \infty,$$

we can indeed consider $\mathcal{F}$ to be a well behaved operator, see [14, Chapter V] for more details. Continuing, we define the set L_B of functions limited to the band $B \subset \mathbb{R}$ with $\sup(B \cup -B) < \infty$ as the set

$$L_B = \left\{ f \in L^2(\mathbb{R}) \cap L^1(\mathbb{R}) \;\middle|\; (\mathcal{F}f)(\mu) = 0 \quad \text{for} \quad \mu \notin B \right\}.$$

This means, the functions' Fourier transform that we consider is only non-zero on a bounded subset of $\mathbb{R}$. It is worth noting that the properties of L^2 make this function space a very well behaved space, because it can be shown to have an orthogonal basis. For the band-limited signals in L_B we are in a different and an even better situation. As we will see in the following results, it allows for a finite dimensional representation.

But first, let us present and discuss the sampling theory that is concerned with these band-limited signals, which is summarized in the following statement.

Theorem 2.1 (Shannon-Nyquist Theorem, [15] Thm 4.1)**.** *Let the function $f \in L^2(R)$ be a band-limited signal in L_B with $\sup(B \cup -B) < \pi/T$ for $T > 0$, then f can be reconstructed from the samples $f(nT)$, with $n \in \mathbb{Z}$ using the reconstruction formula*

$$f(t) = \sum_{n \in Z} f(nT) \operatorname{sinc}\left(\frac{t - nT}{T} \right),$$

where

$$\operatorname{sinc}(t) = \frac{\sin(\pi t)}{\pi t}$$

denotes the sinc *kernel or function.* ∎

Remark 2.1. Let us discuss this theorem in light of the theory we are going to develop in this thesis. First, we see that the sampling, or measurement process, is described by direct sampling in the sense that we acquire point evaluations of f given by the sequence $(f(nT))_{n \in \mathbb{Z}} \subset \mathbb{C}$. Second, we see that the theorem states that the reconstruction process of the complete signal f is linear in the samples $f(nT)$, which here means that it happens by merely forming the linear combination of appropriately shifted sinc-functions. Finally, note that the "complexity" of the signal is denoted by the size or extent of the band B, which contains the locations of the non-zero values of the signals Fourier-transform. ∎

In Remark 2.4 we are going to come back to the above points after we have developed similar statements for Compressed Sensing.

A last constraint we put on our signals f, is to assume that they are periodic in time-domain with period $0 < P < \infty$, so that

$$f(t) = f(t + n \cdot P) \quad \text{for} \quad n \in \mathbb{Z}.$$

This together with all other assumptions we made and Theorem 2.1 allows us to derive that the signals we consider are actually forming a finite-dimensional subspace in L^2. We summarize this in the following easy to obtain result.

Theorem 2.2. *Let $S_{B,P} \subset L_B$ be the set of band-limited functions with band B, which are periodic with period P and are also integrable. Then, $S_{B,P}$ is a finite dimensional vector space.* ∎

The proof can be found in Appendix A.1. The reasons why we went through the above obstacles are twofold as explained below.

Relationship of Sampling and Reconstruction

First, the short glimpse on the traditional sampling theorem and its direct ramifications on sampling and reconstruction dictates that the crucial quantity in the above sampling theory is the *bandwidth* of the signal. As such it determines the efforts we have to make in order to guarantee satisfying reconstruction with the samples we have acquired. During sampling the bandwidth determines the sample rate and during reconstruction is specifies the basis functions we use to restore the previously sampled signal. As we will see, CS will have similar albeit more involved results that depend on the *sparsity* of the signal. ∎

Finite-dimensional Vector Spaces

Second, we can now justify, why in the rest of the thesis we are dealing with finite dimensional vectors over the complex field in order to represent our signals. It is due to the result of Theorem 2.2 that for the signals we consider, we do not need to deal with elements in L^2 directly but can make use of the finite number of discrete samples of these and hence restrict ourselves to the tools of linear algebra, where matrices allow to describe linear mappings between finite dimensional vector spaces very conveniently. It is important to note that in expressions like

$$\boldsymbol{z} = \boldsymbol{\Phi} \cdot \boldsymbol{y},$$

we do not necessarily or implicitly assume that we have access to a sampled version $\boldsymbol{y}$ of the signal, but we only use this discrete to model the

whole compressive sensing system, which does not impose any requirements on the actual hardware implementation. ∎

Now that we have introduced Nyquist-based sampling as a theory that is centered around equidistant sampling of bandlimited signals, we introduce the theory of compressed sensing and its ramifications in the following sections.

2.1.2 The Linear Measurement Process

Since we have a convenient finite dimensional model for our signals, we are in the position to describe the measurement process that is the first corner stone of CS. Assume we are given a signal $\boldsymbol{y} \in \mathbb{C}^N$ and a set of vectors $\boldsymbol{\varphi}_i$ for $i = 1, \dots, m$, where $m < N$. Then, instead of assuming we have direct access to $\boldsymbol{y}$ we only have access to the values

$$z_i(\boldsymbol{y}) = \boldsymbol{\varphi}_i^{\mathrm{H}} \cdot \boldsymbol{y} \quad \text{for} \quad i = 1, \dots, m.$$

This can be interpreted as a sensing step, where each $\boldsymbol{\varphi}_i$ is a linear measurement kernel that is evaluated at $\boldsymbol{y}$. This is the linear and compressive measurement process we are going to consider in its most general form. We call it a compression step, since we choose $m < N$ and also the evaluation of the $\boldsymbol{\varphi}$ is linear in $\boldsymbol{y}$, because we have that $z_i(\lambda \cdot \boldsymbol{y}) = \lambda \cdot z_i(\boldsymbol{y})$ for any $\lambda \in \mathbb{C}$. Due to this linearity we define the matrix $\boldsymbol{\Phi} \in \mathbb{C}^{m \times N}$ via

$$\boldsymbol{\Phi} = \begin{bmatrix} \boldsymbol{\varphi}_1^{\mathrm{H}} \\ \vdots \\ \boldsymbol{\varphi}_m^{\mathrm{H}} \end{bmatrix}$$

and we can concisely describe the observation vector $\boldsymbol{z} = [z_1, \dots, z_m]^{\mathrm{H}}$ via

$$\boldsymbol{z} = \boldsymbol{\Phi} \cdot \boldsymbol{y}. \tag{2.1}$$

Summarizing, we have only indirect access to the signal of interest $\boldsymbol{y}$ by means of the linear functionals encoded by the rows of the sensing matrix $\boldsymbol{\Phi}$.

Example 2.1 (Single Pixel Camera)**.** Consider the case that we have to aquire samples of a $W \times H$-sized image encoded by the vector $\boldsymbol{y} \in \mathbb{C}^N$, where $N = W \cdot H$. However, the image can only captured by a very expensive sensor, since for instance we might be operating at an unconventional wavelength. In the extreme case we are only able to afford *a single* sensor – a single pixel camera [16]!

This means that sampling each "pixel" of the image individually is a too time-consuming process. With the above concepts this conventional pixel-wise sampling would look like

$$z_i = \boldsymbol{e}_i^{\mathrm{H}} \cdot \boldsymbol{y} = y_i \quad \text{for} \quad i = 1, \dots, N,$$

where $\boldsymbol{e}_i$ denotes the i-th canonical basis vector. In other words we have $\boldsymbol{\Phi} = \boldsymbol{I}_N \in \mathbb{C}^{N \times N}$, where $\boldsymbol{I}_N$ is the identity matrix of size N, so we have $m = N$, which means no compression at all and each row of $\boldsymbol{\Phi}$ only acquires information from one of the N pixels.

A different idea we are going to study in detail later would be to draw the entries in $\boldsymbol{\Phi}$ independently and identically distributed from the uniform distribution on $\{-1, +1\}$. This way each sample z_i of the image contains a unique "finger print" about *all* pixels at once. As we will see during the development of this chapter, there are means to recover $\boldsymbol{y}$ from $\boldsymbol{z}$ efficiently, even if $m \ll N$. However, only given some additional assumptions about $\boldsymbol{y}$, resulting in substantially fewer evaluations at the expensive sensor. ∎

Now that we have established the measurement process we are going to consider throughout this work, we must tackle the question of reconstructing the signal $\boldsymbol{y}$ from the observations $\boldsymbol{z}$, since due to the very different nature of sampling, the result provided by Theorem 2.1 is of no use anymore.

2.1.3 Sparsity Assumption

At first sight, the problem of recovering $\boldsymbol{y}$ seems impossible to solve, since we directly see that due to the fact that $\boldsymbol{\Phi} \in \mathbb{C}^{m \times N}$ with $m < N$, the kernel $\ker(\boldsymbol{\Phi})$ – the subspace of all vectors $\boldsymbol{x}$ such that $\boldsymbol{\Phi} \cdot \boldsymbol{x} = \boldsymbol{0}$ – of $\boldsymbol{\Phi}$ is non-trivial. So $\ker(\boldsymbol{\Phi})$ has at least dimension 1, which means there are infinitely many solutions $\boldsymbol{y}$ that solve $\boldsymbol{\Phi} \cdot \boldsymbol{y} = \boldsymbol{z}$. Hence, without any additional assumptions about the nature of the signal $\boldsymbol{y}$ and the measurement process encoded by $\boldsymbol{\Phi}$, we have no uniqueness in terms of the signal $\boldsymbol{y}$ that can explain our measurement $\boldsymbol{z}$. However, in Example 2.1 the image $\boldsymbol{y}$ is not just any element in the set of solutions to $\boldsymbol{\Phi} \cdot \boldsymbol{x} = \boldsymbol{z}$, but rather a very specific one, which we might need to model more carefully. Summarizing, we are in need for additional assumptions.

To this end, we introduce a set of vectors $\{\boldsymbol{a}_1, \dots, \boldsymbol{a}_M\}$, such that their linear hull given by $\operatorname{span}(\{\boldsymbol{a}_1, \dots, \boldsymbol{a}_M\}) = \mathbb{C}^N$, i.e. it spans the whole space $\mathbb{C}^N$. Note that this implies $M \geqslant N$. If we form the matrix $\boldsymbol{A} =$

$[\boldsymbol{a}_1, \ldots, \boldsymbol{a}_M] \in \mathbb{C}^{N \times M}$, we can use it to represent the signal via

$$\boldsymbol{y} = \boldsymbol{A} \cdot \boldsymbol{x} \tag{2.2}$$

for some not necessarily uniquely defined $\boldsymbol{x} \in \mathbb{C}^M$.

The crucial assumption we now make, is that there is a vector $\boldsymbol{x} \in \mathbb{C}^M$ such that it satisfies (2.2) and simultaneously has only $S \ll M$ non-zero entries. We say that $\boldsymbol{x}$ is S-sparse and $\boldsymbol{y}$ has an S-sparse representation with respect to $\boldsymbol{A}$. This *sparsity assumption* will turn out to be the second corner-stone in the theory of CS. Finally, we can introduce our complete observation model as

$$\boldsymbol{z} = \boldsymbol{\Psi} \cdot \boldsymbol{x} = \boldsymbol{\Phi} \cdot \boldsymbol{A} \cdot \boldsymbol{x}, \tag{2.3}$$

where we call $\boldsymbol{A} \in \mathbb{C}^{N \times M}$ the sparsifying dictionary, or in the special case $M = N$ the sparsifying basis. In order to access the sparsity pattern of $\boldsymbol{x}$, we define $\operatorname{supp}(\boldsymbol{x}) \subset [M]$ as the set containing the non-zero entries of $\boldsymbol{x}$, such that $S = |\operatorname{supp}(\boldsymbol{x})|$. Here, $[n]$ denotes the integers $\{1, \ldots, n\}$.

This means, we can transform the problem of finding a unique $\boldsymbol{y}$ satisfying (2.1) to finding a sparse $\boldsymbol{x}$ that satisfies (2.3) and again we can ask the question of uniqueness, but now for the sparse vector $\boldsymbol{x}$.

Remark 2.2 (Uniqueness via the Spark)**.** In order to get a first impression what the assumption of sparsity can do for us, we formulate another criterion for $\boldsymbol{\Psi} \in \mathbb{C}^{m \times M}$. For a given matrix $\boldsymbol{\Psi}$ the *spark* is the smallest number $\operatorname{spark}(\boldsymbol{\Psi}) = k \in \mathbb{N}$ such that there are k columns of $\boldsymbol{\Psi}$ which are linearly dependent. The dual to the spark is the Kruskal-rank $\operatorname{rk}^*(\boldsymbol{\Psi}) \in \mathbb{N}$, which is the largest number k such that every subset of k columns of $\boldsymbol{\Psi}$ is linearly independent. So, we have that $\operatorname{rk}^*(\boldsymbol{\Psi}) + 1 = \operatorname{spark}(\boldsymbol{\Psi})$.

Now, assume that $\operatorname{spark}(\boldsymbol{\Psi}) > 2K$ for some K in $\mathbb{N}$, hence no $2K$ columns of $\boldsymbol{\Psi}$ are linearly dependent. In other words, every submatrix $\boldsymbol{M} \in \mathbb{C}^{m \times 2K}$ consisting of $2K$ columns of $\boldsymbol{\Psi}$ has full rank. Of course, this can only be the case, if $2K \leqslant m$. Next, we consider a $\boldsymbol{y}$ that has an S-sparse representation with respect to $\boldsymbol{A}$, where $S \leqslant K$ and we assume that $\boldsymbol{y}$ has no unique representation with respect to $\boldsymbol{A}$.

So, there are two distinct $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ such that $\boldsymbol{z} = \boldsymbol{\Psi}\boldsymbol{x}_1 = \boldsymbol{\Psi}\boldsymbol{x}_2$. This on the other hand implies first that $\boldsymbol{0} \neq \boldsymbol{x}_1 - \boldsymbol{x}_2$ has at most $2S \leqslant 2K$ non-zeros entries and that $\boldsymbol{0}_m = \boldsymbol{\Phi}(\boldsymbol{x}_1 - \boldsymbol{x}_2)$. But this contradicts the assumption that all $2K$-columned submatrices of $\boldsymbol{\Psi}$ have full rank. Hence, we derived that there is a unique $\boldsymbol{x}$ satisfying (2.3). ∎

Remark 2.2 gives a first insight, how to derive a simple result on the uniqueness of the vector $\boldsymbol{x}$. Additionally, it already shows that the relation

between the measurement strategy $\boldsymbol{\Phi}$, the dictionary $\boldsymbol{A}$ and the possibility to recover $\boldsymbol{x}$ uniquely is not simple to characterize and we will iterate back to this relationship multiple times. It is worth noting that the process of estimating $\boldsymbol{x}$ from $\boldsymbol{z}$ under this sparsity prior is usually called Sparse Signal Recovery (SSR).

Additionally, to guarantee of uniqueness of $\boldsymbol{x}$ and hence of $\boldsymbol{y}$, it can also be of interest to recover a specific $\boldsymbol{x}$ from $\boldsymbol{z}$ instead of being satisfied with any $\boldsymbol{x}$ that is able to reconstruct $\boldsymbol{y}$. To this end, we present the following example, where the locations of the non-zero entries encode the information we would like to retrieve from the observation $\boldsymbol{z}$. This example [17] also serves as the blueprint for the applications considered in Section 4.1 and Section 4.2 as well as Chapter 5.

Example 2.2 (Delay Estimation)**.** As a simple yet instructive example, we consider the problem where we receive a periodic signal $y : \mathbb{R} \to \mathbb{C}$, which is formed by the superposition of *a few* shifted copies of a known and periodic waveform $a : \mathbb{R} \to \mathbb{C}$, i.e.

$$y(t) = \sum_{i=1}^{S} x_i \cdot a(t - \tau_i).$$

In this case, we would like to estimate the unknown amplitudes $x_i \in \mathbb{C}$ and normalized delays $\tau_i \in [0, 1)$ under the assumption that y has a *sparse* representation in terms of a. As such we have encountered a parameter estimation problem (or inverse problem) under a sparsity prior.

Since we assume as in Theorem 2.2 that a and hence also y occupy only a finite frequency band, we can discretize above equation to

$$\boldsymbol{y} = \sum_{i=1}^{S} x_i \cdot \boldsymbol{a}_{\tau_i},$$

where $\boldsymbol{a}_{\tau_i} \in \mathbb{C}^N$ is the vector of discrete samples of $a(t - \tau_i)$. Further, we define the matrix $\boldsymbol{A} \in \mathbb{C}^{N \times M}$ for some pair $N \leqslant M$ via

$$\boldsymbol{A} = \left[\boldsymbol{a}_0, \boldsymbol{a}_{1/M}, \ldots, \boldsymbol{a}_{(M-2)/M}, \boldsymbol{a}_{(M-1)/M}\right]$$

Effectively, we have defined a *parameter grid* that contains the waveforms we expect to observe given the shifts on the set of grid points $G = \{0, 1/M, \ldots, (M-1)/M\}$. The key idea now is to assume that $\boldsymbol{y}$ can be represented as a superposition of shifts residing on the grid G, which means there is a *sparse* vector $\boldsymbol{x} \in \mathbb{C}^M$ such that

$$\boldsymbol{y} = \boldsymbol{A} \cdot \boldsymbol{x}.$$

In this case, a non-zero entry x_i would result in $\boldsymbol{a}_{(i-1)/M}$ contributing to the observed signal y weighted with amplitude x_i. Hence, if we can estimate $\boldsymbol{x}$ from $\boldsymbol{y}$ we are able to recover the unknown x_i and the time-shifts τ_i solely from the sparse vector $\boldsymbol{x}$. ∎

Example 2.2 already sparks many interesting questions that are addressed in this thesis. What happens if we additionally employ compression by means of $\boldsymbol{\Phi} \in \mathbb{C}^{m \times N}$? How exactly should we choose $\boldsymbol{\Phi}$ in this case? What happens for $N \ll M$? How can we deal with the case when the true and unknown τ_i are not in the set G? Also note, how in this simple example we implicitly addressed AOI-M in the sense that the known waveform directly yields the structure of the sparsifying dictionary $\boldsymbol{A}$.

Next, we outline how specifically the assumption of sparsity helps us to efficiently recover the signal $\boldsymbol{y}$ and the sparse vector $\boldsymbol{x}$.

2.1.4 Exploiting Sparsity

In order to proceed, we first define the so called ℓ_0-norm, albeit not being an actual norm as $\|\cdot\|_0 : \mathbb{C}^M \to \mathbb{N}_0$ via

$$\boldsymbol{x} \mapsto \|\boldsymbol{x}\|_0 = |\{i \in \mathbb{N} : x_i \neq 0\}|.$$

The ℓ_0-norm simply counts the non-zero entries in a given vector – as such it directly measures the complexity of $\boldsymbol{x}$ in terms of its sparsity. Naturally, it violates the homogeneity property of the norm, since for $\lambda \neq 0$ we have that $\|\lambda \boldsymbol{x}\|_0 = \|\boldsymbol{x}\|_0$ for all $\boldsymbol{x} \in \mathbb{C}^M$.

Although, we now know that sparsity can help to reocver $\boldsymbol{x}$ uniquely, we have no algorithmic means of approaching the search for this sparse vector. As stated in [12][Chapter 2.2] we can observe the equivalence of the following two statements:

1. The vector $\boldsymbol{x}$ is the unique S-sparse solution to $\boldsymbol{\Psi x} = \boldsymbol{z}$, so

$$\{\boldsymbol{u} \in \mathbb{C}^M : \boldsymbol{\Psi u} = \boldsymbol{z}, \|\boldsymbol{u}\|_0 \leqslant S\} = \{\boldsymbol{x}\}.$$

2. The vector $\boldsymbol{x}$ can be reconstructed as the unique solution of

$$\min_{\boldsymbol{u} \in \mathbb{C}^M} \|\boldsymbol{u}\|_0 \quad \text{subject to} \quad \boldsymbol{\Psi u} = \boldsymbol{z}. \tag{2.4}$$

The equivalence of these statements gives us a first hint on how to formally recover the unknown $\boldsymbol{x}$. We can use the optimization problem in (2.4) as long as we can guarantee uniqueness of $\boldsymbol{x}$ beforehand, for instance by means of Remark 2.2. The approach of solving (2.4) is called

ℓ_0-minimization. As we also see above, unique recovery by means of this technique depends on the sparsity S, the measurement $\boldsymbol{\Phi}$ and the sparsifying dictionary $\boldsymbol{A}$.

Remark 2.3 (Balancing the System)**.** As we can already guess from Remark 2.2 the unique recovery of $\boldsymbol{x}$ becomes less likely for larger sparsity-levels S, since intuitively the conditions on $\boldsymbol{\Phi}$ and $\boldsymbol{A}$ are geometrically harder to be satisfied. There are two strategies one can exploit in order to still guarantee unique recovery.

First one could increase the performance of $\boldsymbol{\Phi}$. Here two strategies exist. For given m and N, one can try to find a good instance of $\boldsymbol{\Phi}$. In Section 2.4 we deal with the problem of evaluating and improving upon the performance of a given measurement matrix $\boldsymbol{\Phi} \in \mathbb{C}^{m \times N}$. On the other hand, one can also increase m allowing more degrees of freedom for $\boldsymbol{\Phi}$ and hence intuitively better performance. However, we would like to keep the measurement effort indicated by the magnitude of m a small as possible – the reason why we employ CS in the first place. This resulting trade-off between the performance of $\boldsymbol{\Phi}$ and the magnitude of m directly translates to the motivation AOI-C outlined in Section 1.2.

Second, one could find a better sparsifying dictionary $\boldsymbol{A}$. Here, we consider $\boldsymbol{A}_1$ to be better than $\boldsymbol{A}_2$ if we have that for every $\boldsymbol{y}$ there exist $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ with $\boldsymbol{y} = \boldsymbol{A}_1\boldsymbol{x}_1 = \boldsymbol{A}_2\boldsymbol{x}_2$ that satisfy $\|\boldsymbol{x}_1\|_0 < \|\boldsymbol{x}_2\|_0$. This is one aspect of motivation AOI-M as described in Section 1.2. Usually, $\boldsymbol{A}$ is derived from assumptions about the physical circumstances or other aspects of the specific system. We will come back to the design of $\boldsymbol{A}$ for a specific example in Section 4.1. ∎

In the following section we study the problem of ℓ_0-minimization in (2.4) more closely, since it is our first hope in designing a signal reconstruction method from an algorithmic perspective.

2.2 Sparse Reconstruction

Currently, we expect ourselves to be in a favorable situation. Given some observation model as in (2.3), we simply solve (2.4) and recover $\boldsymbol{x}$ and thus $\boldsymbol{y}$. But, as shown in [12][Theorem 2.17] the optimization problem one has to solve to carry out ℓ_0-minimization is NP-hard, rendering it an impractical approach for already moderately sized problems in terms of m, N and M.

In the following, we will present two approaches in circumventing this unpractical computational effort under additional assumptions about the

CS scenario. As a first remedy, we formulate an appropriate substitute problem that can be solved efficiently and which allows to recover $\boldsymbol{x}$ under additional assumptions. A second approach would be to only approximate a solution for the ℓ_0-minimization in (2.4) by using the Orthogonal Matching Pursuit (OMP) algorithm. As such, it offers a greedy approach for iteratively increasing the sparsity of the solution until satisfaction. It is studied in more detail in Section 2.3.1.

2.2.1 Just Relax

Given the NP-hardness of (2.4) one can justify to look for a substitute optimization problem to accomplish the signal reconstruction. The geometric reason for the intractability of ℓ_0-minimization is the fact that the ℓ_0-norm has no convex unit-norm balls. So the sets

$$B_0^r = \{\boldsymbol{x} \in \mathbb{C}^N : \|\boldsymbol{x}\|_0 \leqslant r\}$$

for any $r > 0$ are not convex. In fact, B_0^r is the (set) union of all subspaces spanned by at most $\lfloor r \rfloor$ coordinate axes. Note additionally that we can consider the ℓ_0-norm as the point-wise limit of

$$\|\boldsymbol{x}\|_0 = \lim_{p \to 0^+} \left(\sum_{i=1}^{N} |x_i|^p \right)^{1/p} = \lim_{p \to 0^+} \|\boldsymbol{x}\|_p.$$

For $0 < p < 1$ the sets B_p^r defined analogously as B_0^r above are not convex. This can be inferred from the fact that the function $t \mapsto t^p$ is not convex for $p < 1$. However, for $p \geqslant 1$ the balls B_p^r are convex and the function $\|\cdot\|_p$ is in fact a norm on $\mathbb{C}^N$.

The special case we now consider is the borderline case $p = 1$ resulting in

$$\|\boldsymbol{x}\|_1 = \sum_{i=1}^{N} |x_i|$$

as the convex relaxation of $\|\cdot\|_0$. Due to this convexity, the following so called ℓ_1-minimization problem, or Basis Pursuit (BP),

$$\min_{\boldsymbol{u} \in \mathbb{C}^M} \|\boldsymbol{u}\|_1 \quad \text{subject to} \quad \boldsymbol{\Psi} \cdot \boldsymbol{u} = \boldsymbol{z}. \tag{2.5}$$

has substantially better properties than the one in (2.4). In the case when we only consider a real valued scenario, where $\boldsymbol{x}$, $\boldsymbol{y}$ and $\boldsymbol{\Psi}$ are real valued, it can be equivalently recast as a linear optimization program [13, Sec. 1.5],

whereas in the complex case it is equivalent to a second order cone problem [12, Eq. $(P'_{1,\eta})$]. In both cases, theoretically efficient algorithms exist for solving the substitute problems, rendering ℓ_1-minimization a problem that can be solved in polynomial time with general purpose methods.

We would now like to use (2.5) as a proxy to solve (2.4), since the former promises efficient and fast reconstruction. However, we already deduced that ℓ_0-minimization is NP-hard. This means, in the general case the approach of ℓ_1-minimization cannot reproduce the same solution as ℓ_0-minimization. This means, we need to pose additional side constraints on the CS scenario if we wish for our proxy to succeed.

Hence, we ask the following question: What are the conditions on $\mathbf{\Psi}$ and $\boldsymbol{x}$ such that the solution $\boldsymbol{x}_0^*$ of (2.4) coincides with the solution $\boldsymbol{x}_1^*$ to (2.5)?

One main focus of many publications on CS [18, 19, 20] lies in deriving these conditions for various settings. In order to keep it short and connected to the rest of the thesis, we will focus on the two most important notions involved in these conditions.

2.2.2 The Restricted Isometry Property

The spark defined in Remark 2.2 of the matrix $\mathbf{\Psi}$ has been introduced as a quantity that indicates whether $\mathbf{\Psi}$ preserves sparsity of a sparse vector $\boldsymbol{x}$ in the sense that the set of S-sparse vectors sharing the same support set $\mathcal{S}$ are resided in an S-dimensional subspace. However, this is not enough, since this S-dimensional subspace could be almost $S-1$-dimensional in the sense that the basis vectors used to describe it form a matrix with large condition number. Due to the fact that the spark does not evaluate how much exactly the original $\boldsymbol{x}$ is preserved when being transformed as $\mathbf{\Psi}\cdot\boldsymbol{x}$, we need a more elaborate way to phrase the behavior of $\mathbf{\Psi}$. The geometric idea can be visualized in 2 spatial dimensions, where a perfect circle is distorted by $\mathbf{\Psi}$ to a very flat ellipse, hence almost degenerating it to a line. Although it might still be enclosing some space, we cannot claim that $\mathbf{\Psi}$ is well behaved in this case.

Hence we give the following definition that presents a refinement of the concept of the spark, see [12, Def. 6.1].

Definition 2.1 (Restricted Isometry Property)**.** The S-th Restricted Isometry Constant (RIC) $\delta_S(\mathbf{\Psi})$ of the matrix $\mathbf{\Psi}\in\mathbb{C}^{m\times M}$ is the smallest $\delta>0$ such that

$$(1-\delta)\|\boldsymbol{x}\|_2^2 \leqslant \|\mathbf{\Psi}\cdot\boldsymbol{x}\|_2^2 \leqslant (1+\delta)\|\boldsymbol{x}\|_2^2 \tag{2.6}$$

holds for all S-sparse vectors $\boldsymbol{x} \in \mathbb{C}^M$. We say that $\boldsymbol{\Psi}$ satisfies the restricted isometry property, if δ_S is small for some reasonably large S. ∎

If it is clear from context, we usually omit the specification of the matrix $\boldsymbol{\Psi}$. The geometric idea behind this definition is that for small δ_S and vectors up to sparsity order S the linear mapping $\boldsymbol{\Psi}$ almost is an isometry, since the norm of the image $\boldsymbol{\Psi} \cdot \boldsymbol{x}$ may only deviate slightly from $\|\boldsymbol{x}\|_2$. This entails that only small values for δ_S are interesting. Note for instance that $\delta_{2S} < 1$ yields that $\boldsymbol{\Psi}$ is injective on all S-sparse vectors, since for two different S-sparse $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ we have

$$\|\boldsymbol{\Psi}(\boldsymbol{x}_1 - \boldsymbol{x}_2)\|_2^2 \geqslant (1 - \delta_{2S})\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_2^2 > 0.$$

Additionally, if we reconsider the condition on $\boldsymbol{\Psi}$ in terms of the spark that is posed in Remark 2.2, we see that $\delta_{2K} < 1$ is equivalent to $\operatorname{spark}(\boldsymbol{\Psi}) > 2K$. The next result show us, how the Restricted Isometry Property (RIP) can help us with SSR.

Theorem 2.3 ([12] Thm. 6.9)**.** *Suppose that the $2S$-th restricted isometry constant of $\boldsymbol{\Psi} \in \mathbb{C}^{m \times M}$ satisfies*

$$\delta_{2S} \leqslant \frac{1}{3}.$$

Then every S-sparse $\boldsymbol{x} \in \mathbb{C}^M$ is the unique solution of (2.5). ∎

This entails that we have formulated a condition on $\boldsymbol{\Psi}$ and $\boldsymbol{x}$ in order to answer the question when ℓ_1-minimization is a viable proxy for ℓ_0-minimization. Hence, our original question about the equality of ℓ_0- and ℓ_1-minimization got transmuted into: When does a matrix $\boldsymbol{\Psi}$ satisfy $\delta_{2S}(\boldsymbol{\Psi}) < 1/3$? We answer this question in Section 2.4. The conclusion we can draw from (2.19), is given in the following statement.

Remark 2.4 (A Sampling Theorem for CS)**.** *Given the sampling process*

$$\boldsymbol{z} = \boldsymbol{\Phi} \cdot \boldsymbol{y},$$

where $\boldsymbol{y}$ has an S-sparse representation via $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$, then if we choose $\boldsymbol{\Psi} = \boldsymbol{\Phi} \cdot \boldsymbol{A}$ according to Theorem 2.3 we can use the solution $\boldsymbol{u}^*$ of

$$\min_{\boldsymbol{u} \in \mathbb{C}^M} \|\boldsymbol{u}\|_1 \quad \textit{subject to} \quad \boldsymbol{\Psi} \cdot \boldsymbol{u} = \boldsymbol{z}.$$

to recover $\boldsymbol{x} = \boldsymbol{u}^*$. As such we have derived a statement in the spirit of the Nyquist-Theorem 2.1, since we have given conditions on the signal $\boldsymbol{y}$ in the form of sparsity, we have specified the sampling process by

giving conditions on $\mathbf{\Psi}$ in terms of the RIP and we determined an efficient method that is able successfully recover the signal from the compressive measurement process. Hence, we have presented a sampling theorem that is valid for compressive sensing scenarios. ∎

The steps we have taken to derive the statement in Remark 2.4, can be repeated for other criteria we pose in terms of the measurement process, signal properties or means of reconstruction. Hence, the results in the following sections that conduct statements about these criteria allow to formulate additional, usually more specific or refined sampling theorems. For instance, the following section allows to exchange the condition on $\mathbf{\Psi}$ in terms of the RIP with the notion of the coherence of $\mathbf{\Psi}$.

2.2.3 Coherence

As we argue in Section 2.4 the RIC is hard to estimate for a given $\mathbf{\Psi}$ and reasonably large $2S$. So again – similar to the relation between ℓ_0- and ℓ_1-minimization, a valid proxy is needed in order to assess the quality or performance of a given $\mathbf{\Psi}$ in a simpler manner than dictated by the RIP. In contrast to the RIP, the coherence $\mu(\mathbf{\Psi})$ only inspects the relation of pairs of columns of $\mathbf{\Psi}$, which makes it easy to compute and also to interpret.

Definition 2.2 (Coherence). Let $\boldsymbol{A} \in \mathbb{C}^{m\times M}$ be an arbitrary matrix. Its coherence $\mu : \mathbb{C}^{m\times M} \to \mathbb{R}^+$ is defined as

$$\boldsymbol{A} \mapsto \mu(\boldsymbol{A}) = \max_{1\leqslant i<j\leqslant M} \frac{\left|\boldsymbol{a}_i^{\mathrm{H}}\boldsymbol{a}_j\right|}{\|\boldsymbol{a}_i\|_2\|\boldsymbol{a}_j\|_2}, \tag{2.7}$$

where $\boldsymbol{x}^{\mathrm{H}}$ denotes the Hermitian transpose of $\boldsymbol{x}$. ∎

The coherence has some easy to verify properties. First, one can show with the Cauchy-Schwartz inequality [15, Eq. (2.10)] that $0 \leqslant \mu(\boldsymbol{A}) \leqslant 1$. The special case $\mu(\boldsymbol{A}) = 0$ is equivalent to the fact that $\boldsymbol{A}$ is orthogonal, so $\boldsymbol{A}^{\mathrm{H}} \cdot \boldsymbol{A} = \boldsymbol{I}_M$. If additionally $m = M$ holds $\boldsymbol{A}$ is an orthogonal basis of $\mathbb{C}^M$. If we assume that $\boldsymbol{A} \in \mathbb{R}^{m\times M}$ and we only consider real valued signals and rewrite the definition of the coherence as

$$\mu(\boldsymbol{A}) = \max_{1\leqslant i<j\leqslant M} \left| \left(\frac{\boldsymbol{a}_j}{\|\boldsymbol{a}_j\|_2}\right)^{\mathrm{T}} \left(\frac{\boldsymbol{a}_i}{\|\boldsymbol{a}_i\|_2}\right) \right|,$$

we get a geometric intuition for μ. It is equal to the cosine of the smallest angle between two one-dimensional subspaces spanned by any pair of

two columns of $\boldsymbol{A}$. This idea is further developed in [O6] and exploited to use packing bounds in this so-called projective space in order to derive coherence bounds for matrices.

Clearly, for given $\boldsymbol{\Psi}$ the quantity $\mu(\boldsymbol{\Psi})$ is computable in $\mathcal{O}(mM^2)$, which allows conditions on $\boldsymbol{\Psi}$ in terms of the coherence to be verified in a computationally efficient manner. Indeed, one can show the following result on the equivalence of ℓ_1- and ℓ_0-minimization.

Theorem 2.4. *If the inequality*

$$\mu(\boldsymbol{\Psi}) < \frac{1}{2\|\boldsymbol{x}\|_0 - 1} \tag{2.8}$$

holds, then $\boldsymbol{x}$ is the unique solution to (2.4) and can be uniquely recovered by solving (2.5). ∎

It is instructive to compare both Theorems 2.3 and 2.4 to Theorem 2.1 as they both make the same kind of statement. All three results give conditions on the signal (the former in terms of bandwidth, the latter two in form of sparsity) under which a certain type of sampling (traditional digitizing in the former case, compression with $\boldsymbol{\Psi}$ in the latter) allows efficient signal reconstruction (linear sinc-interpolation in the former and non-linear ℓ_1-minimization in the latter) afterwards.

Additionally, considering (2.8) we can infer that the maximum complexity (read: sparsity) of the signals we are able to successfully recover with ℓ_1-minimization is determined by the coherence of $\boldsymbol{\Psi}$, since (2.8) entails that

$$\|\boldsymbol{x}\|_0 < \frac{1}{2}\left(\frac{1}{\mu(\boldsymbol{\Psi})} + 1\right) \tag{2.9}$$

has to hold true. Hence, in order to recover less sparse signals, we need to ensure that the sampling by means of $\boldsymbol{\Psi}$ delivers a *low* coherence. As such this gives the first optimization criterion that we can use in order to address AOI-C.

Again, we can iterate back to Remark 2.2, which relates the spark to uniqueness of the sparse solution. Interestingly, one can show that the coherence and the spark of a matrix are related [13, Lem. 2.1] via

$$\operatorname{spark}(\boldsymbol{\Psi}) \geqslant 1 + \frac{1}{\mu(\boldsymbol{\Psi})} \quad \text{or} \quad \operatorname{rk}^*(\boldsymbol{\Psi}) \geqslant \frac{1}{\mu(\boldsymbol{\Psi})}. \tag{2.10}$$

In fact, the above relation together with Remark 2.2 directly yields a proof for Theorem 2.4.

To conclude, we draw a connection between the simple coherence and the more involved RIP constant. One can show that

$$\delta_S \leqslant (S-1)\mu(\boldsymbol{\Psi})$$

holds for any $S \in \mathbb{N}$. This implies that the coherence can serve as a proxy in order to construct matrices that have a low RIP constant and we will exploit this in Section 2.4.3.

In principle, most criteria for successful and efficient sparse recovery revolve around the notion of the RIP and the coherence. Another popular concept is the so-called nullspace-property [12, See Ch. 4.1], which is closely related to both of the here treated concepts. Since it will serve our purpose for later sections, we will treat a more general RIP in Section 2.2.5.

2.2.4 Measurement Noise

As another generalization of our CS setup, we also need to account for measurement noise in our observation model. Since in applications noise cannot be avoided, we have to take it into consideration when evaluating the performance of the measurement matrices or the validity of the approaches for reconstruction. Hence, we extend (2.3) to

$$\boldsymbol{z} = \boldsymbol{\Phi} \cdot \boldsymbol{A} \cdot \boldsymbol{x} + \boldsymbol{n}, \tag{2.11}$$

where $\boldsymbol{n} \in \mathbb{C}^m$ accounts for zero-mean additive measurement noise. Naturally, we cannot expect perfect recovery of the signal or the sparse representation when the measurements are not perfect. We must be satisfied with the error in reconstruction scaling well behaved in terms of the noise level, usually measured by means of $\mathbf{E}\|\boldsymbol{n}\|_2^2$. First, we reformulate our prior recovery approaches (2.4) and (2.5) to account for the noise term as

$$\min_{\boldsymbol{u}\in\mathbb{C}^M} \|\boldsymbol{u}\|_0 \quad \text{subject to} \quad \|\boldsymbol{\Psi}\cdot\boldsymbol{u} - \boldsymbol{z}\|_2 \leqslant \eta \tag{2.12}$$

and

$$\min_{\boldsymbol{u}\in\mathbb{C}^M} \|\boldsymbol{u}\|_1 \quad \text{subject to} \quad \|\boldsymbol{\Psi}\cdot\boldsymbol{u} - \boldsymbol{z}\|_2 \leqslant \eta \tag{2.13}$$

as generalizations of the former versions, since we can just set $\eta = 0$ to recover the original problems for the noiseless case. We call these approaches quadratically constrained $\ell_{0,1}$-minimization. Note that it is usually not trivial to optimally choose η, since the value depends on the possibly unknown distribution of $\boldsymbol{n}$.

In order to demonstrate that ℓ_1-minimization still recovers practically relevant solutions, we cite only one representative result [12, Thm. 6.12].

Theorem 2.5. *Suppose that the* $2S$*-th RIC of the matrix* $\boldsymbol{\Psi}$ *satisfies*

$$\delta_{2S} < 0.6246.$$

Then, for any $\boldsymbol{x} \in \mathbb{C}^M$ *and* $\boldsymbol{z} \in \mathbb{C}^m$ *with* $\|\boldsymbol{\Psi} \cdot \boldsymbol{x} - \boldsymbol{z}\| \leqslant \eta$ *a solution* $\boldsymbol{x}^*$ *of the problem in* (2.13) *approximates the vector* $\boldsymbol{x}$ *with error*

$$\|\boldsymbol{x}^* - \boldsymbol{x}\|_1 \leqslant C\sigma_S(\boldsymbol{x}) + D\sqrt{S}\eta,$$

where

$$\sigma_S(\boldsymbol{x}) = \inf_{\boldsymbol{u} \in \mathbb{C}^M} \{\|\boldsymbol{x} - \boldsymbol{u}\|_1 : \|\boldsymbol{u}\|_0 = S\}$$

is the smallest S-sparse approximation-distance to $\boldsymbol{x}$*, i.e. the smallest possible distance any S-sparse vector can have to* $\boldsymbol{x}$*.* ∎

At first, we notice that Theorem 2.5 is not only a generalization to Theorem 2.3 but also a result that needs a weaker condition, since in the noise-free case $\eta = 0$ Theorem 2.5 guarantees perfect recovery with ℓ_1-minimization for $\delta_{2S} < 0.6246$ instead of $\delta_{2S} < 1/3$. Second, we see that the error degrades gracefully in terms of η, which is a desirable property.

Naturally, these properties of sparse recovery have been of great interest in the previous years and again [12, See Chs. 4, 6] provides a well balanced overview what can be derived.

2.2.5 Coherent Dictionaries

So far, we have mostly dealt with the whole system matrix $\boldsymbol{\Psi} = \boldsymbol{\Phi} \cdot \boldsymbol{A}$ directly. However, in practice the sparsifying matrix $\boldsymbol{A}$ is fixed, since it usually originates from physics or other inherent properties of the system that cannot be altered. In cases when $\boldsymbol{A}$ is invertible or even orthogonal, we can simply replace $\boldsymbol{\Psi}$ in the above statements with $\boldsymbol{\Phi}$, since as a sensing matrix we could employ $\boldsymbol{\Phi} \cdot \boldsymbol{A}^{-1}$, which yields $\boldsymbol{\Psi} = \boldsymbol{\Phi}$. However, in cases when $\boldsymbol{A}$ is not stably invertible one has to resort to different methods.

In order to introduce a last concept involved in recovery guarantees, we revisit Example 2.2. There, the matrix $\boldsymbol{A}$ is completely determined by the waveform a and the number of columns of $\boldsymbol{A}$ is determined by the resolution of the grid for the normalized delay parameter. Intuitively, on would increase the number of grid points for the normalized delays until the resolution is satisfactory for the intended use of the estimation. However, if we take the grid-density in Example 2.2, namely M^{-1} and calculate for $M > N$ that

$$\max_{1 \leqslant i < j \leqslant M} \left| \boldsymbol{a}_{(i-1)/M}^{\mathrm{H}} \cdot \boldsymbol{a}_{(j-1)/M} \right| = \left| \sum_{k=1}^{M} \exp\left(\jmath 2\pi \frac{k}{M} \right) \|\boldsymbol{a}_0\|_2^2 \right| \overset{M \to \infty}{\to} \|\boldsymbol{a}_0\|_2^2,$$

where $\jmath \in \mathbb{C}$ is the imaginary unit, we see that the coherence of $\boldsymbol{A}$ approaches 1 for increasing number of grid points. As a consequence, none of the previous recovery conditions in terms of the coherence or the RIP will be satisfied, if we strive for a high resolution in the estimated delays. The geometric interpretation is that columns of the sparsifying dictionary that are close in terms of the underlying delay parameter become harder to distinguish. Similar phenomena also occur in imaging with overcomplete wavelet or curvelet dictionaries [21].

One way to adapt for this observation is to taylor the RIP to the sparsifying dictionary at hand as done in [21, Def. 1.3].

Definition 2.3 ($\boldsymbol{A}$-RIP)**.** Let U_S be the union of all subspaces spanned by all subsets of S columns of $\boldsymbol{A} \in \mathbb{C}^{N \times M}$. A measurement matrix $\boldsymbol{\Phi} \in \mathbb{C}^{m \times N}$ obeys the restricted isometry property adapted to $\boldsymbol{A}$ with constant δ, if

$$(1-\delta)\|\boldsymbol{y}\|_2^2 \leqslant \|\boldsymbol{\Phi} \cdot \boldsymbol{y}\|_2^2 \leqslant (1+\delta)\|\boldsymbol{y}\|_2^2 \quad \text{for all} \quad \boldsymbol{y} \in U_S.$$

We call the smallest δ for which above chain of inequalities holds the restricted $\boldsymbol{A}$-isometry constant ($\boldsymbol{A}$-RIC) $\delta_S^{\boldsymbol{A}}$. ∎

The intuition behind this definition is that we require the matrix $\boldsymbol{\Phi}$ to satisfy the standard RIP on the *image* of k-sparse vectors under $\boldsymbol{A}$. As such it can be considered an extension of Definition 2.1. Similarly to Theorem 2.3 one can derive a reconstruction guarantee of the following form [21, Thm. 1.4].

Theorem 2.6. *Let $\boldsymbol{\Phi}$ be a measurement matrix that satisfies the $\boldsymbol{A}$-RIP with $\delta_{2S}^{\boldsymbol{A}} < 0.08$, then the solution $\boldsymbol{x}^*$ to (2.13) satisfies*

$$\|\boldsymbol{x}^* - \boldsymbol{x}\|_2 \leqslant C\eta + D\frac{\|\boldsymbol{A}\boldsymbol{x} - (\boldsymbol{A}\boldsymbol{x})_S\|_1}{\sqrt{S}},$$

where $(\boldsymbol{x})_S \in \mathbb{C}^N$ is the vector having set all entries except the S largest ones in magnitude to 0. Also the constants C and D do not depend on $\delta_{2S}^{\boldsymbol{A}}$. ∎

Interestingly, the statement Theorem 2.6 claims the unknown sparse $\boldsymbol{x}$ can still be recovered robustly depending on the intensity of the noise as quantified by η and how well behaved the sparsifying dictionary $\boldsymbol{A}$ is. As such it will serve us as a tool to motivate the sampling scheme employed in Section 4.1. Also it poses the most general reconstruction guarantee based on sparsity we present in this thesis.

2.2.6 Summary

Summarizing, we have developed several results that give insight in recovery from compressed measurements and under which conditions this can happen by means of an efficient algorithm in the form of ℓ_1-minimization. Often, it is not obvious that a given matrix $\mathbf{\Psi}$ satisfies the conditions that are imposed on it. Most of the presented results still depend on upper bounds of the usually unknown sparsity S of the signal of interest. Much like traditional Nyquist sampling has to impose a priori assumptions on the bandwidth

It remains to be shown how specific methods for solving the problems in (2.12) and (2.13) look like and what additional conditions they require in order to succeed. This is the subject of the next section.

2.3 Algorithms

The previous Section outlines how (2.12) and (2.13) can help us to recover a sparse vector from compressed measurements. There we have made first steps in deriving conditions under which the deployment of a Compressed Sensing scheme yields theoretically tractable optimization problems. However, these problems also need to be solved by specific algorithms. Although we have shown that in the real-valued case $\ell 1$-minimization can be cast as a linear program, the general purpose solvers do not take into account the structural knowledge we have about our problem.

One key property of the signals that are to be reconstructed in our case is *sparsity* with respect to the dictionary $\boldsymbol{A}$. To account for the special structure of the problems that arise in CS, a plethora of algorithms has been developed [22]. In the following, we focus only on two prominent examples, which can be considered a baseline for further refinements [23, 24, 25].

2.3.1 Orthogonal Matching Pursuit

If we revisit the model in (2.3), we see that our observation $\boldsymbol{z}$ is the linear superposition of a few columns of $\boldsymbol{\Psi}$. Hence, we have that

$$\boldsymbol{z} = \sum_{i \in \mathcal{S}} x_i \cdot \boldsymbol{\psi}_i + \boldsymbol{n}$$

for some index set $\mathcal{S} \subset \{1, \dots, M\} = [M]$. Assume now for a moment, we already have knowledge of the set $\mathcal{S}$. In this case, we know the non-zero entries of $\boldsymbol{x}$ and by virtue of that also the columns of $\boldsymbol{\Psi}$ that contribute to $\boldsymbol{z}$. If we collect these columns into the matrix $\boldsymbol{\Psi}_{\cdot,\mathcal{S}}$ and the non-zeros entries of $\boldsymbol{x}$ into $\boldsymbol{x}_{\mathcal{S}}$, we could simply solve a least squares problem

$$\min_{\boldsymbol{x}_{\mathcal{S}} \in \mathbb{C}^{|\mathcal{S}|}} \|\boldsymbol{\Psi}_{\cdot,\mathcal{S}} \cdot \boldsymbol{x}_{\mathcal{S}} - \boldsymbol{z}\|_2^2$$

by means of

$$\hat{\boldsymbol{x}}_{\mathcal{S}} = \boldsymbol{\Psi}_{\cdot,\mathcal{S}}^{\dagger} \cdot \boldsymbol{z},$$

where $\boldsymbol{A}^{\dagger}$ denotes the pseudo-inverse of $\boldsymbol{A}$. This is the reason for the *orthogonal* in the algorithms name, since for any $\mathcal{S}$ it holds that

$$\langle \boldsymbol{\Psi}_{\cdot,\mathcal{S}} \hat{\boldsymbol{x}}_{\mathcal{S}}, \boldsymbol{\Psi}_{\cdot,\mathcal{S}} \hat{\boldsymbol{x}}_{\mathcal{S}} - \boldsymbol{z} \rangle_2 = 0.$$

Additionally in case of Gaussian noise $\boldsymbol{n} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_M)$ the estimator $\hat{\boldsymbol{x}}_{\mathcal{S}}$ is the best linear and unbiased estimator for $\boldsymbol{x}_{\mathcal{S}}$, see [26, Chp. 9].

The key idea of OMP is to iteratively build up the set $\mathcal{S}$ by a greedy approach. Assume that we already already completed k iterations and have the set $\mathcal{S}_k$ for some $k \in \mathbb{N}$ and the respective $\hat{\boldsymbol{x}}_{\mathcal{S}_k}$. We are looking for the next column in $\boldsymbol{\Psi}$ that *matches* the so called residual

$$\boldsymbol{r}_k = \hat{\boldsymbol{x}}_{\mathcal{S}_k} - \boldsymbol{z}$$

best in terms of correlation. To this end, we calculate

$$c_i = \frac{|\langle \boldsymbol{\psi}_i, r_k \rangle_2|}{\|\boldsymbol{\psi}_i\|_2} \quad \text{for} \quad i = 1, \dots, M$$

in order to find the index $i \in \mathbb{N}$ such that c_i is maximized. Then we simply set $\mathcal{S}_{k+1} = \mathcal{S}_k \cup \{i\}$ and continue with the next step until we have reached some iteration $k = S_{\max}$. Finally, OMP returns an $S_{\max}$-sparse vector $\boldsymbol{x}$ with non-zero entries $\boldsymbol{x}_{\mathcal{S}}$ at indices $\mathcal{S}$.

In Algorithm 2.1, we give a formal representation of the described procedure. There are many flavors and variants of this basic approach [27,

Data: Observed measurement vector $\boldsymbol{z}$, Maximum sparsity order $S_{\max} \in \mathbb{N}$;

[1] Set $\mathcal{S}_0 = \emptyset$;
[2] $k = 0$;
[3] $\boldsymbol{r}_0 = \boldsymbol{z}$;
[4] $\hat{\boldsymbol{x}}_0 = \boldsymbol{0}$;
[5] **while** $k < S_{\max}$ **do**
[6] $\quad c_i = \frac{|\langle \boldsymbol{\psi}_i, \boldsymbol{r}_k \rangle_2|}{\|\boldsymbol{\psi}_i\|_2}$ for $i = 1, \ldots, M$;
[7] $\quad \mathcal{S}_{k+1} = \mathcal{S}_k \cup \operatorname{argmax}_i c_i$;
[8] $\quad \hat{\boldsymbol{x}}_{k+1} = \boldsymbol{\Psi}^{\dagger}_{\cdot,\mathcal{S}_{k+1}} \cdot \boldsymbol{z}$ on the set $\mathcal{S}_{k+1}$;
[9] $\quad \boldsymbol{r}_k = \boldsymbol{z} - \boldsymbol{\Psi} \hat{\boldsymbol{x}}_{\mathcal{S}_{k+1}}$;
[10] $\quad k = k + 1$;
[11] **end**
[12] return $\hat{\boldsymbol{x}}_{S_{\max}}$;

Algorithm 2.1: *OMP iteratively increases the sparsity and applies least-squares.* – Iteration scheme of the Orthogonal Matching Pursuit Algorithm.

28, 29]. Some introduce the ability to deal with multiple snapshots at once, others use a different stopping criterion based on $\boldsymbol{r}_k$, others enforce a certain structure on $\mathcal{S}_k$.

If we consider the algorithmic complexity of the steps in Algorithm 2.1, we might stumble into the pitfall of assuming that one needs to actually calculate the numerically involved pseudo-inverse of $\boldsymbol{\Psi}_{\cdot,\mathcal{S}}$ in each step or that the residual $\boldsymbol{r}_k$ has to be calculated from scratch for every k. However, with some additional analysis these steps can be avoided. Due to these subtleties, we defer the discussion of the computational complexity to Section 3.2.1.

When it comes to the algorithmic recovery performance, we can directly infer that for $\boldsymbol{\Psi}$ it has to hold that $\mu(\boldsymbol{\Psi}) < 1$, since in the case $\mu(\boldsymbol{\Psi}) = 1$ there would be two distinct columns $\boldsymbol{\psi}_p$ and $\boldsymbol{\psi}_q$ in $\boldsymbol{\Psi}$ such that there is a $\lambda \in \mathbb{C} \setminus \{0\}$ such that $\boldsymbol{\psi}_p = \lambda \boldsymbol{\psi}_q$. This directly implies that

$$c_p = \frac{\left|\langle \boldsymbol{\psi}_p, \boldsymbol{r}_k \rangle_2\right|}{\|\boldsymbol{\psi}_p\|_2} = \frac{\left|\langle \lambda \boldsymbol{\psi}_q, \boldsymbol{r}_k \rangle_2\right|}{\|\lambda \boldsymbol{\psi}_q\|_2} = \frac{\left|\langle \boldsymbol{\psi}_q, \boldsymbol{r}_k \rangle_2\right|}{\|\boldsymbol{\psi}_q\|_2} = c_q$$

for every step k of OMP. If now $x_q \neq 0$ or $x_p \neq 0$, at some iteration k we would have $\{p, q\} \subset \operatorname{argmax}_i |c_i|$, which would directly produce a non-unique solution $\hat{\boldsymbol{x}}_{\mathcal{S}_k}$.

Luckily, there exist reconstruction guarantees in terms of the notions from Section 2.2 that determine the recovery performance of OMP. To

this end we present one result [12, Prop. 6.24] involving the RIP.

Theorem 2.7. *Consider the model in* (2.3) *for some S-sparse* $\boldsymbol{x}$*. Let* $\boldsymbol{x}_k$ *be the iterate of Algorithm 2.1. If*

$$\delta_{13S} < \frac{1}{6},$$

then there is a constant $C > 0$ *depending only on* δ_{13S} *such that*

$$\|\boldsymbol{z} - \boldsymbol{\Psi} \cdot \boldsymbol{x}_{12S}\|_2 \leqslant C\|\boldsymbol{n}\|_2.$$

■

As we can see, although the algorithm technically produces a k-sparse vector after k iterations, in order to guarantee that the greedy strategy actually finds the true columns of $\boldsymbol{\Psi}$ needed to represent $\boldsymbol{z}$, one has to run Algorithm 2.1 for $12S$ steps to exactly recover an S-sparse vector in the case $\boldsymbol{n} = 0$. As such it can be viewed as an *approximate* solver of the problem in (2.12), since usually one employs some early stopping criterion to detect sufficiently accurate reconstructions before reaching $12S$ steps.

Due to its simplicity in terms of the implementation, it usually serves as a good gateway algorithm to get started in a compressed sensing system. Additionally, the fact that it produces exactly k-sparse solutions makes it a popular choice, when a certain sparsity level is desired.

2.3.2 Fast Iterative Shrinkage Thresholding Algorithm

The previously described OMP algorithm is very intuitive and also has good runtime properties if implemented correctly. However, due to the greedy approach which is adding one column of $\boldsymbol{\Psi}$ in each step it only acts locally on the sparse vector $\boldsymbol{x}$. Hence, it is not possible to run OMP just for "a few" iterations and get a globally reasonable estimate for $\boldsymbol{x}$. However, in some cases it might be beneficial if one can make use of an algorithm that is able to update all entries of its estimate for the sparse vector $\boldsymbol{x}$ in each step. As we will see in Section 4.1 especially in imaging applications this property helps to achieve reconstructions that rapidly show a good approximation of the true image. In the following, we present a prominent example for such a global algorithm.

An in-depth analysis of the algorithm described in this chapter can be found in [30]. To tackle the problem in (2.13), we wish to employ iterative methods of convex optimization in order to recover the sparse vector $\boldsymbol{x}$.

To demonstrate convexity, we can reformulate (2.13) as

$$\min_{\boldsymbol{u}\in\mathbb{C}^M} \|\boldsymbol{\Psi}\cdot\boldsymbol{u}-\boldsymbol{z}\|_2^2+\lambda(\eta)\|\boldsymbol{u}\|_1, \tag{2.14}$$

which means for given $\eta>0$ there is some $\lambda>0$ such that (2.5) and (2.14) have the same solution, see [31, Ex. 5.2 to Ex. 5.4] for the case $\eta=0$ or [12, Thm B.28] for the more general case $\eta>0$. Now, since the sum of two convex functions is convex, the objective in (2.14) is convex as well. Hence, more generally we are confronted with an optimization problem of the form

$$\min_{\boldsymbol{u}\in\mathbb{C}^M} f(\boldsymbol{u})$$

for which we can simply rely on the iteration

$$\boldsymbol{u}_{k+1}=\boldsymbol{u}_k+\alpha_k\nabla f(\boldsymbol{u}_k), \tag{2.15}$$

where the operator ∇ is the complex Wirtinger-gradient, which reads as

$$[\nabla f(\boldsymbol{u})]_{i=1}^M=\left(\frac{\partial}{\partial\Re u_i}-\jmath\frac{\partial}{\partial\Im u_i}\right)f(\boldsymbol{u})\in\mathbb{C}^M.$$

Here, $\Re z$ denotes the real part of the complex number $z\in\mathbb{C}$ whereas $\Im z$ denotes the imaginary part respectively. See Appendix B.2 for details on this derivative for functions depending on complex variables.

In order to derive the Fast Iterative Shrinkage-Thresholding Algorithm (FISTA) scheme we notice that $\boldsymbol{u}_{k+1}$ in (2.15) can also be found by virtue of

$$\boldsymbol{u}_{k+1}=\operatorname*{argmin}_{\boldsymbol{u}}\left\{f(\boldsymbol{u}_k)+\langle\boldsymbol{u}-\boldsymbol{u}_k,\nabla f(\boldsymbol{u}_k)\rangle+\frac{1}{2\alpha_k}\|\boldsymbol{u}-\boldsymbol{u}_k\|_2^2\right\},$$

which can be rewritten to

$$\boldsymbol{u}_{k+1}=\operatorname*{argmin}_{\boldsymbol{u}}\left\{\frac{1}{2\alpha_k}\|\boldsymbol{u}-(\boldsymbol{u}_k-\alpha_k\nabla f(\boldsymbol{u}_k))\|_2^2+\lambda\|\boldsymbol{u}\|_1\right\},$$

and in the special case of (2.14) reads as

$$\boldsymbol{u}_{k+1}=\tau_{\lambda\cdot\alpha_k}\left(\boldsymbol{u}_k-\alpha_k\boldsymbol{\Psi}^{\mathrm{H}}(\boldsymbol{\Psi}\boldsymbol{u}_k-\boldsymbol{z})\right). \tag{2.16}$$

Here, $\tau_s:\mathbb{C}^M\to\mathbb{C}^M$ denotes the shrinkage (or soft-thresholding) operator defined via

$$\boldsymbol{u}\mapsto\tau_s(\boldsymbol{u})=\max\{|\boldsymbol{u}|-s,0\}\sphericalangle(\boldsymbol{u}),$$

Data: Observed measurement vector z, Maximum number of steps $K_{\max} \in \mathbb{N}$;

[1] $k = 0$, $t_0 = 1$, $v_0 = u_0 = \mathbf{0}_M \in \mathbb{C}^M$;

[2] **while** $1 \leqslant k \leqslant K_{\max}$ **do**

[3] $\quad u_{k+1} = \tau_{\lambda \cdot \alpha_k}\left(v_k - \alpha_k \Psi^{\mathrm{H}}(\Psi v_k - z)\right)$;

[4] $\quad t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}$;

[5] $\quad v_{k+1} = u_{k+1} + \left(\frac{t_k - 1}{t_{k+1}}\right)(u_{k+1} - u_k)$;

[6] **end**

Algorithm 2.2: *A momentum-based first order method converging rapidly.* – Iteration scheme for Fast Iterative Shrinkage-Thresholding Algorithm.

where all functions in this expression are acting pointwise on u. Here $\sphericalangle(z)$ denotes the argument of the complex number $z \in \mathbb{C}$. Also, we used that

$$\nabla \|\Psi u - z\|_2^2 = \Psi^{\mathrm{H}}(\Psi u - z).$$

This can be derived with the results in Appendix B.2. Interestingly, the iteration in (2.16) constitutes the Iterative Shrinkage-Thresholding Algorithm (ISTA) scheme, see [30, Sec. 3] or [32], which is a predecessor of FISTA. To derive FISTA from ISTA we not only consider the current iterate u_k but also the previous u_{k-1} by forming a linear combination of these via $v_k = \mu_1 u_k + \mu_2 u_{k-1}$. The original idea comes from the so-called Nesterov-momentum [33], where a specific calculation for μ_1 and μ_2 is employed, which is thoroughly derived in [30].

In Algorithm 2.2 we have given the procedure for FISTA in a very simple form, where we just stop the iteration after a fixed number of steps. Note that far more elaborate stopping criteria can be given, for instance depending on the norm of $\nabla \|\Psi u_k - z\|_2^2$.

The most notable property of FISTA is the speed of convergence, which can be derived [30, Thm. 4.4] as

$$F(u_k) - F(u^*) \leqslant \frac{L\|u_0 - u_k\|_2^2}{(k+1)^2},$$

where F is the objective function in (2.14) and $L \in \mathbb{R}_+$ is an upper bound on the Lipschitz constant of $u \mapsto \|\Psi u - z\|_2^2$. This convergence rate assumes that we have chosen the stepsize according to $\alpha_k = L^{-1}$. Summarizing, the gap between the objective values generated during the FISTA iteration and the true optimal value decreases quadratically with respect

to the number of steps taken, although we only use first-order derivatives of the objective function.

However, in order to demonstrate that FISTA is a viable algorithm for sparse recovery, we are more interested in the convergence of $\boldsymbol{u}_k$ and not $F(\boldsymbol{u}_k)$, since for in Example 2.2 we have to retrieve the unknown $\boldsymbol{x}$ instead of $\boldsymbol{\Psi} \cdot \boldsymbol{x}$. To this end [34, Thm. 3] states that the sequence $(\boldsymbol{u}_k)_k$ generated by FISTA satisfies

$$\lim_{k \to \infty} \langle \boldsymbol{u}_k, \boldsymbol{y} \rangle = \langle \boldsymbol{u}^*, \boldsymbol{y} \rangle \quad \text{for all} \quad \boldsymbol{y} \in \mathbb{C}^M. \tag{2.17}$$

This states at least *weak* convergence of $\boldsymbol{u}_k$. However, in practice one usually observes strong convergence in the sense that

$$\lim_{k \to \infty} \|\boldsymbol{u}_k - \boldsymbol{u}\| = 0.$$

However, theoretical results supporting this observation have not been derived yet.

The result in (2.17) is different than the one in Theorem 2.7 in the sense that it does not pose any additional requirements on the CS setting in order to ensure convergence. It is enough the we satisfy the conditions in Theorem 2.3 or Theorem 2.4 and then FISTA provides a suitable algorithm to solve (2.12) by means of (2.14).

To conclude, we refer to some notable variants of FISTA. The Soft-Thresholding with Exact Line Search Algorithm (STELA) [23] uses suitable pseudo-convex approximations of the objective function to find more efficient step-directions and -sizes.

Two-step Iterative Shrinkage-Thresholding Algorithm (TWISTA) presented in [35] uses an operator splitting method to improve upon the convergence rate for ill-conditioned problems.

Also, we again postpone the discussion of the computational complexity to Section 3.2.2, where we present a more efficient variant of FISTA.

2.3.3 Summary

To wrap up, we note that OMP and FISTA are two popular examples for SSR algorithms, which also serve as baseline for many variants [36, 27], derivations and also as benchmarks. This is due to the fact that theoretical guarantees exist and efficient implementations are – as we will see – rather straightforward.

Now that we have two options for recovery at our disposal, we turn to the evaluation and design of the measurement process $\boldsymbol{\Phi}$, which as

we have seen in in the recovery guarantees, is a crucial step during the design of a well performing CS system.

2.4 Measurement Matrix Design

The sections above have introduced requirements on the matrix $\mathbf{\Psi}$ such that we can guarantee successful recovery by means of an optimization problem that can be solved efficiently. Also, we gave conditions for OMP which were also formulated in terms of $\mathbf{\Phi}$ or $\mathbf{\Psi}$ respectively that ensure convergence. The goal of the following section is to give an overview of various known approaches in constructing matrices that satisfy these conditions. This way we present some approaches and methodology how to design and hence possibly optimize the sampling process in CS. As such, this seection gives a general overview of the methods that can be used to tackle AOI-C.

2.4.1 The RIP for Random Matrices

If we reconsider Definition 2.1 for the Restricted Isometry Constant, one can show **[fou]** that we can reformulate it to

$$\delta_S = \max_{\mathcal{S}\subset[M],|\mathcal{S}|\leqslant S} \sigma_{\max}\left(\mathbf{\Psi}_{\mathcal{S}}^{\mathrm{H}}\mathbf{\Psi}_{\mathcal{S}} - \boldsymbol{I}_S\right), \tag{2.18}$$

where $\sigma_{\max}(\boldsymbol{A})$ denotes the largest singular value of $\boldsymbol{A}$. Hence, to determine the restricted isometry constant one has to compute the singular values of $\mathbf{\Psi}_{\mathcal{S}}^{\mathrm{H}}\mathbf{\Psi}_{\mathcal{S}} - \boldsymbol{I}_M$ of for all possible subsets of $[M]$. This is reminiscent of the condition in Remark 2.2, but in a refined way, since the RIP not only evaluates if the projections of sparse vectors always retain the rank of the transformed subspace, but also how much the basis of the subspace gets distorted during the transformation.

There currently is no efficient method to determine $\delta_S(\mathbf{\Psi})$ or a non-trivial upper bound for a general $\mathbf{\Psi}$[37]. This is why the theoretical results that involve statements when a matrix satisfies the RIP resort to probabilistic approaches. These consider $\mathbf{\Psi}$ as a realization of a random variable $\mathbf{\Psi} : \Omega \to \mathbb{C}^{m\times M}$ for some suitable probability space Ω. In the following, we will outline two prominent examples in the form of Gaussian matrices and random Fourier matrices, which we will also reuse in the applications to follow.

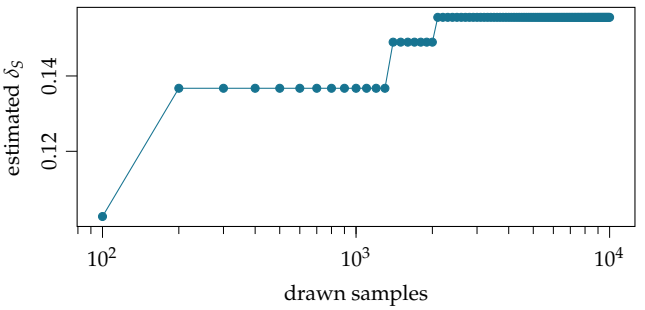


Figure 2.1 *The RIP constant is lower than the bounds suggest.* – The maximal observed value for $\sigma_{\max}\left(\boldsymbol{\Psi}_{\mathcal{S}}^{\mathrm{H}}\boldsymbol{\Psi}_{\mathcal{S}} - \boldsymbol{I}_S\right)$ for the case $N = 4000$, $S = 5$. ■

2.4.1.1 Gaussian Ensemble

For the results presented here, we use slightly less technical variations of statements given in [12]. Given the desired matrix sizes $m, M \in \mathbb{N}$ we define a sample from the Gaussian ensemble as the matrix

$$\boldsymbol{\Psi} = [\psi_{i,j}]_{i,j=1}^{m,M},$$

where the $\psi_{i,j} \sim \mathcal{N}(0,1)$ are independent and identically distributed (iid) random variables drawn from a zero-mean complex valued Gaussian distribution with variance 1. For these matrices we have the following result [12, Thm. 9.6].

Theorem 2.8 (RIP for the Gaussian ensemble)**.** *Let $\boldsymbol{\Psi} \in \mathbb{C}^{m\times M}$ be drawn from the Gaussian ensemble. If*

$$m \geqslant C\delta^{-2}\left(s\ln(eM/S) + \ln(2\varepsilon^{-1})\right),$$

then the restricted isometry constant of $m^{-1/2}\boldsymbol{\Psi}$ satisfies $\delta_s < \delta$ with probability at least $1-\varepsilon$. ■

If we combine this result with the statements in Theorem 2.3 or Theorem 2.7, we can derive conditions on m by setting δ_S to the required values. This way, one can derive probabilistic bounds on m such that recovery with ℓ_1-minimization or approximate ℓ_0 minimization is successful.

However, we would like to stress the fact that this bound is very pessimistic in terms of the constant C. To showcase this, we carry out some simple numerical simulations. If we determine the constant accurately as

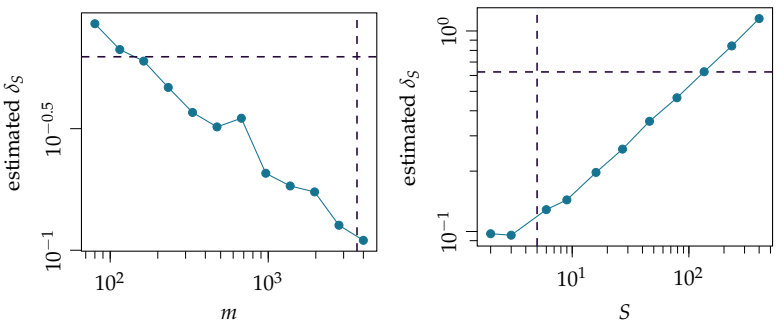


Figure 2.2 *The RIP behaves better than we think.* – Left: the RIP constant for $S = 5$ for given number of rows m of $\mathbf{\Psi}$ together with the critical value $\delta = 0.6246$ and the upper bound stated in (2.19); Right: The RIP constant for varying S for $m = 3646$, where $S = 5$ and the critical value $\delta = 0.6246$ are marked. ■

in [12, p. 293], we get

$$m \geqslant 50 \left(2s \ln(eM/(2S)) + \ln(2\varepsilon^{-1})\right) \tag{2.19}$$

if we want to ensure $\delta_{2S} < 0.6246$. In order to check this empirically we have set $M = 4000$ and $S = 5$ and $1 - \varepsilon = 0.9$. Then the bound above yields $m = 3646$! Which means in order to recover all 5-sparse $\boldsymbol{x}$ from $\boldsymbol{z}$ via ℓ_1-minimization with probability 0.9, we have to take 3646 measurements, since then $\delta_{2S} < 0.6246$. For our numerical experiment, we draw $\eta = 10^4$ trials by forming subsets $\mathcal{S}_i$ of $2S$ columns of a Gaussian $\mathbf{\Psi} \in \mathbb{C}^{3646\times 4000}$ for $i = 1, \ldots, \eta$ and calculate

$$\sigma_i = \sigma_{\max}\left(\mathbf{\Psi}_{\mathcal{S}_i}^{\mathrm{H}} \mathbf{\Psi}_{\mathcal{S}_i} - \boldsymbol{I}_{S_i}\right).$$

In Figure 2.1 we see the running maximum $\sigma_{\max,\mathrm{k}} = \max(\sigma_1, \ldots, \sigma_k)$, which we expect to converge against δ_{2S}. The actual δ_{2S} seems to be significantly lower than 0.6246. Ultimately, this means that the bound as in Theorem 2.8 can merely serve as a scaling law, which indicates how the quantities M, ε and S influence the behavior of the system if they are changed.

To demonstrate this, we conduct the experiment of Figure 2.1 again for varying values of m, while keeping M and S fixed, as well as for varying S while keeping m and M fixed. In both cases we average over 5 trials of random matrices. These results can be found in Figure 2.2. We marked the critical value for δ and also the theoretical lower bound for m, given

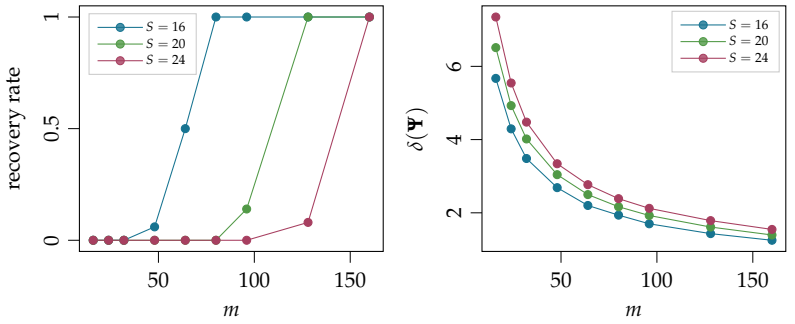


Figure 2.3 *The RIP constant is conservative.* – Monte Carlo simulations for successful recovery by means of ℓ_1-minimization. ■

the previous values for M and S. These experiments suggest that in fact the bounds clearly paint a too pessimistic picture and in practice we can expect better performance from this kind of random matrix ensemble.

2.4.2 The RIP is Conservative

Before studying the bounds which we can achieve for the RIP constant, we would like to put the results in Theorem 2.3 or Theorem 2.7 to the empirical test by carrying out simulations that evaluate how realistic the requirements on $\boldsymbol{\Psi}$ in terms of the RIP constant are. We show that they are very conservative and hence pessimistic.

In Figure 2.3 we carry out Monte Carlo simulations by drawing matrices from the ensemble defined in Section 2.4.1.1 for various values of m and S and for fixed $N = 256$. We average over 50 trials, where each consists of a randomly drawn matrix $\boldsymbol{\Psi} \in \mathbb{R}^{m \times N}$ and a random S-sparse vector $\boldsymbol{x}$. We use the optimization problem in (2.5) which we solve for some $\boldsymbol{u}^*$ by means of an exact general purpose solver. Since we carry out the simulations without any measurement noise, we simply record the empirical probability that $\boldsymbol{x} = \boldsymbol{u}^*$ up to machine precision. As we can see, successful recovery by means of ℓ_1-minimization exposes a very sharp phase transition over m. Additionally we see that at least in the noise-free case, the requirement that $\delta_S < 0.6246$ is very pessimistic. This is due to the fact that the maximum in (2.18) of course also must contain the worst case situation in terms of the support set $\mathcal{S}$.

Together with the analysis carried out in Figure 2.2, we can conclude that the bound derived in (2.19) that defines the means of sampling and

reconstruction in the spirit of Theorem 2.1 and Remark 2.4 is overly pessimistic for practical considerations. Not only does the RIP constant often behave better than expected, but also recovering the sparse signals typically can succeed more reliably than the theory suggests.

2.4.2.1 Subselected Fourier Matrices

The Gaussian matrices have nice and preferable properties when it comes to their reconstruction guarantees, since they provide a sampling scheme that has provable performance. However, when it comes to practical implementations in some cases one has to build the sampling hardware in the analog domain, where random and unstructured matrices pose a problem from a hardware point of view. Usually the compression step is implemented by a network of amplifiers and phase shifters [38, Fig. 1]. In some cases this architecture might be too cumbersome to implement and calibrate, such that the benefits of CS do not outweight these efforts.

Additionally, when it comes to the reconstruction algorithms we will see in Chapter 3 that the computational effort required for the recovery also depends on the degrees of freedom of the measurement matrix. Due to the random nature of the scheme in the previous section, there is no structure one can assume about them to reduce the demands in terms of memory or computation speed during recovery.

In order to overcome these implementational and computational, obstacles one can resort to random matrices that originate from structured orthogonal matrices. In our case, we will focus on randomly subselected Fourier matrices. An analysis of a broader class of subselected bases can be found in [39] and another approach mostly tailored to Fourier matrices in [40]. These random and structured matrices might be able to unify the properties of Gaussian matrices when it comes to reconstruction performance together with easier implementation in terms of hardware or better reconstruction speed. Naturally, we have to expect worse performance in terms of recovery due to the fewer degrees of freedom than completely random matrices.

In the following, we use the so called Fourier matrix $\boldsymbol{F}_n \in \mathbb{C}^{k\times k}$ (or Discrete Fourier Transform (DFT) matrix), which is defined as

$$\boldsymbol{F}_k = \left[\exp\left(-\jmath 2\pi \frac{i\cdot j}{k}\right)\right]_{i,j=1}^{k,k}.$$

Additionally we define the subselection matrix $\mathfrak{S}_C^R$ for two sets $R, C \subset \mathbb{N}$, which is derived from the identity matrix by keeping only the rows

indexed by R and only the columns indexed by C. If either C or R are missing, we do not carry out any subselection along the columns or rows respectively.

In order to keep the presented results short, we only use one statement about randomly subselected Fourier matrices, which bounds the number of rows needed such that the RIP constant is lower than a given threshold.

Theorem 2.9 ([40])**.** *For sufficiently large M and S and some sufficiently small δ, the following holds. For some*

$$m \geqslant C \log^2(1/\delta)\delta^{-2} \cdot S \cdot \log^2(S/\delta) \cdot \log(M)$$

we define the set R such that it has magnitude m and is selected randomly from all subsets of $[M]$ according to the uniform distribution. Then, if we define

$$\Psi = m^{-1/2}(\boldsymbol{F}_M)^R \in \mathbb{C}^{m \times M},$$

the matrix Ψ satisfies the RIP with constant δ and with high probability. ∎

Note that the constant C is not explicitly known as of yet, so a specific bound on the number of measurements such that ℓ_1-minimization can recover the solution of ℓ_0-minimization by means of Ψ cannot be given in terms of hard numbers. This means, a specific scenario is even less predictable in terms of the expected performance of the subselected Fourier matrices and we are again mostly left with empirical studies of the matter and we defer an in depth study of these matrices in Section 4.1.

To shed some light on the scaling behavior of the Fourier matrices we conduct the same numerical experiments as in the previous section (see Figures 2.1 and 2.2) and they are presented in Figure 2.4. We again use $M = 4000$, $S = 5$ and $m = 2000$ for the first plot. As we see, even the more structured Fourier matrices behave better than suggested by the theory and we can have hope that although the bounds seem rather pessimistic in a practical setup, the system behaves better than predicted by the theory.

2.4.2.2 The $\boldsymbol{A}$-RIP for Random Fourier Matrices

In Section 2.2.5, we introduce the extension of the RIP to the case when the sparsifying matrix $\boldsymbol{A}$ is not invertible. Here, we would like to address the question if and how randomly selected Fourier matrices can also satisfy the requirements in Theorem 2.6 for some given $\boldsymbol{A}$.

In case of random Gaussian matrices and orthonormal $\boldsymbol{A}$, this is easy to answer, since the proofs of the results like the one in Theorem 2.8 also

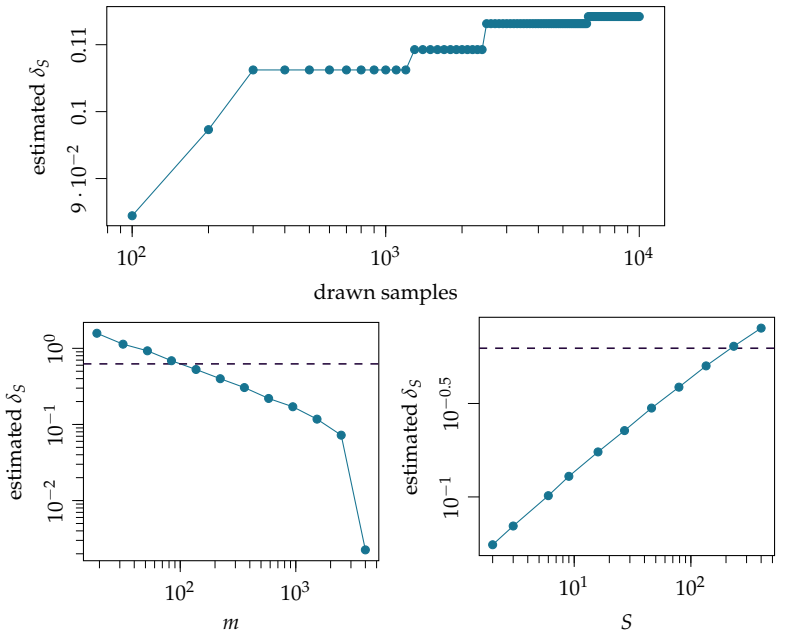


Figure 2.4 *Fourier matrices empirically also behave better than expected.* – Repetition of the results in Figures 2.2 and 2.1. ■

hold if we consider $\boldsymbol{\Psi} = \boldsymbol{\Phi} \cdot \boldsymbol{A}$, which makes the rows of $\boldsymbol{\Psi}$ zero-mean Gaussian random vectors with covariance $\boldsymbol{A}^{\mathrm{H}} \cdot \boldsymbol{A} = \boldsymbol{I}_M$ if we choose $\boldsymbol{\Phi}$ from the Gaussian ensemble as presented in Section 2.4.1.1.

However, the case when $\boldsymbol{A}$ is not orthonormal, or even not provably invertible is more challenging and it will be interesting in Section 4.1. The approach we are taking to address this is geometrically rather interesting. Since we already showed in Theorem 2.9 that subselected Fourier matrices can satisfy the RIP, we can use this result to show that it is a so-called Johnson-Lindenstrauss-embedding.

Definition 2.4 (Johnson-Lindenstrauss-Embedding)**.** A compression matrix $\boldsymbol{A} \in \mathbb{C}^{m \times M}$ is a so-called Johnson-Lindenstrauss-Embedding with constant $\delta > 0$, if for all $\boldsymbol{x}_{i,j} \in \mathcal{S} \subset \mathbb{C}^M$ it holds that

$$(1-\delta)\left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|_2^2 \leqslant \left\|\boldsymbol{A}(\boldsymbol{x}_i - \boldsymbol{x}_j)\right\|_2^2 \leqslant (1+\delta)\left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|_2^2,$$

where $\mathcal{S}$ is a fixed subset of $\mathbb{C}^M$. ∎

If we compare this to the statement given in Definition 2.1, we see that the Johnson-Lindenstrauss-Embeddings are a generalization of the RIP, since for the latter we only consider the set $\mathcal{S}$ to be the set of S-sparse vectors, rendering the differences $\boldsymbol{x}_i - \boldsymbol{x}_j$ then $2S$-sparse vectors. The striking observation now is that a converse observation can also be made, as we see below.

Theorem 2.10 ([41] Thm. 3.1)**.** *Fix $\eta > 0$ and $\varepsilon \in (0,1)$, and consider a finite set $\mathcal{S} \subset \mathbb{C}^M$ of cardinality $|\mathcal{S}| = p$. Set $S \geqslant 40\log(4p/\eta)$, and suppose that $\boldsymbol{\Psi} \in \mathbb{C}^{m \times M}$ satisfies the RIP of order S with constant $\delta \leqslant \varepsilon/4$. Let $\boldsymbol{\xi} \in \mathbb{R}^M$ be a Rademacher sequence, i.e., uniformly distributed on $\{-1,+1\}^M$. Then with probability exceeding $1-\eta$,*

$$(1-\varepsilon)\|\boldsymbol{x}\|_2^2 \leqslant \|\boldsymbol{\Psi}\,\mathrm{diag}(\boldsymbol{\xi})\boldsymbol{x}\|_2^2 \leqslant (1+\varepsilon)\|\boldsymbol{x}\|_2^2$$

for all $\boldsymbol{x} \in \mathcal{S}$. ∎

The interpretation of the above result is that matrices which satisfy the RIP, so wich are not changing the norm of S-sparse vectors significantly, can be turned into Johnson-Lindenstrauss-Embeddings by means of premultiplying them with a so called mixing matrix $\mathrm{diag}(\boldsymbol{\xi})$. Finally, we show an easy to obtain corollary about the RIP of Fourier matrices.

Corollary 2.1. *Let $\boldsymbol{A} \in \mathbb{C}^{M \times N}$ be an arbitrary matrix. Suppose the subselected Fourier matrix $\mathfrak{S}_C \boldsymbol{F}_M$ is chosen such that the requirements of Theorem 2.9 and Theorem 2.10 are satisfied, namely the RIP for constant δ. Then the matrix*

$$\mathcal{S}_C \boldsymbol{F}_M \,\mathrm{diag}(\boldsymbol{\xi})$$

satisfies the $\boldsymbol{A}$-RIP of order $\delta/4$. ∎

Summarizing, we have taken the route from the general RIP for Gaussian matrices, over the RIP for subselected Fourier matrices until we show in Corollary 2.1 that suitably modified these also satisfy the $\boldsymbol{A}$-RIP for coherent sparsifying dictionaries. This serves as a motivation for the sampling scheme proposed in Section 4.1.3.

This is as far as theoretic results can carry us, where explicit bounds and constructions exist in terms of variants of the RIP. However, two main drawbacks originate from considering the RIP. First, it is computationally infeasible to evaluate $\delta_S(\boldsymbol{\Psi})$ and together with its unpleasant geometrical interpretation as suggested by (2.18) it is rendered a quantity that is hard to optimize for. Second, the random ensembles have the disadvantage that when being implemented in practice they show a too unpredictable behavior in terms of estimation performance, when considered for different signals form the space of interest. In order to address these issues, we present some (semi-)deterministic methods to construct measurement matrices with uniformly good performance in the following Section 2.4.3 as well as later in Section 5.3.

2.4.3 Coherence – A Proxy for the RIP

As we have seen, the RIP constant $\delta_S(\boldsymbol{\Psi})$ has a direct influence on the recovery performance but it is hard to to analyze analytically and explicitly. Hence, research soon turned to valid proxies to study the recovery performance of a specific $\boldsymbol{\Psi}$ much like one strives to replace ℓ_0-minimization with ℓ_1-minimization. As we have seen, there exist results like Theorem 2.4 that determine wether ℓ_1-minimization is a valid approach in terms of the coherence $\mu(\boldsymbol{\Psi})$. Similarly to the RIP there is an alternate way to calculate μ via

$$\mu(\boldsymbol{\Psi}) = \left\| \boldsymbol{\Psi}^{\mathrm{H}}\boldsymbol{\Psi} - \boldsymbol{I}_N \right\|_{\infty} \tag{2.20}$$

for a columns normalized matrix $\boldsymbol{\Psi} \in \mathbb{C}^{m\times M}$, where $\|\boldsymbol{A}\|_{\infty}$ denotes the maximum absolute value of the entries in the matrix $\boldsymbol{A}$.

Intuitively, we would like to find a good measurement matrix by means of

$$\min_{\boldsymbol{\Psi}\in\mathbb{C}^{m\times N}} \left\| \boldsymbol{\Psi}^{\mathrm{H}}\boldsymbol{\Psi} - \boldsymbol{I}_N \right\|_{\infty}. \tag{2.21}$$

The desire to solve this optimization problem is not unique to CS. In fact, there are already many iterative construction methods like in [42]

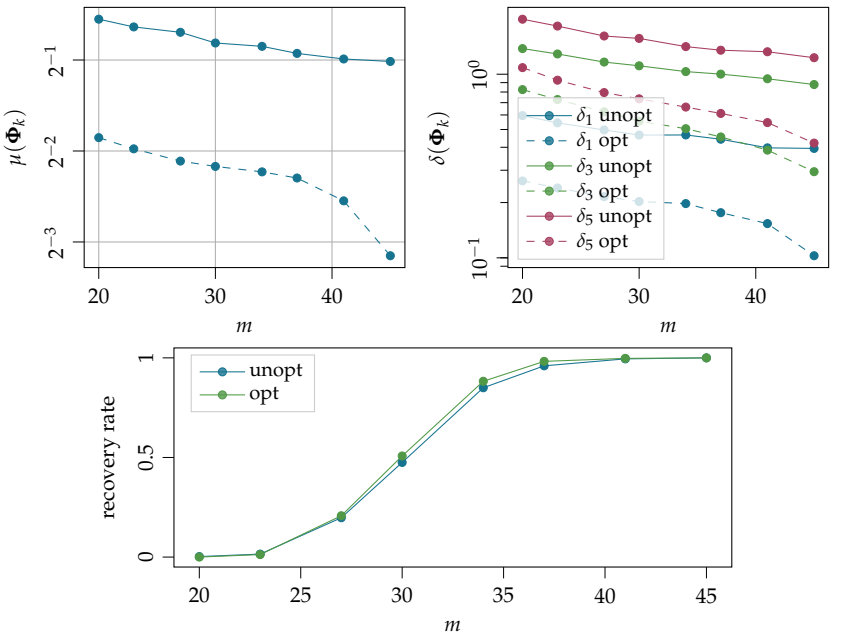


Figure 2.5 *Coherence is a valid proxy for the RIP.* – We consider the case $N = 50$ and study the influence of optimization via spring embedding on the coherence, the RIP constant and rates of recovery. ■

and in coding theory this problem arises in constructing good spherical codes. An overview of theory and algorithms for code design can be found in [43]. Additionally, the connection to the restricted isometry constant can easily be made by

$$\mu(\mathbf{\Psi}) = \delta_2(\mathbf{\Psi}).$$

Moreover, one can show [12, Prop. 6.2] that higher order RIP constants are bounded by

$$\delta_S(\mathbf{\Psi}) \leqslant (S-1)\mu(\mathbf{\Psi}).$$

This means, minimization of the coherence has an influence on the RIP constant as well.

In Figure 2.5 we show that this indeed is the case. To construct the results depicted there, we construct real valued random Gaussian matrices as in Section 2.4.1.1 and then apply a spring embedding algorithm as outlined in [O6]. We use Gaussian random matrices as initializations and

compare the coherence before and after optimization as well as the RIP constants together with the recovery rate of random S-sparse vectors in the noise-free case. This is done for various values of m, while keeping $N = 50$ fixed. In the results one can notice first that although random matrices already have good recovery properties in terms of the RIP constant as well as the coherence, by means of optimization one can still make significant improvements to the measurement process. Additionally, even in the noise-free case recovery rates can be improved slightly by means of optimization.

The mentioned spring embedding algorithm as developed in [O6] cannot deal with the case, when the dictionary $\boldsymbol{A}$ is not invertible. In the case of full-rank but rectangular $\boldsymbol{A}$ one has to modify the optimization routine for the coherence. As outlined in [44] in this case one has to resort to a modified version of (2.21) in order to account for an over-complete dictionary by means of

$$\min_{\boldsymbol{\Phi} \in \mathbb{C}^{m \times N}} \left\| \boldsymbol{A}^{\mathrm{H}} \boldsymbol{\Phi}^{\mathrm{H}} \boldsymbol{\Phi} \boldsymbol{A} - \boldsymbol{H} \right\|_F^2 \quad , \tag{2.22}$$

where $\boldsymbol{H} \in \mathbb{C}^{M \times M}$ is a so-called target matrix and $\|\boldsymbol{A}\|_F = \sqrt{\mathrm{tr}(\boldsymbol{A}^{\mathrm{H}} \boldsymbol{A})}$ denotes the Frobenius norm. There are many flavors of the above problem and an overview can be found in [45]. The authors in [44] use a gradient algorithm to optimize the convex objective function in (2.22) to find an optimal $\boldsymbol{\Phi}$ and a projection onto a suitable convex set to determine the target $\boldsymbol{H}$ on the fly during the optimization. One should notice that the replacement of $\|\cdot\|_\infty$ with $\|\cdot\|_F$ does not directly optimize the coherence anymore, but rather *all* elements of the so-called Gram matrix $\boldsymbol{A}^{\mathrm{H}} \boldsymbol{\Phi}^{\mathrm{H}} \boldsymbol{\Phi} \boldsymbol{A}$ jointly.

The authors in [46] show how to directly optimize the objective in (2.22) explicitly and how to exploit the remaining degrees of freedom. However, the approach is based on the singular value decomposition (SVD) of $\boldsymbol{A}$. For large scale dictionaries (think of Example 2.2) this becomes increasingly unfeasible and unstable. In the grid-free CS setting treated in Chapter 5 this is even more extreme where one deals with infinite dictionaries. Hence, it motivates us to modify the approach in (2.22) to find good measurement matrices for Direction of Arrival (DoA) estimation in Section 5.3 by means of a stochastic and computationally low-cost optimization.

Additionally, we first want to note that there are many different paths one can take in order to construct matrices that have an improved coherence. The manifold of the k-dimensional subspaces of $\mathbb{C}^N$ is studied

in [42] where the authors study suitable packing problems in this manifold. In [47] Fourier matrices are used to construct structured matrices with low coherence. Since coding theory is related to the design of $\boldsymbol{\Psi}$ in [43] the authors take this path to construct well behaved measurement approaches.

Finally, we would like to mention that one can even extend above approach of coherence optimization to the case where $\boldsymbol{\Psi}$ or $\boldsymbol{\Phi}$ have to obey some structural constraints. We refer to Section 2.5.5.2 where we construct Vandermonde matrices that have a low coherence and to [O6] where we treat the case that $\boldsymbol{\Psi}$ is a Khatri-Rao product.

2.4.4 Performance for Parameter Estimation

To introduce anther tool that evaluates the quality of the sensing matrix $\boldsymbol{\Phi}$ we borrowed some concepts from estimation theory, see for instance [26, Ch. 6] for more details. Assume, we are given noisy observations similarly to Example 2.2 in the form

$$\boldsymbol{z} = \boldsymbol{\Phi}\boldsymbol{a}(\boldsymbol{\theta}) + \boldsymbol{n} \in \mathbb{C}^m,$$

where $\boldsymbol{n}$ follows a complex zero-mean Gaussian distribution and covariance $\boldsymbol{R}$. Also, we assume the parameter $\boldsymbol{\theta}$ is unknown. This implies that $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{\Phi}\boldsymbol{a}(\boldsymbol{\theta}), \boldsymbol{R})$ and we wish to estimate the parameter of its mean $\boldsymbol{\Phi}\boldsymbol{a}(\boldsymbol{\theta})$ given the observation $\boldsymbol{z}$. In probability terms, the density of the now modeled as random variable $\boldsymbol{z}$ has an unknown parameter $\boldsymbol{\theta} \in \mathbb{R}^p$ we want to estimate.

To this end, the principle of maximum likelihood formulates the search of $\boldsymbol{\theta}$ as an optimization problem via solving

$$\max_{\theta \in \mathbb{R}^p} \ln p_{\boldsymbol{\theta}}(\boldsymbol{z})$$

for $\boldsymbol{\theta}^*$, where p is the density of the normal distribution with mean $\boldsymbol{\Phi}\boldsymbol{a}(\boldsymbol{\theta})$ and covariance $\boldsymbol{R}$. Since $\boldsymbol{z}$ is random, the values of $\ln p_{\boldsymbol{\theta}}(\boldsymbol{z})$ are random as well, and hence also $\boldsymbol{\theta}^*$. Since the density of the normal distribution is differentiable, we can define the score function

$$\boldsymbol{s}(\boldsymbol{\theta}, \boldsymbol{z}) = \frac{\partial}{\partial \boldsymbol{\theta}} \ln p_{\boldsymbol{\theta}}(\boldsymbol{z})$$

if the model function $\boldsymbol{a}$ is differentiable.

If we consider $\boldsymbol{\theta}^*$ as a random variable, we can ask for this random variable's mean $\boldsymbol{E}(\boldsymbol{\theta}^*)$. If we are in the lucky situation that we can show that in our scenario $\boldsymbol{E}(\boldsymbol{\theta}^*) = \boldsymbol{\theta}$ holds, we may call the estimator that

delivers $\boldsymbol{\theta}^*$ unbiased. If we still consider $\boldsymbol{z}$ as random, we can also study the covariance of the score function, namely

$$\boldsymbol{J}(\boldsymbol{\theta}) = \boldsymbol{E}\left(\boldsymbol{s}(\boldsymbol{\theta}, \boldsymbol{z}) \cdot \boldsymbol{s}(\boldsymbol{\theta}, \boldsymbol{z})^{\mathrm{H}}\right), \tag{2.23}$$

where $\boldsymbol{J}$ is called Fisher Information Matrix (FIM). If we now consider the covariance of the zero-mean random variable $\boldsymbol{\theta}^*$, we have that

$$\boldsymbol{E}\left((\boldsymbol{\theta} - \boldsymbol{\theta}^*)(\boldsymbol{\theta} - \boldsymbol{\theta}^*)^{\mathrm{H}}\right) \succcurlyeq \boldsymbol{J}(\boldsymbol{\theta})^{-1}, \tag{2.24}$$

which is called the Cramér-Rao (lower) Bound [26, Eq. 6.73]. Here $\boldsymbol{A} \succcurlyeq \boldsymbol{B}$ means that the matrix $\boldsymbol{A} - \boldsymbol{B}$ is positive definite. So in this sense, the inverse of $\boldsymbol{J}(\boldsymbol{\theta})$ serves as a lower bound for the error covariance of *any* unbiased estimator $\boldsymbol{\theta}^*$.

For our initial example the expression for the FIM $\boldsymbol{J}(\boldsymbol{\theta})$ can be derived from the Slepian-Bangs formulation [48] and reads as

$$\boldsymbol{J}(\boldsymbol{\theta}) = 2\Re\left\{\frac{\partial \boldsymbol{a}}{\partial \boldsymbol{\theta}}^{\mathrm{H}} \boldsymbol{\Phi}^{\mathrm{H}} \boldsymbol{R}^{-1} \boldsymbol{\Phi} \frac{\partial \boldsymbol{a}}{\partial \boldsymbol{\theta}}\right\}. \tag{2.25}$$

Now in order to assess the performance of $\boldsymbol{\Phi}$, also in a CS setting, one can randomly draw realizations of $\boldsymbol{\theta}$, calculate $\boldsymbol{J}(\boldsymbol{\theta})^{-1}$ and take a suitable average over these values. This gives an indication how $\boldsymbol{\Phi}$ might perform on average over the parameter space $\boldsymbol{\theta}$ resides in, if one had access to an unbiased estimator. Also, the Cramér-Rao (lower) Bound (CRB) allows to detect parameter configurations $\boldsymbol{\theta}$ whose estimation is an inherently ill-posed problem and one might be tempted to take measures to account for this during the design of $\boldsymbol{\Psi}$, which we will showcase for a single example in Section 5.3.3.

In some sense, this is an algorithm independent way of evaluating the measurement process, since it does not depend on the optimization problem employed to recover $\boldsymbol{\theta}$ or $\boldsymbol{a}(\boldsymbol{\theta})$ respectively. Usually, the resulting estimates of $\boldsymbol{a}(\boldsymbol{\theta})$ or $\boldsymbol{\theta}$ are biased due to the recovery method and only become effectively unbiased in the high-Signal-to-Noise Ratio (SNR) regime. Hence, the findings using the CRB generally cannot be used directly to infer the performance of a complete CS system. However, in Section 4.1 we present a system whose behavior can be predicted reasonably well by the CRB.

2.4.5 Conclusion

After having worked through this chapter, it should be clear that the quest for constructing the optimal sensing strategy in terms of the measurement matrix $\boldsymbol{\Psi}$ in a given scenario has no single path to the princess' castle. In fact, depending on the requirements on the systems, the results may vary.

If one desires fast reconstruction with decent performance, one should resort to randomly sub-sampled Fourier matrices, since they obey provable bounds for their quality and also offer the possibility for a fast matrix-vector product, which as we will see in Section 3.2 is a beneficial property. In case of parameter estimation and when the estimation performance should be the same over the parameter regime, several non-linear optimization methods should be employed to improve upon randomly drawn matrices. However, as we have argued, these pose some obstacles when implemented directly in the analog domain.

As we have seen, the results that infer lower bounds on the necessary number of measurement m, always depend on the complexity of the signal in terms of the sparsity $S = \|\boldsymbol{x}\|_0$, which hence has to be known before being able to estimate a bound for m. Motivated by the fact that the previous sections greatly emphasize the importance of sparsity, the next section proposes an estimator for the sparsity prior to the reconstruction of $\boldsymbol{x}$, in order to make more educated decisions that depend on the complexity of the scenario at hand.

2.5 Single-Snapshot Sparsity Order Estimation

In this chapter, we study the problem of estimating the unknown degree of sparsity from compressive measurements without the need to carry out a sparse recovery step. This is useful, since many steps in the reconstruction and measurement pipeline depend on the sparsity of the signal. Which is why it would be advantageous, if the sparsity of the signal was known as early as possible during the processing pipeline.

Given the model in (2.3) as

$$\boldsymbol{z} = \boldsymbol{\Psi} \cdot \boldsymbol{x} = \boldsymbol{\Phi} \cdot \boldsymbol{A} \cdot \boldsymbol{x},$$

we wish to estimate $S = \|\boldsymbol{x}\|_0$ directly from $\boldsymbol{z}$ before reconstructing $\boldsymbol{x}$.

It is shown that specially designed measurement matrices allow to rearrange the measurement vector $\boldsymbol{z}$ into a matrix such that its effective rank coincides with the effective sparsity order. In fact, it is proven that

matrices which are composed of a Khatri-Rao product of smaller matrices generate measurements that allow to infer the sparsity order. Moreover, if some samples z_i are used more than once, one of the matrices needs to be Vandermonde. These structural constraints reduce the degrees of freedom in choosing the measurement matrix which may incur a degradation in the achievable coherence. Thus, we also address suitable choices of the measurement matrices. In particular, Khatri-Rao and Vandermonde matrices are analyzed in terms of their coherence and a new design for Vandermonde matrices that achieves a low coherence, which also is computationally cheap, is proposed.

2.5.1 Introduction

As we have seen in Section 2.4 the number m of measurements required to efficiently reconstruct $\boldsymbol{x}$ depends on its sparsity order S, which is typically not known when acquiring a signal. Therefore, one usually determines m according to some upper bound $S \leqslant S_{\max}$ which must be known a priori in order to design the measurement as indicated by (2.19).

However, since S may vary quite significantly for a fixed CS scenario but different $\boldsymbol{x}$ currently observed, such an approach can lead to systems that are too conservative, i.e., they take more measurements than necessary. Therefore, being able to estimate and monitor the sparsity order of a signal would be an attractive feature of a CS system. It would allow to adjust the number of measurements to the current sparsity of the signal which might vary in time (e.g., with the number of transmissions in cognitive radio [49] or with the complexity of an image in CS-based image acquisition [50]).

In some applications of CS one can acquire multiple measurement vectors (MMV) which enjoy a joint sparsity in a given basis. In this case the observation model for $K \in \mathbb{N}$ snapshots reads as

$$\boldsymbol{Z} = \boldsymbol{\Psi} \cdot [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_K] = \boldsymbol{\Psi} \cdot \boldsymbol{X},$$

where $\boldsymbol{Z} \in \mathbb{C}^{m \times K}$ and $\boldsymbol{X} \in \mathbb{C}^{N \times K}$. Also one assumes that all columns $\boldsymbol{x}_k$ of $\boldsymbol{X}$ have the same support $\mathcal{S} = \operatorname{supp}(\boldsymbol{x}_1)$ and the matrix $\boldsymbol{X}$ has Kruskal rank $\geqslant S$.

In order to estimate $|\mathcal{S}|$ one can use an estimator for the rank of $\boldsymbol{Z}$, given some simple assumptions about $\boldsymbol{\Psi}$. However, we do not want to treat this case, since such a setting is more the exception than the rule. It is for instance true in scenarios where we observe modulated signals that enjoy sparsity in some domain. Examples for this setting include

DOA/TDOA estimation (see Examples 2.2 and 5.1), where sparsity is in the angular/delay domain and the signal modulation at the transmitter provides the amplitude variations.

The MMV case is also prominent in certain image processing applications where patches of images may enjoy a sparse representation in the same basis. However, in this case, the support is not constant across snapshots, which renders a sparsity order estimation over several vectors relatively meaningless. That said, in an MMV setting, the sparsity order estimation is much simpler since we can infer the sparsity order directly from the effective rank of the observation matrix. As we discuss in this chapter, this case has been treated in earlier publications, like [51] and [52, 53, 54].

Below we address the much more challenging single measurement vector (SMV) case. Note that this is much more common, as in most applications we only have one set of observations of the phenomenon. Taking another set of observations at another time may either be too time-consuming or it may not provide a linearly independent sparse representation as the coefficients do not change unless the source applies some modulation.

From this discussion it becomes clear that MMV techniques are not applicable to the SMV setting. The proposed framework will show that by applying Khatri-Rao structured measurement matrices it is possible to transform the challenging SMV case into a virtual MMV case where then rank estimation schemes can be applied, similar to a MMV setting. Moreover, knowing the sparsity order $\|\boldsymbol{x}\|_0$ allows to improve reconstruction algorithms by tuning algorithm-specific parameters such as the regularization parameter λ in (2.14) of the necessary number of steps for running OMP as indicated by Theorem 2.7.

2.5.2 Related Work

Due to the prominent role the sparsity order plays in sparse signal recovery, the lack of knowledge of the sparsity order has been recognized as a fundamental gap between theory and practice [55, 56]. Early papers on this subject have proposed to employ sequential measurements [56] and cross-validation type techniques [36, 55] where sequential reconstructions of the signal are considered. Similarly, [10] shows that the sparsity order can be estimated from the reconstruction, stating bounds on the number of measurements that are required for this step. However, the bounds are only found numerically, and the reconstruction process

involves cumbersome optimization problems. As the following results show, this can be avoided by estimating the sparsity order directly based on the compressed observations.

A different approach is taken in [57], where the authors show that a specifically tailored measurement procedure which consists of a Cauchy and a Gaussian distributed measurement matrix allows to estimate a continuous measure of sparsity given by the ratio of squared one- and two-norm of the signal. However, this measure is not equal to the sparsity order. In fact, it is continuous and hence needs to be rounded to an integer number (which is not discussed in [57]). Moreover, the measurement process is very restrictive since the distribution of the measurement matrices is pre-specified. Finally, according to the authors one parameter of these distributions should be chosen with respect to the noise variance, which might be unknown, whereas the results presented here do not require the noise's variance to be known.

The authors in [58] propose to use sparse sensing matrices since these allow to infer the degree of sparsity of a signal from the degree of sparsity of the measurement. The resulting estimator has a very low complexity. However, it is only approaching the true sparsity in the large system limit and hence not applicable to lower-dimensional problems. Moreover, the proposed measurement matrices incur a certain performance degradation at the reconstruction stage due to the somewhat higher coherence, which is imposed by the restriction to sparse matrices.

A link between sparsity order estimation and rank estimation was put forward in [51] for the MMV setting, which has also been studied both for the stationary case [52, 53] as well as the case of time-varying support for block-stationary signals [54]. However, these approaches require the already mentioned stationarity in the support pattern of $\boldsymbol{X}$ ins the MMV case as well as a temporal variation in the coefficients of the sparse representation to create linearly independent observations. This limits their applicability in many practical problems.

2.5.3 Contributions in this Thesis

We introduce a method for estimating the sparsity order of a signal from a single snapshot of the compressive measurement. In particular, rearrangements of the observation vector into a matrix are considered and it is shown under which conditions the rank of this matrix coincides with the sparsity order of the unknown signal. Thereby, the sparsity order can be estimated by applying any known rank estimation scheme [59]. Since

there exist many efficient algorithms which estimate the model order in presence of perturbations such as noise, this approach allows us to handle noisy measurements as well as the case of approximate sparsity. By doing so, we contribute to the AOI-M aspect of CS.

The proposed approach only requires the measurement matrix to possess a Khatri-Rao structure, which leaves considerable room for optimizing their choice. Moreover, in the case of overlapping blocks, one of the factors needs to be a Vandermonde matrix. Therefore, the second part of this chapter discusses the design of measurement matrices in presence of these structural constraints. To facilitate the sparsity order estimation as well as the sparse reconstruction, the factors need to possess a low coherence, as we have outlined in (2.8). Moreover, in the presence of a Khatri-Rao structure, it is best to optimize the factors of the Khatri-Rao product independently. Therefore we investigate the coherence of Vandermonde matrices and a new design algorithm is proposed, which efficiently constructs Vandermonde matrices with low coherence. Additionally, simple upper and lower bounds for the resulting coherence of this algorithm are derived. Summarizing, these results contribute to AOI-C as defined in Section 1.2.

2.5.4 Sparsity Order Estimation Algorithm

In this section, we consider a special form of (2.3), which reads as

$$\boldsymbol{z} = \boldsymbol{\Phi} \cdot \boldsymbol{A} \cdot \boldsymbol{x} + \boldsymbol{n}, \tag{2.26}$$

where $\boldsymbol{x} \in \mathbb{C}^N$ is S-sparse, $\boldsymbol{A} \in \mathbb{C}^{N \times N}$ is a sparsifying *basis*, and $\boldsymbol{\Phi} \in \mathbb{C}^{m \times N}$ with $m \ll N$ is the measurement matrix as before. This means we assume explicitly that $\boldsymbol{A}$ is invertible. So for the exposition presented here, it suffices to consider the case $\boldsymbol{A} = \boldsymbol{I}_N$ without loss of generality, since $\boldsymbol{\Phi}$ can be chosen freely and can always be replaced by $\bar{\boldsymbol{\Phi}} = \boldsymbol{\Phi} \cdot \boldsymbol{A}^{-1}$ to account for an $\boldsymbol{A} \neq \boldsymbol{I}_N$. Moreover, for clarity the noise-free case $\boldsymbol{n} = \boldsymbol{0}$ is considered first. The role of additive noise is discussed in Section 2.5.4.3.

2.5.4.1 Sparsity order estimation for non-overlapping blocks

In order to estimate the sparsity order S directly from $\boldsymbol{z}$, we propose to consider rearrangements of $\boldsymbol{z}$ into a matrix. Specifically, divide $\boldsymbol{z}$ into $k \in \mathbb{N}$ blocks $\boldsymbol{z}_i$ of length $\ell \in N$ where the i-th block is given by

$$\boldsymbol{z}_i = \begin{bmatrix} z_{1+p\cdot(i-1)}, & z_{2+p\cdot(i-1)}, & \ldots, z_{\ell+p\cdot(i-1)} \end{bmatrix}^{\mathrm{T}} \in \mathbb{C}^{\ell} \tag{2.27}$$

for $i = 1, 2, \ldots, k$, where $p \in \mathbb{N}$ specifies by how many samples consecutive blocks are advancing. It is clear from Equation (2.27) that for $p = \ell$ the blocks $\boldsymbol{z}_i$ do not overlap whereas for $p < \ell$ the overlap grows with decreasing p, up to the case of maximum overlap without producing the same column twice for $p = 1$. Moreover, since m samples are available, the parameters ℓ, p, k, m should satisfy $\ell + p \cdot (k-1) = m$.

The blocks $\boldsymbol{z}_i$ can be used to form the columns of a matrix

$$\boldsymbol{Z} = \begin{bmatrix} \boldsymbol{z}_1, & \boldsymbol{z}_2, & \ldots, & \boldsymbol{z}_k \end{bmatrix} \in \mathbb{C}^{\ell \times k}.$$

Moreover, define submatrices $\boldsymbol{\Phi}_i$ by selecting the corresponding rows from $\boldsymbol{\Phi}$ such that

$$\boldsymbol{z}_i = \boldsymbol{\Phi}_i \cdot \boldsymbol{x}. \tag{2.28}$$

The main idea of the proposed approach is to show that for a suitably chosen $\boldsymbol{\Phi}$, it holds that $\operatorname{rk} \boldsymbol{Z} = S$ for any S-sparse $\boldsymbol{x}$ and therefore, the sparsity order can be inferred from the rank of $\boldsymbol{Z}$.

The following theorem summarizes the conditions on $\boldsymbol{\Phi}$ to facilitate the Sparsity Order Estimation (SOE) for the case of non-overlapping blocks ($p = \ell$).

Theorem 2.11. *For k blocks of the measurement $\boldsymbol{z}$ of length $\ell = m/k$ and any $r \leqslant \min(k, \ell)$ the following statements are equivalent.*

1. *For all $s \leqslant r$ and all $\boldsymbol{x}$ with $|\operatorname{supp}\{\boldsymbol{x}\}| = s$ it holds that $\operatorname{rk} \boldsymbol{Z} = s$.*
2. *$\boldsymbol{\Phi} = \boldsymbol{U} \diamond \boldsymbol{W}$ for some $\boldsymbol{U} \in \mathbb{C}^{k \times N}$, $\boldsymbol{W} \in \mathbb{C}^{\ell \times N}$ with $\operatorname{rk}^* \boldsymbol{U} \geqslant r$ and $\operatorname{rk}^* \boldsymbol{W} \geqslant r$.*

∎

Note that $\operatorname{rk}^* \boldsymbol{A}$ is the Kruskal-rank of the matrix $\boldsymbol{A}$. The proof for this result can be found in Appendix A.2.1. In other words, Theorem 2.11 states that in the case of non-overlapping blocks, the sparsity order S

can be obtained from the rank of $\boldsymbol{Z} \in \mathbb{C}^{\ell \times k}$ if and only if the measurement matrix $\boldsymbol{\Phi}$ possesses a Khatri-Rao structure. Moreover, the theorem also shows that the highest sparsity order possible to estimate is given by $S_{\max} = \min(\mathrm{rk}^*(\boldsymbol{U}), \mathrm{rk}^*(\boldsymbol{W})) \leqslant \min(k, \ell)$. In the case of non-overlapping blocks it follows that $k \cdot \ell = m$ which means $S_{\max} \leqslant \min(k, \frac{m}{k}) \leqslant \sqrt{m}$ and one should choose k and ℓ close to $\sqrt{m}$ to maximize $S_{\max}$.

2.5.4.2 Sparsity order estimation for overlapping blocks

As has been shown for non-overlapping blocks, in the case where $\boldsymbol{\Phi}$ is a Khatri-Rao product of two matrices of equal size, the maximum sparsity order that can be estimated is $S_{\max} \leqslant \sqrt{m}$. This bound is tight, iff m is a square number $m = q^2$; then set $k = \ell = q$. For overlapping block the size of the matrix $\boldsymbol{B}$ grows, which allows to estimate larger sparsity orders $S_{\max}$. However, depending on the overlap, additional constraints on $\boldsymbol{\Phi}$ have to be posed.

One of these constraints is that one factor has Vandermonde structure, which are defined below.

Definition 2.5. Let $\mathcal{V}^{n \times m}(\mathbb{C})$ be the space of $n \times m$ Vandermonde matrices. Then it holds that the non-linear mapping

$$\boldsymbol{\nu}_n : \mathbb{C}^m \to \mathcal{V}^{n \times m} \quad \text{with} \quad \boldsymbol{v} \mapsto \begin{pmatrix} v_1 & \dots & v_m \\ v_1^2 & \dots & v_m^2 \\ \vdots & \vdots & \vdots \\ v_1^n & \dots & v_m^n \end{pmatrix} \tag{2.29}$$

is bijective. So a Vandermonde matrix $\boldsymbol{V}$ it completely defined by its first row. In the following, the v_i are called the generating elements of $\boldsymbol{V}$. ∎

Now, we can formulate the following theorem for the procedure of SOE for overlapping blocks, whose proof can be found in Appendix A.2.2.

Theorem 2.12. *For k overlapping blocks of length ℓ, block advance p and any $r \leqslant \min(k, \ell)$ the following statements are equivalent:*

1. *For all $s \leqslant r$ and all s-sparse $\boldsymbol{x}$ it holds that $\mathrm{rk}\, \boldsymbol{Z} = s$.*
2. *$\boldsymbol{\Phi}$ consists of the first m rows of $\boldsymbol{V} \diamond \boldsymbol{W}$ with $\boldsymbol{W} \in \mathbb{C}^{p \times N}$ being arbitrary, $\boldsymbol{V} \in \mathbb{C}^{\lceil m/p \rceil \times N}$ being a Vandermonde matrix such that the matrix $\hat{\boldsymbol{\Phi}}$*

originating from restricting $\boldsymbol{V}_{\lceil \ell/p \rceil} \diamond \boldsymbol{W}$ *to its first* ℓ *rows has Kruskal rank* r *and the matrix* $\boldsymbol{V}$ *restricted to its first* k *rows also has Kruskal rank* r*.*

■

The assumptions in 2 in Theorem 2.12 on $\boldsymbol{V}$ and $\boldsymbol{W}$ seem rather strict and technical. The following will show that they are fulfilled given some simple criteria. In case when the Vandermonde matrix $\boldsymbol{V}$ is considered it is enough to require that the generating elements $v_1, \ldots, v_N$ are pairwise distinct for $\boldsymbol{V}$ to have maximal Kruskal rank. For a square Vandermonde matrix $\boldsymbol{V} \in \mathbb{C}^{n \times n}$ with generating elements $v_1, \ldots, v_n$ the determinant reads as

$$\det(\boldsymbol{V}) = \prod_{1 \leqslant i < j \leqslant n} (v_i - v_j).$$

This means that a Vandermonde matrix with more rows than columns and pairwise distinct generating elements always has full column rank. Moreover, if the matrix has more columns than rows, and pairwise distinct generating elements, every square submatrix has full rank and as such the whole matrix has full Kruskal rank.

Now turning to the arbitrary factor $\boldsymbol{W} \in \mathbb{C}^{p \times N}$, choose it in general position and the following reasoning shows that this is sufficient for the requirements of Theorem 2.12. To this end, fix some r-sparse $\boldsymbol{x} \in \mathbb{C}^N$ with support set $\mathcal{S}$, a Vandermonde matrix $\boldsymbol{V} \in \mathbb{C}^{m \times N}$ which has full Kruskal rank and consider the Khatri-Rao product $\boldsymbol{V} \diamond \boldsymbol{W} \in \mathbb{C}^{m \cdot p \times N}$. Now assume that

$$\boldsymbol{m} = (\boldsymbol{V} \diamond \boldsymbol{W}) \cdot \boldsymbol{x} = \boldsymbol{0}.$$

If the vector $\boldsymbol{m}$ is reshaped according to the block construction in Theorem 2.12 to

$$\boldsymbol{M} = \boldsymbol{W} \operatorname{diag}(\boldsymbol{x}) \boldsymbol{V}^{\mathrm{T}} = \widetilde{\boldsymbol{W}} \boldsymbol{V}^{\mathrm{T}},$$

which is a rank decomposition of $\boldsymbol{M}$ and this implies that for non-zero $\boldsymbol{x}$ the matrix $\boldsymbol{M}$ is non-zero.

Note that the special case $p = 1$ in Theorem 2.12 implies that the entire sensing matrix $\boldsymbol{A}$ is a Vandermonde matrix with rescaled columns. In the context of harmonic retrieval, the mapping from $\boldsymbol{b}$ to $\boldsymbol{B}$ is also known as spatial smoothing [60] and is applied as a preprocessing step for subspace-based estimators in order to decorrelate coherent signals.

Remark 2.5 (On the choice of k, ℓ and p)**.** Regarding the choice of the parameters observe that $S_{\max} \leqslant \min(k, \ell)$ where k and ℓ satisfy $(k-1) \cdot p + \ell = m$. Therefore, for a given block advance p, the number of blocks k is equal to $k = (m-\ell)/p + 1$. This means that to maximize $S_{\max}$ one should choose ℓ as the closest integer to $(m+p)(p+1)$, which leads to $k \approx \ell$. Obviously, larger amounts of overlap (corresponding to smaller values of p) lead to a higher maximum sparsity order $S_{\max}$ where the maximum overlap case $p = 1$ corresponds to $S_{\max} = \lfloor (m+1)/2 \rfloor$. This shows that there is a fundamental tradeoff between the SOE stage and the SSR stage: while a larger amount of overlaps improves the SOE capability, it leads to a more rigidly structured measurement matrix with a higher coherence, which is detrimental to the SSR step. The achievable coherence is analyzed in more depth in Section 2.5.5. ∎

2.5.4.3 Sparsity Order Estimation in the presence of noise

In the presence of additive, say Gaussian, noise as in equation (2.26) the entries of the matrix $\boldsymbol{Z}$ are disturbed with a Gaussian noise matrix $\boldsymbol{N}$ as well, i.e.

$$\hat{\boldsymbol{Z}} = \boldsymbol{Z} + \boldsymbol{N}. \tag{2.30}$$

It is easy to see that the rank of $\hat{\boldsymbol{Z}}$ is maximal with probability 1. It is still possible to determine the "effective" rank for a model like (2.30) if the statistics of the additive noise are known. Interestingly, although $\boldsymbol{N}$ is a reshaped version of the vector $\boldsymbol{n}$ in (2.26) with possibly the same noise sample at multiple positions (depending on the overlap), it can be shown that if the noise samples in $\boldsymbol{n}$ are i.i.d., the noise matrix $\boldsymbol{N}$ is "white" in the sense that $\mathbf{E}\left(\boldsymbol{N}\boldsymbol{N}^{\mathrm{H}}\right) = C \cdot \boldsymbol{I}_{\ell}$. To this end, let us assume the elements in $\boldsymbol{n}$ have zero mean and variance 1 and let us define the selection matrices

$$\boldsymbol{J}_i = \left[\mathbf{0}_{\ell \times p(i-1)}, \quad \boldsymbol{I}_{\ell \times \ell}, \quad \mathbf{0}_{\ell \times (N-\ell-p(i-1))}\right] \in \mathbb{R}^{\ell \times m},$$

which satisfy $\boldsymbol{J}_i \cdot \boldsymbol{J}_i^{\mathrm{H}} = \boldsymbol{I}_{\ell \times \ell}$. Then one can rewrite the measurement and noise vectors as

$$\boldsymbol{z}_i = \boldsymbol{J}_i \cdot \boldsymbol{z} \quad \text{and} \quad \boldsymbol{n}_i = \boldsymbol{J}_i \cdot \boldsymbol{n},$$

where $i = [k]$. Now calculate

$$\mathbf{E}\left(\boldsymbol{N} \cdot \boldsymbol{N}^{\mathrm{H}}\right) = \mathbf{E}\left(\sum_{i=1}^{k} \boldsymbol{J}_i \cdot \boldsymbol{n} \cdot (\boldsymbol{J}_i \cdot \boldsymbol{n})^{\mathrm{H}}\right) = \sum_{i=1}^{k} \boldsymbol{J}_i \cdot \mathbf{E}\left(\boldsymbol{n} \cdot \boldsymbol{n}^{\mathrm{H}}\right) \cdot \boldsymbol{J}_i^{\mathrm{H}}$$

$$= \sum_{i=1}^{k} \boldsymbol{J}_i \cdot \boldsymbol{J}_i^{\mathrm{H}} = k \cdot \boldsymbol{I}_\ell. \tag{2.31}$$

Summarizing the rank of the matrix $\boldsymbol{Z}$ has to be determined in presence of noise according to (2.30) (here, the additive noise is white). This task is known as model order selection and a number of efficient algorithms are available. Examples include information-theoretic criteria such as MDL, AIC, BIC (see [59] for a survey), the Eigenvalue Threshold Test (ETT) [52] or the Exponential Fitting Test (EFT) presented in [61]. Here, the latter is used for the numerical experiments in Section 2.5.6, because it is derived specifically for models disturbed by additive white Gaussian measurement noise.

2.5.5 Sensing Matrix Design

As derived in the last section, sparsity order estimation can be achieved via rank estimation of a matrix obtained by rearranging the measurement vector, provided that the sensing matrix obeys certain structural constraints. Firstly, it has to be a Khatri-Rao product of two smaller matrices and secondly, in the case of overlapping blocks, one of the blocks has to be a Vandermonde matrix.

In this section, the implications of this particular sensing matrix structure for the design of the measurement matrix are analyzed, with a particular focus on the Vandermonde matrices.

2.5.5.1 Khatri-Rao structured measurement matrix optimization

The results in Theorem 2.11 show that in the case of no overlap one is able to recover the sparsity order by using a Khatri-Rao structured sensing matrix, whose factors have to fulfill the condition of having a high Kruskal rank. This condition is difficult to optimize for, since the Kruskal rank is hard to compute. However, the inequality in (2.10) links it to the coherence defined in (2.7). Therefore, for the Kruskal rank of the factors to be high, one should make use of matrices with low coherence. These are desirable also from the viewpoint of the subsequent SSR step, which in general works better the lower the coherence of the sensing matrix is. In fact, we have presented Theorem 2.4 which states that the under-determined system of equations $\boldsymbol{z} = \boldsymbol{\Psi} \cdot \boldsymbol{x}$ has a unique solution for S-sparse vectors $\boldsymbol{x}$ if $S < S_{\max}$ with

$$S_{\max} = \frac{1}{2}\left(1 + \frac{1}{\mu(\boldsymbol{\Psi})}\right). \tag{2.32}$$

Note that BP(see (2.5)) as well as OMP(see Algorithm 2.1) are able to achieve this bound in the sense that they can recover any S-sparse $\boldsymbol{x}$ in the noise-free case as long as $S < S_{\max}$. For these reasons we aim at minimizing the coherence of the measurement matrix $\boldsymbol{\Psi}$.

Since in the case of SOE as presented here $\boldsymbol{\Psi}$ is a Khatri-Rao product, it might seem advantageous to take this structure into account when optimizing the coherence of the sensing matrix using the two factors. In the real valued setting, one can use packing arguments in projective matrix spaces of rank 1 to optimize the coherence of Khatri-Rao products, as developed in [O6] or [42]. As a result one obtains that the best coherence of a Khatri-Rao product $\boldsymbol{\Phi} = \boldsymbol{U} \diamond \boldsymbol{W} \in \mathbb{R}^{m \times N}$ is achieved if $\boldsymbol{U} \in \mathbb{R}^{m_1 \times N}$ and $\boldsymbol{W} \in \mathbb{R}^{m_2 \times N}$ contain repeated columns according to $\boldsymbol{U} = \hat{\boldsymbol{U}} \otimes \mathbf{1}_{1 \times N_2}$ and $\boldsymbol{W} = \mathbf{1}_{1 \times N_1} \otimes \hat{\boldsymbol{W}}$ where $\hat{\boldsymbol{U}} \in^{m_1 \times N_1}$, $\hat{\boldsymbol{W}} \in^{m_2 \times N_2}$ with $m = m_1 \cdot m_2$ and $N = N_1 \cdot N_2$. This case yields $\boldsymbol{\Phi} = \boldsymbol{U} \diamond \boldsymbol{W} = \hat{\boldsymbol{U}} \otimes \hat{\boldsymbol{W}}$ so that $\boldsymbol{\Phi}$ is actually Kronecker structured.

Due to the repeating columns, this implies that both $\boldsymbol{U}$ and $\boldsymbol{W}$ have Kruskal rank equal to 1, i.e. Khatri-Rao products of minimal coherence have factors that are not suitable for SOE when done as proposed in Theorems 2.11 and 2.12. Instead, one should optimize the coherence of the factors $\boldsymbol{U}$ and $\boldsymbol{W}$ independently in order to ensure the highest Kruskal rank possible.

This also has a positive effect on the coherence of $\boldsymbol{\Phi}$ since the trivial upper bound

$$\mu(\boldsymbol{\Phi}) = \mu(\boldsymbol{U} \diamond \boldsymbol{W}) \leq \mu(\boldsymbol{U} \otimes \boldsymbol{W}) = \mu(\boldsymbol{U})\mu(\boldsymbol{W})$$

holds. Therefore, the remainder of this section discusses the minimization of coherence of the matrices $\boldsymbol{U}$ and $\boldsymbol{W}$.

In the case of no overlap during the construction of $\boldsymbol{Z}$ these matrices can be arbitrary as there are no further structural requirements from the SOE method. One can therefore apply any method for coherence minimization from Section 2.4. This means that the only regime where any improvement upon existing results is possible is in the case of overlap during the construction of the matrix $\boldsymbol{Z}$ as in Theorem 2.12. Here, one factor obeys a Vandermonde structure. As long as the generating elements of the matrix $\boldsymbol{V}$ are pairwise different it is well known that this matrix has maximal Kruskal rank. But since a low coherence is crucial for efficient recovery to happen, the following section contains a more thorough analysis of this problem.

2.5.5.2 Coherence minimization for Vandermonde matrices

In case of overlap of the blocks $\boldsymbol{z}_i$ to form $\boldsymbol{Z}$ we have seen there appears the strong requirement of the Vandermonde structure for one of the Khatri-Rao factors. However, this structure also allows us allow us to derive an explicit term for the inner product of two columns of a Vandermonde matrix that does not involve any summation and just depends on the amplitudes and phases of the generating elements v_i. This is key, because of the definition of the coherence $\mu(\boldsymbol{A})$ of a matrix given in (2.7).

Theorem A.1 in Appendix A.2.3 allows us to derive a construction for orthogonal Vandermonde matrices, which can be considered a generalization of the fact that the Fourier matrix is orthogonal. The Fourier matrix is, depending on the definition, a scaled Vandermonde matrix, where the generating elements are placed on a regular grid on the unit circle, i.e. $\exp(2\pi ik/n)$, $k \in [n-1]$. The corollary below additionally allows a uniform phase shift of the generating elements.

Corollary 2.2. *Let $v_1, \ldots, v_n$ be chosen with absolute value 1 such that*

$$\arg\left(v_i v_j^*\right) = \frac{(i-j)2\pi}{n} \quad \text{for} \quad 1 \leqslant j < i \leqslant n.$$

Then for the matrix $\boldsymbol{V} = \boldsymbol{\nu}_n(v_1, \ldots, v_n)$ it follows that

$$\mu(\boldsymbol{V}) = 0.$$

■

But in the scenario typical for compressed sensing the involved matrices have more columns than rows and thus it is impossible for them to be orthogonal. Hence one should minimize the coherence as far as possible in this case as well. The following algorithm, which is also one of the central novelties in this work, makes use of Theorem A.1 to achieve this.

The general geometric idea is twofold. First, the upper envelope κ, see Theorem A.1, for the absolute value of inner product of two columns of a Vandermonde matrix in Theorem A.1 suggests that a large angular distance between generating elements yields a lower inner product, since it is monotonically decreasing on $(0, \pi)$. To maximize the mutual angular difference between each pair of generating elements, place them on a regular angular grid on $[0, 2\pi]$. Second, facts 3 and 5 in Theorem A.1 suggest that the inner product of two columns is minimized if their generating elements have amplitudes that are reciprocals of each other or, geometrically speaking, are reflections on the unit circle in $\mathbb{C}$ of each other.

Data: Dimensions n and m, Positive constant c;
[1] Set $\hat{m} = 2\lceil m/2 \rceil$, $c_1 = c$ and $c_2 = 1/cy$.;
[2] Set $\phi_k = 4\pi\frac{k-1}{\hat{m}}$ and $z_k = c_1 \cdot e^{i\cdot\phi_k}$ for $k \in [\hat{m}/2]$.;
[3] Set $\phi_k = 4\pi\frac{k-1}{\hat{m}} + 2\pi\frac{1}{\hat{m}}$ and $z_k = c_2 \cdot e^{i\cdot\phi_k}$ for $k \in \{\hat{m}/2+1, \ldots, \hat{m}\}$.;
[4] Return the Vandermonde matrix $\boldsymbol{V} = v(z_1, \ldots, z_m) \in \mathbb{C}^{n\times m}$.;

Algorithm 2.3: *Vandermonde matrices with low coherence can be constructed explicitly.* – Algorithm to construct Vandermonde matrices with low coherence that obey the coherence bound in Theorems 2.13 and 2.14.

Combining the above two observations yields the method as outlined in Algorithm 2.3. Loosely speaking, it places the angularly nearest neighbors among the generating elements on different sides of the complex unit circle and the ratio between the amplitudes is the parameter $c > 0$. The algorithm above still depends on the input parameter c, but the optimal value can be computed with a simple bisection algorithm and depends only on n and m. This allows it to be stored in a persistent lookup table.

Additionally bounds on the coherence are necessary to evaluate the performance of compressed sensing algorithms. The bounds exploit the fact that the above algorithms output is deterministic and thus the upper and lower envelope κ and η from Theorem A.1 can be made use of. The following two Theorems deal with the upper and lower bounds respectively.

A lower bound on the coherence can be used together with equation (2.32) to derive lower bounds on the number of measurements required in order to achieve a recovery guarantee for a given sparsity order $S_{\max}$.

Theorem 2.13. *Given n, $m > n$, $c \in (0,1)$, κ as in Theorem A.1 and the output $\boldsymbol{V}$ of Algorithm 2.3 for inputs n, m and c it holds that*

$$\mu(\boldsymbol{V}) \geqslant \max\Big\{\sqrt{\kappa(c, c, 4\pi/m)}, \sqrt{\kappa(c, 1/c, 2\pi/m)}\Big\}. \tag{2.33}$$

■

Conversely equation (2.32) and upper coherence bounds can be used to derive upper bounds on the number of measurements required to guarantee successful recovery in scenarios with a certain level of sparsity.

Theorem 2.14. *Define the function* $u : (0,1] \times \mathbb{N} \to \mathbb{R}$ *by*

$$u(c,m) = \begin{cases} \max\{\eta(c,c,4\pi/m), \eta(1/c,c,2\pi/m)\} \\ \quad \textit{for} \quad m < 2n, \\ \max\{\eta(c,c,4\pi/m), \lambda(1/c,c,2\pi/m)\} \\ \quad \textit{for} \quad 2n \leqslant m \leqslant 4n, \\ \max\{\lambda(c,c,4\pi/m), \lambda(1/c,c,2\pi/m)\} \\ \quad \textit{for} \quad m > 4n, \end{cases}$$

and call Algorithm 2.3 with parameters n, m *and* c, *then for its output* $\boldsymbol{V}$ *it follows that*

$$\mu(\boldsymbol{V}) \leqslant \sqrt{u(c,m)},$$

where η *and* λ *are defined as in Theorem A.1.* ∎

Summarizing, it is possible to derive an algorithm for Vandermonde matrices with proveable behavior in terms of the coherence. Up to this point we have presented theoretical considerations that outline the theoretic framework for the proposed SOE methodology. Next, we would like to put this approach to a test.

2.5.6 Testing the Sparsity Order Estimation Procedure

This chapter is dedicated to empirical investigations for showcasing the performance of the proposed methods. To this end, consider measurement scenarios as in (2.26). In order to generate the ground truths $\boldsymbol{x}$, the amplitudes on the support of $\boldsymbol{x} \in \mathbb{C}^N$ for $N = 512$ are drawn i.i.d. from the set $\{\pm 1 \pm i, \pm 1 \mp i\}$ according to a uniform distribution. All simulations are carried out with m noisy measurements where the components of the additive noise vector $\boldsymbol{n} \in \mathbb{C}^m$ were drawn independently from a zero mean circularly symmetric Gaussian distribution with variance σ^2. Depending on the situation at hand, the construction of $\boldsymbol{\Phi}$ changes accordingly, which is chosen once for each scenario and kept fixed when sampling the ground truths for a certain scenario type and size. Moreover, 2000 trials for each scenario and level of noise variance were simulated.

[1] Choose the number of measurements m and the overlap parameter p.;
[2] Select k and ℓ according to Remark 2.5.;
[3] Depending on the value of p construct $\boldsymbol{\Phi} \in \mathbb{C}^{m \times N}$ according to the requirement in Theorem 2.11 and Theorem 2.12.;
[4] For each acquired measurement $\boldsymbol{z}$, arrange its entries in matrix $\boldsymbol{Z}$ as described in (2.27).;
[5] Estimate the rank of $\boldsymbol{Z}$ using the EFT or ETT algorithm to estimate the sparsity order of the signal resulting in the measurement $\boldsymbol{z}$, where in both algorithms a singular value decomposition of $\boldsymbol{Z}$ has to be calculated.;

Algorithm 2.4: Summary for the proposed process of SOE

2.5.6.1 Sparsity Order Estimation

To depict the performance of the proposed method for SOE as summarized in Algorithm 2.4, the procedure is simulated for various combinations of parameters m and p. Here, the sparsity order of the ground truth $\boldsymbol{x} \in \mathbb{C}^N$ is set to 8. After setting the number of measurements m and the overlap p, select k and ℓ in order to maximize the size of the reshaped matrix $\boldsymbol{Z}$ for optimal performance according to Remark 2.5. The Vandermonde factors in the columnwise Kronecker product are constructed by Algorithm 2.3 and the unstructured factors are drawn once from a Gaussian ensemble, which means that all elements are drawn independently from a Gaussian distribution with variance 1. Both factors in each of the occurring instances of a columnwise Kronecker product have normalized columns, resulting in the product having normalized columns as well.

To study the influence of the overlap the parameter p is varied for fixed signal size N and number of measurements m while adapting k and ℓ appropriately. For the estimation of the effective rank of $\boldsymbol{Z}$, the ETT [52] applied to the described scenario with a target false rejection rate of 0.005. The training stage of this model order selection method is the computationally most time consuming part, but this procedure only has to be done once in advance and one has to make sure that enough training data is generated such that the actual false rejection rate during application is close to the one imposed during the training stage. If the training data were insufficient, one would significantly deteriorate the estimation performance of the proposed SOE method.

Figure 2.6 displays the empirical mean of the estimated sparsity order across all trials for a varying levels of the noise variance for the method proposed here and the one in [57]. The performance metric employed

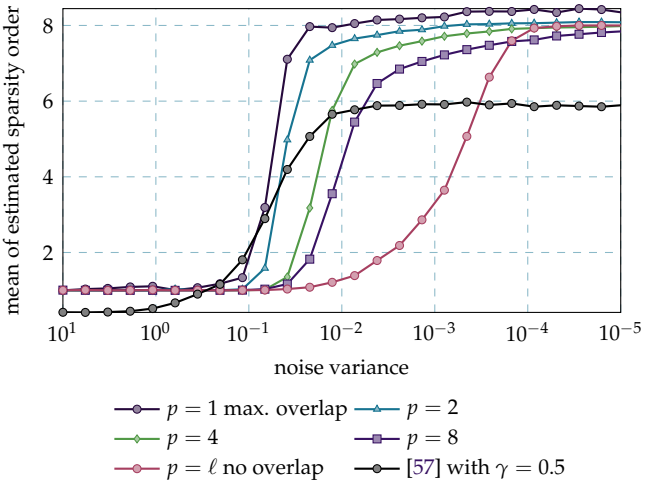


Figure 2.6 *The block overlap during the construction of $\boldsymbol{Z}$ has substantial influence on the performance of SOE* – Overlap p vs. sparsity order estimation performance with $m = 121$ measurements. ■

here allows to detect a possible phase transition with respect to the noise variance and also allows to detect possible a bias present for the analyzed estimation routines.

The algorithm in [57] also provides means to estimate the sparsity order from a single snapshot, where the measurement matrix $\boldsymbol{\Phi} \in \mathbb{R}^{m \times N}$ consists of $m_1 = \lceil m/2 \rceil$ rows with zero mean and scale γ Cauchy distributed entries and $m_2 = m - m_1$ rows with centered Gaussian entries of variance γ, where γ is set to 0.5. As expected for the pure task of SOE the case $p = 1$ displays the best performance, because $\min\{k, \ell\}$ is maximal. For increasing $p \geqslant 2$, including the case $p = \ell$, the phase transitions happens at an increasing level of noise variance and the transition itself is not as sharp as in the cases of overlap. In comparison to the method of [57], which displays a similar phase transition, but also saturates into a too low region around 6 instead of the true value 8.

To examine a more realistic processing pipeline where the true sparsity order is not known in advance but just a more or less sharp upper bound, the process of SOE is carried out in advance of the reconstruction. This preprocessing step can be used to provide information about the underlying signal's sparsity. Then one can carry out the actual reconstruction. As a means of reconstruction OMP, see Section 2.3.1, is used. If one

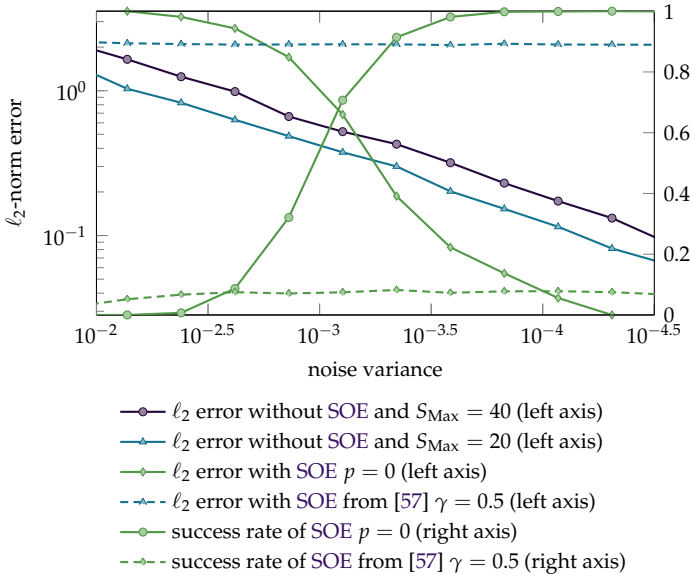


Figure 2.7 *As soon as SOE works, the SSR step is also improved substantially.* – Influence of SOE on reconstruction quality when using estimated sparsity order as parameter for OMP in case of $m = 169$. ■

makes use of this algorithm, one has to provide a useful stopping criterion, for instance, in form of a maximum number of iterations to be run, which in our case is the estimated sparsity order $\hat{S}$ from the preprocessing step. This approach is compared to an "unguided" OMP reconstruction, where only rough upper bounds $S_{\text{Max}} = 40$ and $S_{\text{Max}} = 20$ are known and used as stopping criteria. Here, the ℓ^2-norm error $\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|_2^2$ between reconstruction $\hat{\boldsymbol{x}}$ and ground truths $\boldsymbol{x}$ is measured and averaged of the trials.

This error metric is chosen, because first it is the most widely used and second many of the performance guarantees for various reconstruction algorithms in compressed sensing are formulated with respect to it, as in Theorem 2.7. Moreover, to include comparable results obtained by another method, the algorithm in [57] is applied to this scenario as well.

As one can see in Figure 2.7, where the success rate of SOE and the resulting ℓ_2 error after reconstruction are displayed in a combined plot, as soon as SOE starts to work, the reconstruction error decreases significantly below the one of the "unguided" reconstruction. So in this case the

additional computational effort for SOE results in a significant increase in reconstruction precision. Additionally, the results of the method proposed in [57] show that the measurement process imposed by it does not deliver comparable results, since for increasing noise variance the reconstruction precision does not improve anymore and is saturated very quickly. This might seem surprising at first, but the authors of [58] made the same observations. The reason for this is that part of the measurement matrix consists of Cauchy distributed entries, which influence the behavior of the system with respect to the noise level in the observed fashion. Moreover the probability of estimating the correct support size is also very low and does not improve with decreasing noise variance.

As a last study, the reconstruction performance of the matrices that obey the structural constraints imposed by the proposed means of SOE are investigated. To this end, a scenario with a priori known sparsity order is simulated to remove the influence of the SOE procedure on the reconstruction process. Here one should expect differing performance since for decreasing p, the highly structured Vandermonde block in the sensing matrix becomes more dominant thus increasing the overall coherence of $\mathbf{\Phi}$. The results are depicted in Figure 2.8. Here one can see that the case of no overlap $p = \ell$ comes close to the Gaussian measurement matrices despite the imposed columnwise Kronecker structure. Moreover, one notices that the Vandermonde factor in the columnwise Kronecker product for the case of $p < \ell$ is the reason for a worse reconstruction performance.

Taking Figures 2.6 and 2.8 into account, one can see that the choice of the parameter p is crucial and a trade-off between the performances of reconstruction and SOE has to be maintained according to the requirements given by the specific application at hand. When precice reconstruction is neccessary, one should choose $p = \ell$ and lower values, if one is more interested in carrying out SOE.

2.5.6.2 Vandermonde Matrices

As derived before, in the case of block overlap the proposed method of SOE requires that one of the factors has a Vandermonde structure and motivated by that Algorithm 2.3 provides matrices which deliver good reconstruction performance even with these rigid structural constraints. The numerical investigations are concluded by a presentation of results that display the gain achieved through Algorithm 2.3 compared to several other methods for constructing Vandermonde structured sensing matri-

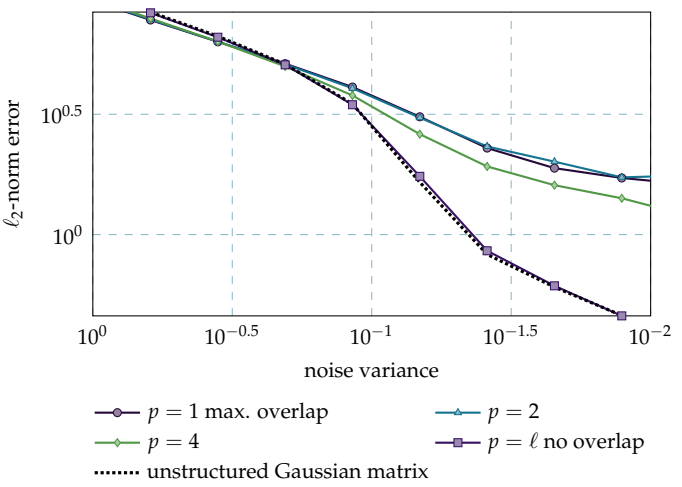


Figure 2.8 *The lower the overlap defined by p the better the reconstruction performance.* – Influence of overlap p on the reconstruction stage using OMP with $m = 100$ measurements. ■

ces and also compared to unstructured matrices drawn from a Gaussian ensemble as a baseline, since they are well known for their good performance during reconstruction as outined in Section 2.4.1.1. Again OMP is used as a means of reconstruction, because it is known that the coherence of the involved sensing matrix determines the performance and stability of OMP.

In the following, reconstruction results are presented which assess the performance of various ways to construct Vandermonde structures as compression matrices where no Khatri-Rao product is present; these results are of course not related to sparsity order estimation but the purpose of those simulations is to isolate the effects of the different construction methods for Vandermonde matrices by analyzing their performance when used as compression matrices in a very synthetic CS scenario.

For comparison, Gaussian sensing matrices are constructed by drawing each entry identically and independently from a complex zero mean circular symmetric distribution. Then each column is normalized to unit length. This ensemble is compared with three types of Vandermonde matrices: (a) independently drawing their generating elements $v_1, \ldots, v_N$ from $Y \sim \exp(\jmath 2\pi U)$, where $U \sim \mathrm{Unif}[0,1]$; (b) deterministically constructing the generating elements by placing them on a regular grid on

the complex unit circle, so $v_k = \exp(\jmath 2\pi(k-1)/N)$; (c) according to Algorithm 2.3. As in the case of the Gaussian matrices, each column of the Vandermonde matrices is scaled to unit length.

The support sets of the underlying signals $\boldsymbol{x}$ are restricted such that the minimum distance between nonzero elements is larger than a certain quantity. To motivate this structural assumption, note that Vandermonde matrices have highly correlated columns, if their generating elements' projections on the complex unit circle are closely located. This has already been established in Theorem A.1, where it is shown that

$$\lambda(c_1, c_2, \arg(v_1 - v_2)) \to 1 \quad \text{for} \quad |\arg(v_1 - v_2)| \to 0.$$

Clearly, if one increases the number of generating elements for the mentioned methods (b) or (c) and fixes one arbitrary generating element, the distance with respect to the adjacent generating elements' arguments decreases and thus the coherence of the resulting $\boldsymbol{V}$ gets closer to 1. In terms of recovering a sparse vector $\boldsymbol{x}$ in this context means that if two distinct $i, j \in \operatorname{supp}(\boldsymbol{x})$ are close to each other in the sense that $|i-j|/m \leqslant \varepsilon(n)$, then stably recovering this vector becomes impossible. This drawback originates from the structural impositions on $\boldsymbol{V}$ and as such cannot be circumvented.

In the context where the argument of the generating elements correspond to a parameter $\vartheta \in [0, 2\pi)$ and the columns of $\boldsymbol{V}$ are atoms described by this parameter, e.g. [9], one can think of the number of columns m of $\boldsymbol{V}$ as a granularity on $[0, 2\pi)$. So increasing m yields a finer grid on the parameter set, or in terms of the function λ a finer discrete sampling. However, this does *not* decrease $\varepsilon(n)$ from above and as such does not reduce the distance between two parameters that are present in $\boldsymbol{x}$ and one still can resolve during reconstruction.

To include these observations in the simulations, the set of possible supports of the underlying signals $\boldsymbol{x}$ is restricted. This is done by considering a Vandermonde matrix $\boldsymbol{V} \in \mathbb{C}^{n \times m}$ with its m generating elements regularly placed on the unit circle. Then support patterns are sampled such that the resulting ground truth $\boldsymbol{x}$ only contains columns such that their generating elements' distances with respect to their arguments exceeds $2\pi/n$. This corresponds to making an assumption about the structure of the occurring signals and is known under the notion structured sparsity [62].

The results in Figure 2.9 display the effect of Algorithm 2.3 on the rate of correct support detection during reconstruction. This is a valid performance measure for reconstruction schemes that produce exactly sparse

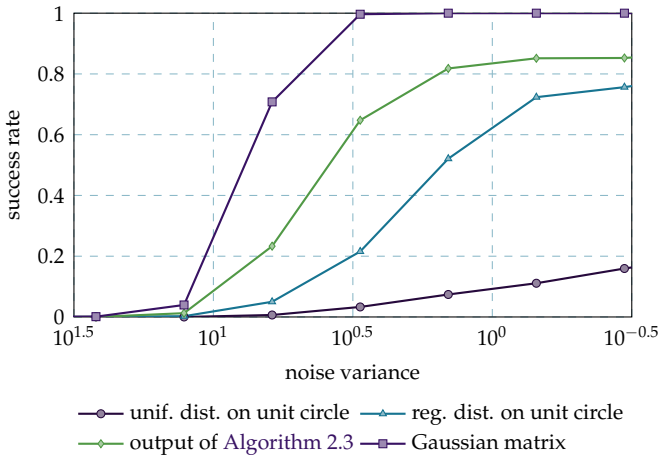


Figure 2.9 *Algorithm 2.3 can substantially improve the performance of Vandermonde matrices when used during reconstruction.* – Comparison of correct support detection during reconstruction using various Vandermonde algorithms against the Gaussian ensemble with $m = 96$ measurements. ■

results, like the OMP algorithm does, since it chooses the amplitudes on the detected support from the solution to the least squares problem on the span of the selected columns. The proposed method yields matrices that improve on the performance of the Vandermonde matrices with elements placed on a uniform grid on the complex unit circle. It is also clear why not all supports are correctly detected, since some of them have non separable elements due to the small distance of the corresponding generating elements of the Vandermonde matrices, which in the presence of noise results in reconstruction errors.

As a second performance metric the error in the ℓ_2-norm between reconstruction $\hat{\boldsymbol{x}}$ and ground truth $\boldsymbol{x}$ is measured. These results can be found in Figure 2.10. Here, the numerical findings correspond to those above and the proposed method displays a lower reconstruction error than the other two means of constructing Vandermonde matrices. As expected, no instance of these highly structured matrices can compete with the performance displayed by matrices from the Gaussian ensemble, which do not have to fulfill any structural requirements.

Finally, Figure 2.11 illustrates the performance of the matrices proposed by [57] when just used for sparse reconstruction by means of OMP. In this case one can see that these matrices run into a lower error. This means

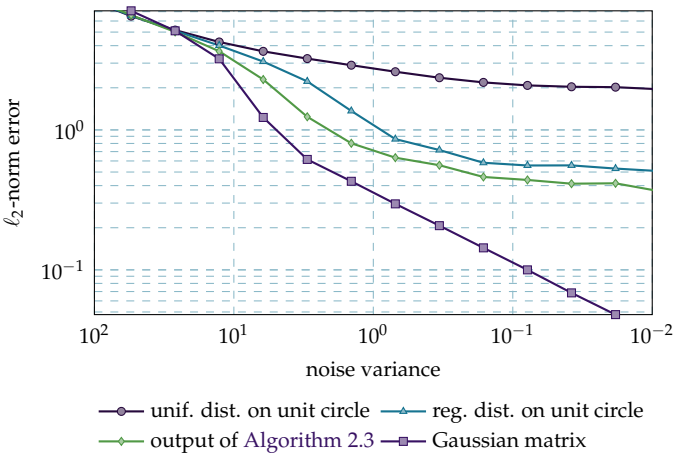


Figure 2.10 *Also for the ℓ_2-distance Algorithm 2.3 delivers good results.* – Comparison in terms of the ℓ_2 norm between the ground truth and its reconstruction for various Vandermonde algorithms and the Gaussian ensemble with $m = 96$ measurements. ■

that despite their possible use for SOE they cannot be recommended for the SSR step afterwards to estimate $\boldsymbol{x}$.

2.5.7 Conclusions

To conclude, it is indeed possible to reliably estimate the sparsity order from a single compressed measurement vector. Hence, we successfully contributed to AOI-M. As we show the performance of this estimation process depends on the coherence of the matrices involved in the Khatri-Rao structured sensing matrix. Additionally, the procedure allows a trade-off between estimation accuracy of the sparsity order reconstruction quality via SSR algorithms by means of the overlap parameter p. A larger amount of overlap allows a higher sparsity order to be estimated but also leads to more stringent structural constraints on the measurement matrix.

The Khatri-Rao and Vandermonde constraints on the compressed sensing measurement motivate us to analyze the suitable choice of the measurement matrices that allows the proposed sparsity order estimation and yet achieves a low coherence. As we have shown, these two goals go hand in hand. In particular, the achievable coherence of Vandermonde matrices can be optimized by the proposed procedure compared to a ran-

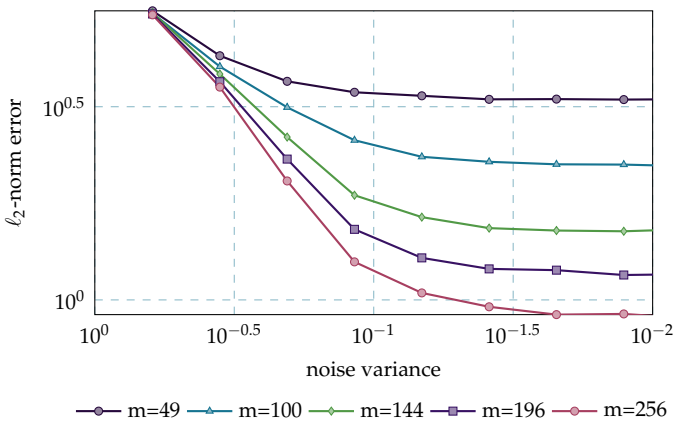


Figure 2.11 *The matrices proposed by [57] do not deliver estimation errors that scale well with the noise level.* – The reconstruction performance of the matrices, which are proposed by [57] for SOE using OMP. ■

dom approach and the intuitive deterministic approach. These results establish an improvement with respect to AOI-C.

The presented numerical results demonstrate the trade-off between the maximal sparsity order that can be estimated and the coherence of the corresponding measurement matrices. Moreover we show that the usage of SOE as a pre-processing step to some SSR algorithm can substantially improve upon the reconstruction error compared to an unguided reconstruction with unknown sparsity order. Concluding, these results directly address AOI-R. Finally, they clearly show the benefit of the proposed low-coherence Vandermonde matrix design.

2.6 Summary

The first parts of this chapter outline how Compressed Sensing can be viewed as an alternative sampling theory in addition to the traditional and well known Nyquist sampling and we show how to design a processing pipeline from compressed signal acquisition over reconstruction methods to evaluation metrics to determine the performance of a given CS system. We have outlined how plentiful the possibilities are when choosing the measurement and estimation procedures compared to conventional Nyquist sampling and also that these degrees of freedom have

implications on the signal recovery guarantees one can provide as well. As such we have already outlined the relation of the four AOI introduced in Section 1.2.

Due to the increased complexity in terms of non-linear reconstruction the metrics that allow to infer successful signal estimation seem at first inaccessible, since the RIP constant is equally hard to compute as the solution to ℓ_0-minimization. This motivates the search for viable proxies. As we show these can be found in terms of the coherence μ as a performance metric and ℓ_1-minimization or approximate ℓ_0-minimization for substitute reconstruction approaches.

This diversity for sampling and reconstruction that is enabled by CS is the reason for the wide range of applications that can be addressed with it. For instance, we have seen the freedom that is allowed during the sampling allows to estimate the model order of a signal prior to reconstruction, while the proposed sampling scheme still enables efficient and reliable signal estimation afterwards.

In the next chapter we focus on one approach to significantly speed up the reconstruction process of the acquired signal and bring ease to the modeling step of the whole CS system by means of a library to efficiently handle linear mappings.

Chapter 3

Matrix-Free Sparse Signal Recovery

> "It's very hard to find your own words – and you don't actually exist until you have your own words."
>
> *Jordan B. Peterson*

Chapter 2 has shown that a crucial step during the processing of a CS pipeline is the reconstruction of the signal $\boldsymbol{y}$ or the sparse vector $\boldsymbol{x}$ which were acquired by means of

$$\boldsymbol{z} = \boldsymbol{\Psi x} = \boldsymbol{\Phi A x} = \boldsymbol{\Phi y}.$$

Not only do $\boldsymbol{\Phi}$ and $\boldsymbol{\Psi}$ determine the accuracy and robustness of the reconstruction procedure but as we have seen in Algorithm 2.1 and Algorithm 2.2, they are heavily involved in the actually used algorithm for estimating $\boldsymbol{x}$ or $\boldsymbol{y}$. The following chapter is devoted to first presenting an alternative representation for linear mappings in general. Secondly, we show how this representation facilitates the development of even more efficient reconstruction schemes in terms of computational effort by extending Algorithm 2.1 and Algorithm 2.2 to make use of this more efficient representation.

Finally, we present the architecture of a free software package, which makes the definition, and usage of this alternative representation very convenient. Hence, it make these concepts readily available for very efficient and flexible Sparse Signal Recovery and beyond.

3.1 Motivation and Introduction

If we remember Example 2.2, concerning delay estimation for another time, we see that the dimension of $\boldsymbol{A} \in \mathbb{C}^{N \times M}$ depends on the number of parameter grid points M in the delay domain as well as the number of frequency samples N. Both quantities increase when one strives for higher estimation accuracy or a higher bandwidth during measurements. Likewise, the size of the compression matrix $\boldsymbol{\Phi} \in \mathbb{C}^{m \times N}$ also increases according to the scaling laws derived in Section 2.4. Although this specific and still quite simple example will most likely not scale to a point where modern computers exhaust their memory or computation capabilities, this easily happens when acquiring volumetric data as in Section 4.1 or a lot of samples per second as in Section 4.2.

Such large scale linear models are not a new phenomenon, since they arise when solving partial differential equations [63] or when studying large graph structures [64][1]. Often, they are also derived from models of physical processes [65, 66] or arise in day to day signal processing tasks [67].

Additionally, these types of problems have already been solvable for several decades despite the large size of the involved matrices. The reason for this is that for these algorithms the occurring matrices are not required to be represented as 2D arrays. Instead, one only needs access to the *linear action* of the matrices to arbitrary vectors, which is usually tailored to the specific properties of the matrices and represented by a specialized algorithm.

To facilitate the introduction of this alternative representation of linear mappings, the following paragraphs takes the converse direction of what we plan to do in this chapter. As such, it demonstrates why and how matrices are able to encode linear mappings between finite dimensional spaces.

[1] See `https://sparse.tamu.edu/`

Where do matrices come from?

Consider two vector spaces U and V over the complex field $\mathbb{C}$. Note that this not (yet) means that elements in U and V are denoted by tuples of complex numbers. This is indicated by the fact that elements in U and V are not bold-faced. Due to the vector space structure, both in U and V, we can form so-called linear combinations

$$u = \lambda_1 u_1 + \cdots + \lambda_k u_k \in U$$

for a finite number of scalars $\lambda_i \in \mathbb{C}$ and vectors $u_1, \ldots, u_k \in U$. Likewise for V. In other words, the spaces U and V are closed with respect to these linear combinations. Additionally, there is one special vector in U (as well as one in V) which we denote as 0_U, which satisfies $u + 0_U = u$ for every $u \in U$ – the zero-vector. Now, we consider a set of vectors $B_U = \{u_1, u_2, \ldots\} \subset U$ linearly independent, if

$$0_U = \sum_i \lambda_i u_i$$

can only hold, if $0 = \lambda_1 = \lambda_2 = ...$ is satisfied.

Next, we say that $\text{span}(B_U)$ is the set of all possible finite linear combinations of the form

$$u = \sum_{k=1}^{N_u} \lambda_k u_{i_k} \quad \text{for} \quad \boldsymbol{u}_{i_k} \in B_U,$$

where $N_u \in \mathbb{N}$. If B_U is linearly independent and $\text{span}(B_U) = U$, we call B_U a basis of U. One can show that the cardinality of all bases of a vector space is the same, which allows us to define the dimension of U as $|B_U|$ for some arbitrary basis B_U of U. We assume from here on that $|B_U| = D_U < \infty$ and $|B_V| = D_V < \infty$. Continuing, a linear mapping $f : U \to V$ has the property that

$$f(\lambda_1 u_1 + \lambda_2 u_2) = \lambda_1 f(u_1) + \lambda_2 f(u_2).$$

This means that the image of linear combinations of vectors is the linear combination of these vectors' images. As such, linear mappings fit to the linear vector space structure. Assume further, we are given two bases B_U and B_V in U and V respectively. Now, we can write any vector $u \in U$ as

$$u = \lambda_1 b_{U,1} + \cdots + \lambda_m b_{U,D_u} \in U.$$

The vector $\boldsymbol{\lambda} \in \mathbb{C}^{N_U}$(!) is called the coordinate vector of u with respect to the basis B_U. If we now apply f to u we get due to the linearity of f that

$$f(u) = \lambda_1 f(b_{U,1}) + \cdots + \lambda_{D_U} (b_{U,D_U}) \in V.$$

Since we also have a basis in V, we can express every $f(b_{U,i})$ in the coordinates with respect to the basis in V via

$$f(b_{U,i}) = t_{1,i} b_{V,1} + \cdots + t_{D_V,i} b_{V,D_V},$$

which we can plug back into the expression for $f(u)$ and after collecting the summand coefficients for each $b_{V,i}$ we get

$$f(u) = \left(\sum_{i=1}^{N_U} \lambda_i t_{1,i} \right) b_{V,1} + \cdots + \left(\sum_{i=1}^{N_U} \lambda_i t_{n,i} \right) b_{V,n} \in V.$$

If we define $\boldsymbol{\mu} \in \mathbb{C}^{N_V}$ to be the coordinate vector in V of $f(u)$ with respect to the basis B_V we can now simply write

$$\boldsymbol{\mu} = \boldsymbol{T} \cdot \boldsymbol{\lambda}, \tag{3.1}$$

where the 2D array $\boldsymbol{T} \in \mathbb{C}^{N_V \times N_U}$ contains the entry in the i-th row and the j-th equal to $t_{i,j}$ and we make use of the conventional matrix-vector product.

In other words, we can encode the linear mapping f by means of the matrix $\boldsymbol{T}$ as soon as we have fixed B_U and B_V. This means that we can identify $\boldsymbol{T}$ with f as long as we are clear about the chosen bases. Many tools from linear algebra now allow us to infer properties of f based on properties of $\boldsymbol{T}$. ∎

This concludes our detour into basic linear algebra and we are now able to easily formulate our alternate representation for linear mappings, which does not follow the reasoning above.

How to not use Matrices

As we have seen by now, it is generally not necessary to encode a linear mapping by means of a suitable matrix. Alternatively, one can use a *matrix-free* representation of the linear operator at hand. For a given linear mapping $f : \mathbb{C}^{n_2} \to \mathbb{C}^{n_1}$, this is achieved by not storing the *dense* representation (3.1) $\boldsymbol{T}$ in as the 2D array that generally uses $\mathcal{O}(n_1 \cdot n_2)$ memory, in terms of the big $\mathcal{O}$ notation. Instead, we only provide two linear functions

$$\phi_{\boldsymbol{T}} : \mathbb{C}^{n_2} \to \mathbb{C}^{n_1} \quad \text{and} \quad \beta_{\boldsymbol{T}} : \mathbb{C}^{n_1} \to \mathbb{C}^{n_2} \tag{3.2}$$

such that the equalities $\boldsymbol{x} \mapsto \phi_{\boldsymbol{T}}(\boldsymbol{x}) = \boldsymbol{T} \cdot \boldsymbol{x}$ and $\boldsymbol{y} \mapsto \beta_{\boldsymbol{T}}(\boldsymbol{x}) = \boldsymbol{T}^{\mathrm{H}} \cdot \boldsymbol{x}$ hold, if we had access to $\boldsymbol{T}$. Here we call ϕ the forward transform and β the backward transform.

It is important to notice that we only require $\phi_{\boldsymbol{T}}$ to reproduce the action of $\boldsymbol{T}$ on a vector $\boldsymbol{x}$, but we do not specify how this has to be achieved. This means that the function $\phi_{\boldsymbol{T}}$ can – and usually should – be implemented differently than simply calculating

$$y_i = \phi_{\boldsymbol{T}}(x)_i = \sum_{k=1}^{n_2} b_{i,k} \cdot x_k. \tag{3.3}$$

This is the reason why this representation is matrix-free, since the calculation of the forward and backward transform does not depend on the explicit knowledge of the encoding matrix $\boldsymbol{T}$ to represent a linear mapping f. It is clear that $\phi_{\boldsymbol{T}} = c \circ f \circ c^{-1}$, where c denotes the mapping between the vector space and its coordinate space. Additionally, we also require independent definition and implementation of $\beta_{\boldsymbol{T}}$, since many numerical algorithms depend both on the forward and the backward transform of $\boldsymbol{T}$ or f respectively.

Additionally, we note that we slightly abuse notation when applying these transforms to matrices, where we naturally demand that

$$\phi_{\boldsymbol{T}}(\boldsymbol{C}) = \boldsymbol{T} \cdot \boldsymbol{C}$$

This way we can argue that both representations are equivalent, since we can recover $\boldsymbol{T}$ from $\phi_{\boldsymbol{T}}$ by means of $\boldsymbol{T} = \phi(\boldsymbol{I}_{n_2})$ and given $\boldsymbol{T}$ one can simply define $\phi_{\boldsymbol{T}}$ by means of (3.3). In order to further motivate this additional and seemingly unhandy representation of matrices, we give a short example below.

Example 3.1 (Fast Fourier Transform)**.** A prominent and certainly one of the most important examples is the DFT, which transforms periodic discrete signals into the respective frequency domain. As such it has a plethora of applications in spectral analysis, radar, array processing and beyond. Given the canonical standard basis in $\mathbb{C}^n$, the corresponding matrix elements are expressed as

$$\boldsymbol{F}_n = \left[f_{i,j}\right]_{i,j=1}^{n} = \left[\exp\left(\frac{-\jmath 2\pi}{n} \cdot i \cdot j\right)\right]_{i,j=1}^{n}. \tag{3.4}$$

As we can see, the matrix $\boldsymbol{F}_n \in \mathbb{C}^{n \times n}$ is highly structured, essentially only needing the integer value $n \in \mathbb{N}$ in order to define it completely. Note that sometimes, we drop the index n if it is clear from context. As such the DFT matrix has very few, in fact just one, degree(s) of freedom, since the size n of the involved vector space already defines its elements.

It is well known that one should in fact not use (3.3) in order to multiply the matrix in (3.4) to a vector. Instead one should make use of the Fast Fourier Transform (FFT) algorithm to calculate $\boldsymbol{y} = \boldsymbol{F} \cdot \boldsymbol{x}$ with runtime complexity $\mathcal{O}(n \log n)$ [68]. This runtime complexity is one of the reasons why modern signal-processing is possible and practically feasible. It is due to this that the DFT is one of the most fundamental transforms, which sparked the need for many different efforts to deduce efficient implementations [69], [70], [71] in terms of memory, speed and accuracy.

Summarizing, we have $\phi_{\boldsymbol{F}}(\boldsymbol{x}) = \texttt{fft}(\boldsymbol{x})$ and $\beta_{\boldsymbol{F}}(\boldsymbol{x}) = \texttt{ifft}(\boldsymbol{x})$ as the two matrix-free representations of $\boldsymbol{F}$. ∎

Example 3.1 illuminates that as soon as one can exploit *structure* in the involved linear mapping, one can indeed improve upon the dense representation in terms of runtime and required memory.

When it comes to numerical algorithms for linear algebra, there are methods to solve systems of linear equations using matrix-free representations like the method of conjugate gradients [72] or there are efficient methods to calculate eigenvalues and eigenvectors using deflation techniques as outlined in [73]. In optimization, one can also make use of matrix-free representations when describing the set of feasible points [74]. Nowadays, the field of deep learning is also exploiting structured neural net components, where convolutional layers [75] are one prominent example.

In the following sections, we outline how this structural knowledge about the linear transform the can be exploited in CS when implementing the SSR algorithms using matrix-free representations.

3.2 Application to Compressed Sensing

In this chapter, we want to make use of the previously defined representation for linear mappings to render the algorithms presented in Section 2.3 more efficient in terms of memory consumption and runtime. This means that the algorithms are not only derived specifically for sparse reconstruction, but they are also able to account for the specific linear structure of the CS scenario one is dealing with. Hence, we can expect that these modifications render OMP and FISTA and possible derivatives well performing algorithms.

3.2.1 Matrix-free Orthogonal Matching Pursuit

A first and simple approach for SSR is presented in Algorithm 2.1 and we wish to reformulate it such that the computationally most intensive steps are represented by matrix-vector products. In the previously given naïve formulation, there are two seemingly computationally intensive parts. The calculation of the correlation values c_i in Line 6 and the calculation of the values of $\boldsymbol{x}$ on the estimated support set $\mathcal{S}_k$ by means of the pseudo-inverse matrix of $\boldsymbol{\Psi}_{\mathcal{S}_k}$ in Line 8. We wish to address both of these steps and find more refined computation steps to replace them.

Fast Correlation

Assume we are in step k of OMP. In order to find the next index, where the sparse $\boldsymbol{x}$ – we are in search of – has a non-zero entry, we calculate the correlation values by means of

$$c_i = \frac{|\langle \boldsymbol{\psi}_i, \boldsymbol{r}_k \rangle_2|}{\|\boldsymbol{\psi}_i\|_2} \quad \text{for} \quad i = 1, \ldots, M.$$

when resorting to matrix-vector products this can be rephrased as

$$\boldsymbol{c} = \left| \mathbf{diag}([\|\boldsymbol{\psi}_1\|_2^{-1}, \ldots, \|\boldsymbol{\psi}_N\|_2^{-1}]) \cdot \boldsymbol{\Psi}^{\mathrm{H}} \cdot \boldsymbol{r}_k \right|,$$

where $\boldsymbol{r}_k$ is the residual vector in step k and the absolute value is taken entry-wise. Additionally, we can decompose the action of $\boldsymbol{\Psi}^{\mathrm{H}}$ further and we get

$$\boldsymbol{c} = \left| \mathbf{diag}([\|\boldsymbol{\psi}_1\|_2^{-1}, \ldots, \|\boldsymbol{\psi}_N\|_2^{-1}]) \cdot \boldsymbol{A}^{\mathrm{H}} \cdot \boldsymbol{\Phi}^{\mathrm{H}} \cdot \boldsymbol{r}_k \right|,$$

which is an expression that retains the fact that $\boldsymbol{\Psi}$ is a product of two matrices. If we define

$$\boldsymbol{D}_{\boldsymbol{\Psi}} = \mathbf{diag}([\|\boldsymbol{\psi}_1\|_2^{-1}, \ldots, \|\boldsymbol{\psi}_N\|_2^{-1}]),$$

we can make use of the matrix-free representations introduced in Section 3.1 such that we finally end up with

$$\boldsymbol{c} = \left| \phi_{\boldsymbol{D}_{\boldsymbol{\Psi}}}(\beta_{\boldsymbol{A}}(\beta_{\boldsymbol{\Phi}}(\boldsymbol{r}_k))) \right| \tag{3.5}$$

as a replacement for the calculation of the c_i in Line 6 in Algorithm 2.1.

Fast Pseudo-Inverse

Next, we continue with the replacement of the pseudo-inverse, which is used to update all non-zero entries of $\boldsymbol{x}$ by means of

$$\hat{\boldsymbol{x}}_{\mathcal{S}_{k+1}} = \left(\boldsymbol{\Psi}_{\bullet,\mathcal{S}}\right)^{\dagger} \cdot \boldsymbol{z}.$$

What we have not been considering so far is the fact that the matrix $\boldsymbol{\Psi}_{\bullet,\mathcal{S}}$ is only growing by one single column in each iteration. Furthermore, we have already seen that $\boldsymbol{\Psi}_{\bullet,\mathcal{S}_k} \cdot \hat{\boldsymbol{x}}_k$ is the orthogonal projection onto the subspace spanned by $\boldsymbol{\Psi}_{\bullet,\mathcal{S}_k}$. Suppose we define the matrix

$$\boldsymbol{B}_k = \boldsymbol{\Psi}_{\bullet,\mathcal{S}_k}^{\dagger}.$$

As we will show, it is indeed possible to iteratively – and hence cheaply – update $\boldsymbol{B}_k$. As a starting point we have in step $k = 1$ that

$$\boldsymbol{B}_1 = \frac{\boldsymbol{\psi}_j^{\mathrm{H}}}{\left\|\boldsymbol{\psi}_j\right\|_2^2} \in \mathbb{C}^{1\times m},$$

if $j = \operatorname{argmax}_i c_i$ was selected by the correlation in the first step of the algorithm. Now suppose we are in step $k+1$ and have already computed $\boldsymbol{B}_k$. We wish to add a non-zero at index j and hence define

$$\boldsymbol{u}_k = \left(\boldsymbol{I}_k - \boldsymbol{\Psi}_{\bullet,\mathcal{S}_k} \cdot \boldsymbol{B}_k\right) \cdot \boldsymbol{\psi}_j.$$

Then we can find the updated pseudo inverse by means of

$$\boldsymbol{B}_{k+1} = \begin{bmatrix} \boldsymbol{B}_k \left(\boldsymbol{I}_m - \frac{\boldsymbol{\psi}_j \cdot \boldsymbol{u}_k^{\mathrm{H}}}{\|\boldsymbol{u}_k\|_2^2}\right) \\ \frac{\boldsymbol{u}_k^{\mathrm{H}}}{\|\boldsymbol{u}_k\|_2^2} \in \mathbb{C}^{1\times m} \end{bmatrix},$$

which is based on the idea that we identify the parts in $\boldsymbol{\psi}_j$ which are orthogonal to the sub-space spanned by $\boldsymbol{\Psi}_{\bullet,\mathcal{S}_k}$. The geometric interpretation here is to remove the orthogonal complement of the subspace spanned by $\boldsymbol{u}_k$ that is already contained in $\boldsymbol{B}_k$, which constitutes the calculation of the first k rows of $\boldsymbol{B}_{k+1}$ based on the previous $\boldsymbol{B}_k$. This is outlined in Figure 3.1.

As a summary for the above considerations, Algorithm 3.1 presents a reformulation of Algorithm 2.1, which operates matrix-free and without the need to explicitly calculate the pseudo-inverse in each iteration.

Finally, we want to analyze the computational complexity of Algorithm 3.1 in order to get an understanding how a practical implementation scales in terms of problem size. Clearly, line 7 is hard to quantify,

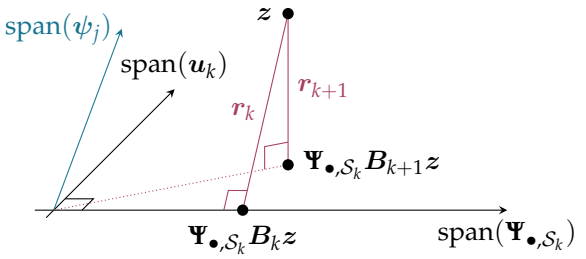


Figure 3.1 *Pseudo Inverse $\boldsymbol{B}_k$ can be found iteratively.* – Iteration process and geometric relationships of the quantities involved in calculating $\boldsymbol{B}_k$. ■

since it depends on the complexity of $\beta_{\boldsymbol{\Psi}} = \beta_{\boldsymbol{A}} \circ \beta_{\boldsymbol{\Phi}}$. If we assume that this calculation scales as $T : \mathbb{N}^3 \to \mathbb{R}$, where $(m, M, N) \mapsto T(m, M, N)$ such that $\beta \in \mathcal{O}(T(m, M, N))$ and we store $\boldsymbol{\Psi}_{\bullet,\mathcal{S}_k}$ in a dedicated variable, we can conclude that the runtime of step k is of the order $\mathcal{O}(T(m, M, N) + km + m^2)$, where the second summand stems from Line 10 and the last summand from the update of $\boldsymbol{B}_{k+1}$ in Line 11. Which means we finally have a runtime of $\mathcal{O}(S_{\max}T(m, M, N) + S_{\max}^2 m + S_{\max}m^2)$. Usually, when the system requirements demand to implement this matrix-free scheme, the cost for $\beta_{\boldsymbol{\Psi}}$ is dominating this expression and then the runtime scales proportional to the number of steps $S_{\max}$.

In Figure 3.2 we carry out a numerical investigation, where we compare the performance of Algorithm 3.1, a dense version of Algorithm 3.1, which only makes use of the more efficient pseudo-inverse and a naïve dense representation as given in Algorithm 2.1. We use a ciruclant matrix $\boldsymbol{\Psi} = \boldsymbol{\Gamma}(\boldsymbol{c})$, see Section 3.3.2.1, for $\boldsymbol{c} \in \mathbb{C}^M$ and we set $S = \lfloor 0.1 \cdot M \rfloor$ as the sparsity level. Each algorithm instance is hence run for $S_{\max} = S$ steps and we average the runtime over 25 runs. As such, we only simulate a sparse recovery setup without any compression. The runtime in Figure 3.2 is the total runtime of each algorithm divided by $S_{\max}$ in order to only account for the algorithms performance depending on the involved transform sizes.

As we can see, the purely matrix-free implementation easily outperforms the other two realizations of OMP, which also happens already for moderately sized problems. And even if we use the less efficient dense matrix-vector product in Line 7 the improved update of the pseudo-inverse matrix in Line 10 allows for another substantial improvement compared to the implementation in Algorithm 2.1.

Data: Observed measurement vector $\boldsymbol{z}$, Maximum sparsity order $S_{\max} \in \mathbb{N}$;

[1] $\mathcal{S}_0 = \emptyset$;
[2] $k = 0$;
[3] $\boldsymbol{r}_0 = \boldsymbol{z}$;
[4] $\boldsymbol{B}_0 = \boldsymbol{0} \in \mathbb{C}^{0 \times m}$;
[5] $\hat{\boldsymbol{x}}_0 = \boldsymbol{0}$;
[6] **while** $k < S_{\max}$ **do**
[7] $\quad \boldsymbol{c} = |\phi_{\boldsymbol{D}_\Psi}(\beta_{\boldsymbol{A}}(\beta_{\boldsymbol{\Phi}}(\boldsymbol{r}_k)))|$;
[8] $\quad j = \operatorname{argmax}_i c_i$;
[9] $\quad \mathcal{S}_{k+1} = \mathcal{S}_k \cup \{j\}$;
[10] $\quad \boldsymbol{u}_{k+1} = (\boldsymbol{I}_m - \boldsymbol{\Psi}_{\bullet,\mathcal{S}_k} \cdot \boldsymbol{B}_k) \cdot \psi_j$;
[11] $\quad \boldsymbol{B}_{k+1} = \begin{bmatrix} \boldsymbol{B}_k \left(\boldsymbol{I}_m - \frac{\psi_j \cdot \boldsymbol{u}_k^{\mathrm{H}}}{\|\boldsymbol{u}_k\|_2^2} \right) \\ \frac{\boldsymbol{u}_k^{\mathrm{H}}}{\|\boldsymbol{u}_k\|_2^2} \in \mathbb{C}^{1 \times m} \end{bmatrix}$;
[12] $\quad \hat{\boldsymbol{x}}_{k+1} = \boldsymbol{B}_{k+1} \cdot \boldsymbol{z}$ on the set $\mathcal{S}_{k+1}$;
[13] $\quad \boldsymbol{r}_{k+1} = \boldsymbol{z} - \boldsymbol{\Psi}_{\bullet,\mathcal{S}_k} \cdot \hat{\boldsymbol{x}}_{\mathcal{S}_{k+1}}$;
[14] $\quad k = k + 1$;
[15] **end**
[16] return $\hat{\boldsymbol{x}}_{S_{\max}}$;

Algorithm 3.1: *The OMP algorithm has a fast matrix-free variant.* – Matrix-free version of Algorithm 2.1 using rank-1 updates for the pseudo-inverse matrix and matrix-free correlation.

3.2.2 Matrix-free Fast Iterative Shrinkage-Thresholding Algorithm

As another popular choice of algorithm we presented a simple variant of FISTA given by Algorithm 2.2 in Section 2.3.2. Based on what we have developed for OMP, it is straightforward to apply the same matrix-free approach here. We simply exchange the matrix-vector products involving $\boldsymbol{\Psi}$ with the respective matrix-free functions that carry out the linear transforms without resorting to the dense representation of $\boldsymbol{\Psi}$.

As we will see, this slight modification renders Algorithm 3.2 a viable approach in ultrasonic imaging as outlined in Section 4.1. This is due to the fact that a single step of the algorithm is in $\mathcal{O}(T(m, M, N))$. Consequently, the whole iteration is in $\mathcal{O}(K_{\max} T(m, M, N))$, which similarly to OMP renders the computational effort proportional to the number of iterations carried out.

Finally, we would like to note that the stepsize α_k used in Line 3 must be chosen in adaption to $\boldsymbol{\Psi}$ in order to achieve rapid convergence and to

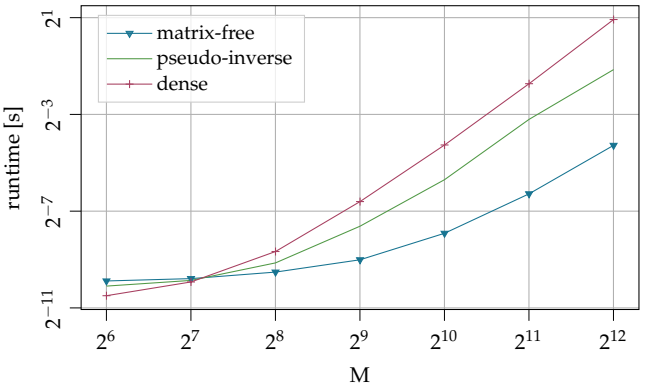


Figure 3.2 *The matrix-free representation substantially decreases the runtime per iteration step of OMP.* – Comparison of the different approaches for OMP in terms of runtime per iteration: Algorithm 3.1 (blue), Algorithm 3.1 where a dense matrix-vector product is the used instead of β (green), Algorithm 2.1 (red). ■

avoid divergence of the iteration scheme. Usually one chooses α_k depending on the largest singular value of $\mathbf{\Psi}$. Since in the matrix-free setting we do not have access to direct methods to calculate a singular value decomposition directly, one has to resort to methods for estimating and approximating this quantity. In Section 4.1 we will present various ways to address this issue for the specific scenarios considered there. As we will see, the matrix-free representation facilitates this process tremendously.

In Figure 3.3 we carry out similar simulations as in Figure 3.2, but instead we compare two types of structures here, which are Circulant matrices and Hadamard (see Section 3.3.3 [76]) matrices. Also, we only compare the execution times for the dense and the matrix-free cases, since there are only these two variants of Algorithm 3.2. We average the runtimes of the algorithms over 64 trials. As we can see again, the matrix-free version have a substantially better scaling behavior in terms of dependency on the problem size. Obviously, this improvement in performance only stems from the usage of ϕ and β, as given in Algorithm 3.2. Interestingly, the dense representation of the Hadamard matrices does outperform the dense representation of the Circulant matrices. This is due to the fact that the software framework, depicted in Section 3.4, allows to handle data types very efficiently and we can retain the information that a Hadamard matrix can be represented using integers only, which allows us to take advantage of this when using a standard matrix-vector

Data: Observed measurement vector z, Maximum number of steps $K_{\max} \in \mathbb{N}$;

[1] $k = 0$, $t_0 = 1$, $v_0 = u_0 = \mathbf{0}_M \in \mathbb{C}^M$;

[2] **while** $1 \leqslant k \leqslant K_{\max}$ **do**

[3] $\quad u_{k+1} = \tau_{\lambda \cdot \alpha_k}\left(v_k - \alpha_k \beta_\Psi(\phi_\Psi(v_k) - z)\right)$;

[4] $\quad t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}$;

[5] $\quad v_{k+1} = u_{k+1} + \left(\frac{t_k - 1}{t_{k+1}}\right)(u_{k+1} - u_k)$;

[6] **end**

Algorithm 3.2: *FISTA can easily be adapted to a matrix-free iteration.* – Matrix-free Fast Iterative Shrinkage-Thresholding Algorithm, where we exchange matrix-vector products with the matrix-free counterparts.

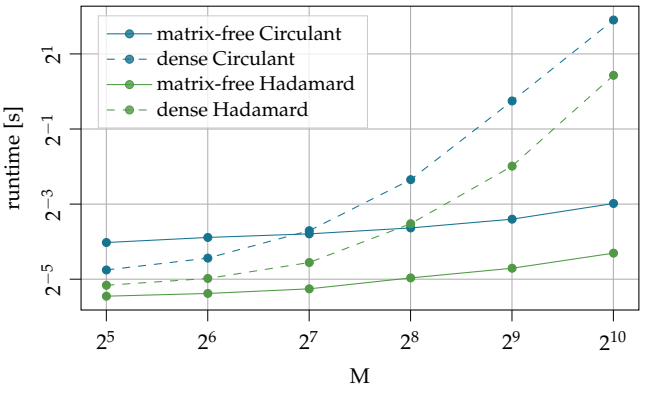


Figure 3.3 *The performance of FISTA is mainly influenced by the performance of ϕ and β.* – Comparison of the runtimes of Algorithms 2.2 and 3.2 for the esitmation of $x \in \mathbb{C}^M$ averaged over 64 trials. ■

product.

To conclude we wish to note that similar comparisons can be made in terms of memory consumption and the conclusions would also match those derived for the runtime.

3.3 Deriving Matrix-free Representations

Since in the previous Section 3.2 we depict how to use matrix-free representations in Compressed Sensing we now present methods how to derive these matrix-free representations for various types of matrices that we use in later chapters. We are aware of the fact that the explanations are more excessive than needed, since especially fast convolution algorithms are well known for many years. However, the more detailed derivations help us when motivating certain design decisions for the developed software package which is described in Section 3.4.

3.3.1 Products of Matrices

In order to gain some intuition with respect to the concept of matrix-free representations, we start with some simple examples involving two kinds of matrix-matrix-products.

Regular Matrix-Matrix-Product

Assume we have two matrices $\boldsymbol{A} \in \mathbb{C}^{n \times k}$ and $\boldsymbol{B} \in \mathbb{C}^{k \times m}$. In virtue of the previous chapter we also have access to $\phi_{\boldsymbol{A}}, \phi_{\boldsymbol{B}}$ and $\beta_{\boldsymbol{A}}, \beta_{\boldsymbol{B}}$ respectively. Based on these, we want to derive $\phi_{\boldsymbol{A} \cdot \boldsymbol{B}}$. Since the linear transform associated with $\boldsymbol{A} \cdot \boldsymbol{B}$ is simply the concatenation (in the function sense) of the linear transform associated with $\boldsymbol{B}$ and $\boldsymbol{A}$. So, we have

$$\phi_{\boldsymbol{A} \cdot \boldsymbol{B}} = \phi_{\boldsymbol{A}} \circ \phi_{\boldsymbol{B}}, \tag{3.6}$$

which means that $\phi_{\boldsymbol{A} \cdot \boldsymbol{B}} : \mathbb{C}^m \to \mathbb{C}^n$ with $\boldsymbol{x} \mapsto \phi_{\boldsymbol{A}}(\phi_{\boldsymbol{B}}(\boldsymbol{x}))$. Due to the fact that $(\boldsymbol{A} \cdot \boldsymbol{B})^{\mathrm{H}} = \boldsymbol{B}^{\mathrm{H}} \cdot \boldsymbol{A}^{\mathrm{H}}$ we can also reason that

$$\beta_{\boldsymbol{A} \cdot \boldsymbol{B}} = \beta_{\boldsymbol{B}} \circ \beta_{\boldsymbol{A}}.$$

Note that this product can be extended to more than two factors by making use of the fact that the regular matrix-matrix-product is associative.

At first it might seem inefficient to conserve the information that $\boldsymbol{C} = \boldsymbol{A} \cdot \boldsymbol{B}$ is a product of two matrices. However, consider the case when

$\boldsymbol{B}$ is a DFT matrix and $\boldsymbol{A}$ is a diagonal matrix. Then explicitly forming the product of $\boldsymbol{A}$ and $\boldsymbol{B}$ yields a matrix that generally has no efficient matrix-free representation, however if we retain the information, we can still exploit the individual matrices' fast transforms. By doing so one preserves the structure of the linear mapping.

Kronecker Matrix-Matrix-Product

Assume we have two matrices $\boldsymbol{A} \in \mathbb{C}^{n_1 \times m_1}$ and $\boldsymbol{B} \in \mathbb{C}^{n_2 \times m_2}$. Then the Kronecker Product of $\boldsymbol{A}$ and $\boldsymbol{B}$ is defined via

$$\boldsymbol{A} \otimes \boldsymbol{B} = \begin{bmatrix} a_{1,1} \cdot \boldsymbol{B} & \dots & a_{1,m_1} \cdot \boldsymbol{B} \\ \vdots & \ddots & \vdots \\ a_{n_1,1} \cdot \boldsymbol{B} & \dots & a_{n_1,m_1} \cdot \boldsymbol{B} \end{bmatrix} \in \mathbb{C}^{n_1 n_2 \times n_1 n_2}.$$

This implies that the dense representation of $\boldsymbol{C}$ uses $\mathcal{O}(n_1 n_2 m_1 m_2)$ memory and hence a single matrix-vector product is in $\mathcal{O}(n_1^2 n_2^2 m_1^2 m_2^2)$ operations. However, similarly to the derivation for the regular matrix-matrix-product we can retain the fact that $\boldsymbol{C}$ is a Kronecker-Product of $\boldsymbol{A}$ and $\boldsymbol{B}$. This implies that the memory consumption is only in the order needed to store $\boldsymbol{A}$ and $\boldsymbol{B}$.

As outlined in [77], one can rewrite the Kronecker-product as

$$\boldsymbol{C} = (\boldsymbol{A} \otimes \boldsymbol{I}_{n_2}) \cdot (\boldsymbol{I}_{n_1} \otimes \boldsymbol{B}), \tag{3.7}$$

where the authors make the assumption that $n_1 = m_1$ and $n_2 = m_2$, so $\boldsymbol{A}$ and $\boldsymbol{B}$ are square matrices. Then, they show how to efficiently apply a matrix with the structure in (3.7) to a vector, which yields an efficient algorithm for $\phi_{\boldsymbol{A} \otimes \boldsymbol{B}}$ as well as for $\beta_{\boldsymbol{A} \otimes \boldsymbol{B}}$.

However, this squaredness assumption can be dropped if we imagine $\boldsymbol{C}$ as a linear transform on a 2D input array $\boldsymbol{\mathcal{X}} \in \mathbb{C}^{m_1 \times m_2}$. Note that here we do not consider $\boldsymbol{\mathcal{X}}$ as a matrix encoding a linear transform, but rather a 2D array, where each dimension in turn is associated with a vector space – in this case $\mathbb{C}^{m_1}$ and $\mathbb{C}^{m_2}$. This means that here the term "dimension" denotes the number of independent indices used to account for the structure of the input $\boldsymbol{\mathcal{X}}$.

Hence, the fact that $\boldsymbol{C}$ is a Kronecker-product means that $\boldsymbol{A}$ is applied along one dimension of $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{B}$ along the other one. This means we can simply write for some $\boldsymbol{x} \in \mathbb{C}^{m_1 \cdot m_2}$ and its reshaping $bm\mathcal{X}(\boldsymbol{x})) \in \mathbb{C}^{m_1 \dot{m}_2}$ that

$$\boldsymbol{x} \mapsto \phi_{\boldsymbol{C}}(\boldsymbol{x}) = \mathrm{vec}((\boldsymbol{B} \cdot (\boldsymbol{A} \cdot \boldsymbol{\mathcal{X}}(\boldsymbol{x}))^{\mathrm{T}})^{\mathrm{T}}) = \mathrm{vec}((\phi_{\boldsymbol{B}}(\phi_{\boldsymbol{A}}(\boldsymbol{\mathcal{X}}(\boldsymbol{x})))^{\mathrm{T}})^{\mathrm{T}}).$$

This means, we apply $\phi_{\boldsymbol{A}}$ along the columns of $\boldsymbol{\mathcal{X}}$ and $\phi_{\boldsymbol{B}}$ along the rows of $\boldsymbol{\mathcal{X}}$. Naturally, this can be generalized to more Kronecker-factors that make up $\boldsymbol{C}$. As such we have provided a slight but useful generalization to the results in [77], since we have shown that the assumption of square $\boldsymbol{A}$ and $\boldsymbol{B}$ is not necessary. Moreover, we have shown that retaining the information that $\boldsymbol{C}$ is a Kronecker-product, reduces its memory and runtime consumption to the complexity of $\phi_{\boldsymbol{A}}$ and $\phi_{\boldsymbol{B}}$ respectively. Also note that this can easily be generalized to higher dimensions and also for the definition of $\beta_{\boldsymbol{c}}$.

3.3.2 Convolution Matrices

In the application presented in Section 4.1 we make use of translational invariance present in 3D ultrasound data aligned in a synthetic aperture. This allows us to derive efficient matrix-free reconstruction schemes and it is why the following section is dedicated to efficient forward and backward projections of multilevel circulant and Toeplitz matrices, where the first two Sections 3.3.2.1 and 3.3.2.2 revise the procedure to derive efficient implementations for circulant and Toeplitz matrices acting on vectors representing 1D data.

3.3.2.1 Circulant Matrices

We define the mapping $\boldsymbol{\Gamma} : \mathbb{C}^n \to \mathbb{C}^{n\times n}$ for given $n \in \mathbb{N}$ and $\boldsymbol{c} \in \mathbb{C}^n$ via

$$\boldsymbol{c} \mapsto \boldsymbol{\Gamma}(\boldsymbol{c}) = \begin{bmatrix} c_1 & c_n & \dots & c_2 \\ c_2 & c_1 & \ddots & c_3 \\ \vdots & & \ddots & c_n \\ c_n & c_{n-1} & \dots & c_1 \end{bmatrix}.$$

Now in order to derive $\phi_{\boldsymbol{\Gamma}(\boldsymbol{c})}$ and $\beta_{\boldsymbol{\Gamma}(\boldsymbol{c})}$ we make use of the well known fact that for any $\boldsymbol{c} \in \mathbb{C}^n$ it holds that

$$\boldsymbol{\Gamma}(\boldsymbol{c}) = \frac{1}{n}\boldsymbol{F}_n^{\mathrm{H}} \cdot \mathbf{diag}(\boldsymbol{F}_n \boldsymbol{c}) \cdot \boldsymbol{F}_n, \tag{3.8}$$

where $\boldsymbol{F}_n$ is the DFT matrix as introduced in Example 3.1. This means we can derive for the forward transform $\phi_{\boldsymbol{\Gamma}(c)}$ via

$$\phi_{\boldsymbol{\Gamma}(\boldsymbol{c})}(\boldsymbol{x}) = \texttt{ifft}(\hat{\boldsymbol{c}} \odot \texttt{fft}(\boldsymbol{x})), \tag{3.9}$$

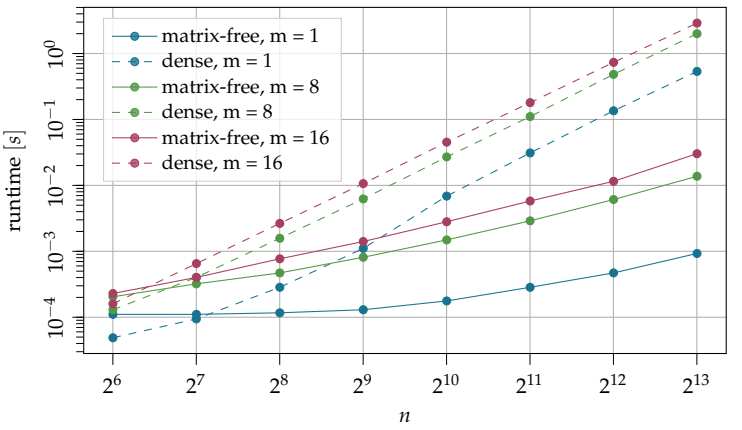


Figure 3.4 *Circulant Matrices have a fast matrix-free representation.* – Comparison of the runtime between the dense and matrix-free implementation for $\boldsymbol{\Gamma}(\boldsymbol{c}) \cdot \boldsymbol{x}$ for $\boldsymbol{x} \in \mathbb{C}^{n \times m}$. ■

where we can also precompute and store $\hat{\boldsymbol{c}} = \texttt{fft}(\boldsymbol{c})$ and $\odot$ denotes the elementwise product, i.e. the Hadamard product, of two vectors. This also shows that it is possible to represent $\boldsymbol{\Gamma}(\boldsymbol{c})$ memory-efficiently as $\boldsymbol{F}_n \boldsymbol{c}$, reducing the memory complexity to $\mathcal{O}(n)$, compared to $\mathcal{O}(n^2)$ in the case of its dense representation. This is illustrated in the runtime comparison results in Figure 3.4[2].

However, the key idea behind the FFT algorithm is also its greatest caveat, because it only achieves reasonable performance if the transform size n factors into many small prime factors. In the extreme case where n is prime on the other hand, the FFT is not faster than a standard matrix-vector multiplication. To circumvent this shortcoming one has to employ suitable zero-padding as outlined in Appendix B.1.

[2]Results are obtained on Intel(R) Core(TM) i7-8565U CPU with 1.80GHz, 32GiB SODIMM DDR4 Synchronous 2400 MHz Memory and 256KiB/1MiB/8MiB L1/L2/L3 cache.

3.3.2.2 Toeplitz Matrices

The concepts of Section 3.3.2.1 can be extended to the more general case, where the matrix is not circulant, but has Toeplitz structure instead. For $n \in \mathbb{N}$ and $\boldsymbol{t} \in \mathbb{C}^{2n-1}$, we define a mapping $\boldsymbol{\Theta} : \mathbb{C}^{2n-1} \to \mathbb{C}^{n \times n}$ via

$$\boldsymbol{t} \mapsto \boldsymbol{\Theta}(\boldsymbol{t}) = \begin{bmatrix} t_1 & t_{2n-1} & \dots & t_{n+1} \\ t_2 & t_1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & t_{2n-1} \\ t_n & t_{n-1} & \dots & t_1 \end{bmatrix}.$$

The idea behind the derivation of $\phi_{\boldsymbol{\Theta}(\boldsymbol{t})}$ is to embed $\boldsymbol{\Theta}(\boldsymbol{t})$ into a suitable circulant matrix. To derive the defining vector of this circulant matrix, we conjure a mapping $\boldsymbol{\vartheta}_k : \mathbb{C}^{2n-1} \to \mathbb{C}^{2n-1+k}$, as

$$\boldsymbol{\vartheta}_k(\boldsymbol{t}) = [t_1, t_2, \dots, t_n, \mathbf{0}_k, t_{n+1}, \dots, t_{2n-1}]^{\mathrm{T}}.$$

Note that $\boldsymbol{z}$ is just an auxiliary vector, which can be discarded after the computation. With this definition at hand it is clear that for $\boldsymbol{t} \in \mathbb{C}^{2n-1}$, $\boldsymbol{x} \in \mathbb{C}^n$ and $\boldsymbol{z} \in \mathbb{C}^{k+n-1}$ the relation

$$\begin{bmatrix} \boldsymbol{\Theta}(\boldsymbol{t}) \cdot \boldsymbol{x} \\ \boldsymbol{z} \end{bmatrix} = \boldsymbol{\Gamma}(\boldsymbol{\vartheta}_k(\boldsymbol{t})) \cdot \begin{bmatrix} \boldsymbol{x} \\ \mathbf{0}_{k+n-1} \end{bmatrix} \tag{3.10}$$

holds for any $k \geqslant 0$. This implies that Toeplitz matrices also have an efficient forward and backward transform by means of the algorithm provided for circulant matrices in (3.9). This is illustrated in the runtime comparison results in Figure 3.5. For the zero-padding step in (3.10), we need to find a suitable $k \geqslant 0$, such that the FFT calculations in (3.10) (or (3.9)) are most efficient. To this end we again refer to Appendix B.1, where this zero-padding is discussed in more detail.

3.3.2.3 Multilevel Circulant Matrices

As a next generalization, we define so called multilevel circulant matrices, which are not circulant by themselves, but consist of multiple nested levels of circulant structures as illustrated in Figure 3.6. They can be considered as matrices that apply a convolution to higher dimensional data structures represented by vectors. Here, the convolution separates along the data, which first allows us to treat the different convolutions separately and ultimately allows for a recursive formulation of the matrix-structure.

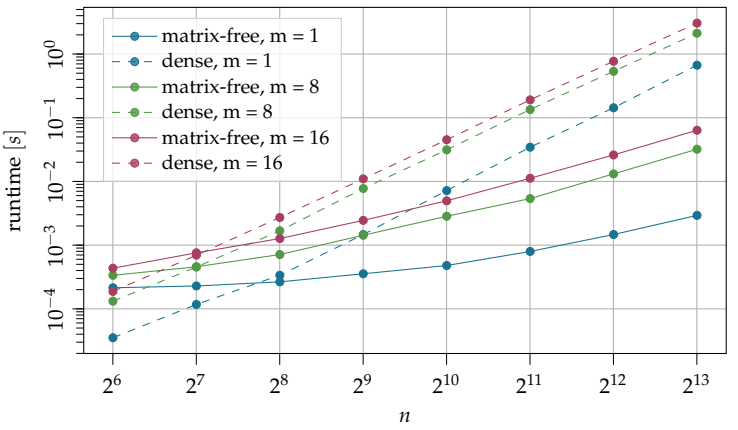


Figure 3.5 *Toeplitz Matrices have a fast matrix-free representation.* – Comparison of the runtime between the dense and matrix-free implementation for $\mathbf{\Theta}(\boldsymbol{t}) \cdot \boldsymbol{x}$ for $\boldsymbol{x} \in \mathbb{C}^{n \times m}$. ∎

To formulate the algorithm precisely, let $d \geqslant 1$ represent the number of nested levels, $\boldsymbol{n} = [n_1, \ldots, n_d] \in \mathbb{N}^d$ represent the respective levels' sizes, $\boldsymbol{n}_{1-} = [n_1, \ldots, n_{d-1}] \in \mathbb{N}^{d-1}$ and $\boldsymbol{n}_{-1} = [n_2, \ldots, n_d] \in \mathbb{N}^{d-1}$. Additionally, we are given a d-dimensional complex sequence $\boldsymbol{c} = [c_{\boldsymbol{k}}]$ for the multi index $\boldsymbol{k} \in \mathbb{N}^d$. This means that we have a d-dimensional array $\boldsymbol{c} \in \mathbb{C}^{n_1 \times \cdots \times n_d}$ that is supposed to define the d-level circulant matrix $\boldsymbol{C}_{\boldsymbol{n},d}$ recursively via

$$\boldsymbol{C}_{\boldsymbol{n},d} = \begin{bmatrix} \boldsymbol{C}^1_{[\boldsymbol{n}_{-1}],d-1} & \boldsymbol{C}^{n_1}_{[\boldsymbol{n}_{-1}],d-1} & \cdots & \boldsymbol{C}^2_{[\boldsymbol{n}_{-1}],d-1} \\ \boldsymbol{C}^2_{[\boldsymbol{n}_{-1}],d-1} & \boldsymbol{C}^1_{[\boldsymbol{n}_{-1}],d-1} & \cdots & \boldsymbol{C}^3_{[\boldsymbol{n}_{-1}],d-1} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{C}^{n_1}_{[\boldsymbol{n}_{-1}],d-1} & \boldsymbol{C}^{n_1-1}_{[\boldsymbol{n}_{-1}],d-1} & \cdots & \boldsymbol{C}^1_{[\boldsymbol{n}_{-1}],d-1} \end{bmatrix}.$$

In order to illustrate this recursion for two levels we set $\boldsymbol{n} = [2,2]$ and $\boldsymbol{c} \in \mathbb{C}^{2\times 2}$, from which we get

$$\boldsymbol{C}_{[2,2],2} = \begin{bmatrix} \boldsymbol{C}^1_{[2],1} & \boldsymbol{C}^2_{[2],1} \\ \boldsymbol{C}^2_{[2],1} & \boldsymbol{C}^1_{[2],1} \end{bmatrix} = \begin{bmatrix} c^{1,1} & c^{1,2} & c^{2,1} & c^{2,2} \\ c^{1,2} & c^{1,1} & c^{2,2} & c^{2,1} \\ c^{2,1} & c^{2,2} & c^{1,1} & c^{1,2} \\ c^{2,2} & c^{2,1} & c^{1,2} & c^{1,1} \end{bmatrix}.$$

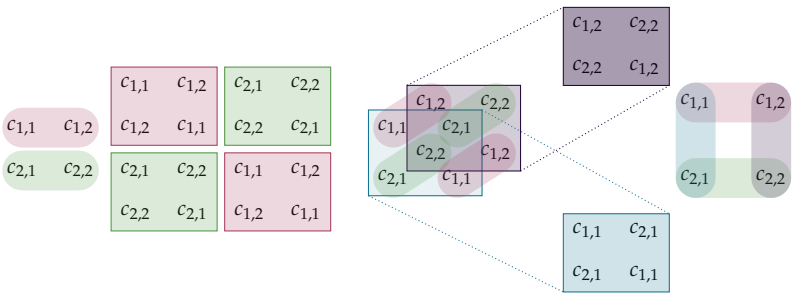


Figure 3.6 *For 2-level Circulant matrices we have two ways of being ciruclant.* – Visualization of the fact that multi-level circulant matrices have a circulant structure along *all* dimensions of the defining array $\boldsymbol{c}$. ■

It is worth noting that we stick with $\boldsymbol{c}$ containing all defining elements as the representation of a circulant matrix even in the d-level case, where $\boldsymbol{c}$ then is a d-dimensional array, as a natural extension to the circulant case for $d = 1$. To clarify how the elements in $\boldsymbol{c}$ are placed into $\boldsymbol{C}_{\boldsymbol{n},d}$, we note that

$$\boldsymbol{C}_{[n_1,\dots,n_{d-1},k],d-1} = \boldsymbol{\Gamma}\left(c_{[n_1,\dots,n_{d-1},k]}\right)$$

for all $k = 1, \dots, n_d$ and we again refer to Figure 3.6.

Fast Multiplication

In spirit of the sections before, we aim at providing efficient means of representing $\boldsymbol{C}_{\boldsymbol{n},d}$ and deriving its matrix-free forward and backward transforms $\phi_{\boldsymbol{C}_{\boldsymbol{n},d}}$ and $\beta_{\boldsymbol{C}_{\boldsymbol{n},d}}$ respectively. To this end, we exploit the multilevel structure, by means of the diagonalization of multilevel circulant matrices [78], which reads as

$$\boldsymbol{C}_{\boldsymbol{n},d} = \bigotimes_{i=1}^{d} \frac{1}{n_i} \boldsymbol{F}_{n_i}^{\mathrm{H}} \operatorname{diag}\left(\bigotimes_{i=1}^{d} \boldsymbol{F}_{n_i} \cdot \operatorname{vec} \boldsymbol{c}\right) \bigotimes_{i=1}^{d} \boldsymbol{F}_{n_i}. \tag{3.11}$$

This expression directly yields an algorithm to efficiently multiply $\boldsymbol{C}_{\boldsymbol{n},d}$ or $\boldsymbol{C}_{\boldsymbol{n},d}^{\mathrm{H}}$ to a vector, because Fourier matrices allow a fast matrix-free representation as illustrated in Example 3.1. Additionally, (3.7) based on [77] describes how to efficiently compute the forward and backward transform of a Kronecker product, which also only makes use of the forward and backward transform of the factors involved. The resulting gain in computational efficiency is depicted in Figure 3.7. Note that the maximum k such that $k^3 = 2^13$ is only around ≈ 20, so the problem size is fairly small compared to practically relevant scenarios.

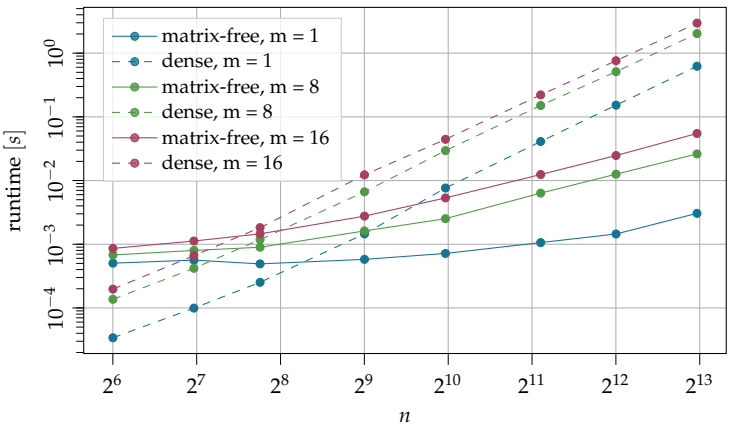


Figure 3.7 *Multilevel Circulant Matrices have a fast matrix-free representation.* – Comparison of the runtime between the dense and matrix-free implementation for $\boldsymbol{C}_{[k,k,k],3} \cdot \boldsymbol{x}$ for $\boldsymbol{x} \in \mathbb{C}^{k^3 \times m}$ with $k^3 = n$. ■

3.3.2.4 Multilevel Toeplitz Matrices

Similar to the previous section, we can also efficiently handle the more general multilevel Toeplitz case, which represent non-circular multidimensional convolutions that separate along these dimensions.

Let $d \geqslant 1$ represent the number of nested levels, $\boldsymbol{n} = [n_1, \ldots, n_d] \in \mathbb{N}^d$ represent the respective levels' sizes, $\boldsymbol{n}_{1-} = [n_1, \ldots, n_{d-1}] \in \mathbb{N}^{d-1}$ and $\boldsymbol{n}_{-1} = [n_2, \ldots, n_d] \in \mathbb{N}^{d-1}$. Additionally, we are given a d-dimensional complex sequence $\boldsymbol{t} = [t_{\boldsymbol{k}}]$ for the multi index $\boldsymbol{k} \in \mathbb{N}^d$. This means that we have a d-dimensional array $\boldsymbol{t} \in \mathbb{C}^{2n_1-1 \times \cdots \times 2n_d-1}$ which is used to define the d-level Toeplitz matrix $\boldsymbol{T}_{(\boldsymbol{n},d)}$ recursively via

$$\boldsymbol{T}_{(\boldsymbol{n},d)}(\boldsymbol{t}) = \begin{bmatrix} \boldsymbol{T}^{1}_{[\boldsymbol{n}_{-1}],d-1} & \boldsymbol{T}^{2n_1-1}_{[\boldsymbol{n}_{-1}],d-1} & \cdots & \boldsymbol{T}^{n_1+1}_{[\boldsymbol{n}_{-1}],d-1} \\ \boldsymbol{T}^{2}_{[\boldsymbol{n}_{-1}],d-1} & \boldsymbol{T}^{1}_{[\boldsymbol{n}_{-1}],d-1} & \cdots & \boldsymbol{T}^{n_1+2}_{[\boldsymbol{n}_{-1}],d-1} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{T}^{n_1}_{[\boldsymbol{n}_{-1}],d-1} & \boldsymbol{T}^{n_1-1}_{[\boldsymbol{n}_{-1}],d-1} & \cdots & \boldsymbol{T}^{1}_{[\boldsymbol{n}_{-1}],d-1} \end{bmatrix}. \tag{3.12}$$

For example, setting $\boldsymbol{n} = [2, 2]$ as in the multi-level circulant case and

$\boldsymbol{t} \in \mathbb{C}^{3\times 3}$ yields

$$\boldsymbol{T}_{[2,2],2} = \begin{bmatrix} \boldsymbol{T}^1_{[2],1} & \boldsymbol{T}^3_{[2],1} \\ \boldsymbol{T}^2_{[2],1} & \boldsymbol{T}^1_{[2],1} \end{bmatrix} = \begin{bmatrix} t^{1,1} & t^{1,3} & t^{3,1} & t^{3,3} \\ t^{1,2} & t^{1,1} & t^{3,2} & t^{3,1} \\ t^{2,1} & t^{2,3} & t^{1,1} & t^{1,3} \\ t^{2,2} & t^{2,1} & t^{1,2} & t^{1,1} \end{bmatrix}.$$

We also retain the notation $\boldsymbol{t}$ for the defining elements of $\boldsymbol{T}_{(\boldsymbol{n},d)}$ when it naturally becomes a d-dimensional array in the case $d \geqslant 2$.

As in Section 3.3.2.2 we describe how a d-level Toeplitz matrix can be embedded into a larger d-level circulant matrix such that one can use the efficient methods available for those to implement $\phi_{\boldsymbol{T}_{(\boldsymbol{n},d)}}$ and $\beta_{\boldsymbol{T}_{(\boldsymbol{n},d)}}$ respectively. To this end, we have to correctly treat the recursive nature of the definition of $\boldsymbol{T}_{(\boldsymbol{n},d)}$. We do this by considering single levels first. A given block Toeplitz matrix $\boldsymbol{T}_{n,m} \in \mathbb{C}^{nm\times nm}$ consists of matrices that are aligned as blocks in a Toeplitz-fashion and it reads as

$$\boldsymbol{T} = \begin{bmatrix} \boldsymbol{T}_1 & \boldsymbol{T}_{2n-1} & \dots & \boldsymbol{T}_{n+1} \\ \boldsymbol{T}_2 & \boldsymbol{T}_1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ \boldsymbol{T}_n & \boldsymbol{T}_{n-1} & \dots & \boldsymbol{T}_1 \end{bmatrix}.$$

So it consists of $2n-1$ arbitrary matrices $\boldsymbol{T}_i \in \mathbb{C}^{m\times m}$. Next, we define the actual embedding mapping $\mathfrak{G}_{n,m} : \mathbb{C}^{nm\times nm} \to \mathbb{C}^{(2n-1)m\times(2n-1)m}$, which extends the block Toeplitz matrix to a block circulant matrix by

$$\boldsymbol{T} \mapsto \begin{bmatrix} \boldsymbol{T}_1 & \boldsymbol{T}_{2n-1} & \dots & \boldsymbol{T}_{n+1} & \boldsymbol{T}_n & \dots & \boldsymbol{T}_2 \\ \boldsymbol{T}_2 & \boldsymbol{T}_1 & \ddots & \vdots & & & \boldsymbol{T}_3 \\ \vdots & \ddots & \ddots & \vdots & & & \\ \boldsymbol{T}_n & \boldsymbol{T}_{n-1} & \dots & \boldsymbol{T}_1 & \boldsymbol{T}_{2n-1} & & \vdots \\ \boldsymbol{T}_{n+1} & \boldsymbol{T}_n & \dots & \boldsymbol{T}_2 & \boldsymbol{T}_1 & & \vdots \\ \vdots & & & & & \ddots & \\ \boldsymbol{T}_{2n-1} & \boldsymbol{T}_{2n-2} & & \dots & & & \boldsymbol{T}_1 \end{bmatrix}.$$

As such, it works as the embedding step for a single level. This means that for given block matrices aligned in a Toeplitz-fashion, we embed those

same blocks in a block circulant matrix, such that the original blocks $\boldsymbol{T}_i$ form the original block matrix $\boldsymbol{T}$. It is worth noting that for each n it holds that $\mathfrak{G}_{n,1}(\boldsymbol{T}) = \boldsymbol{\Gamma}\left(\boldsymbol{\Theta}^{-1}(\boldsymbol{T})\right)$ for all single-level Toeplitz matrices $\boldsymbol{T}$, so we recover the original embedding of a single-level Toeplitz matrix into a circulant matrix.

To finalize the embedding of the matrix $\boldsymbol{T}_{(\boldsymbol{n},d)}$ into a multilevel circulant matrix, we iteratively apply the appropriate $\mathfrak{G}_{n_i,m_j}$ via

$$\mathfrak{T}_1(\boldsymbol{T}_{(\boldsymbol{n},d)}) = \mathfrak{G}_{n_1,K}\left(\begin{bmatrix} \mathfrak{T}_\ell(\boldsymbol{T}^1_{[\boldsymbol{m}],\ell}) & \dots & \mathfrak{T}_\ell(\boldsymbol{T}^{n_1+1}_{[\boldsymbol{m}],\ell}) \\ \mathfrak{T}_\ell(\boldsymbol{T}^2_{[\boldsymbol{m}],\ell}) & \dots & \mathfrak{T}_\ell(\boldsymbol{T}^{n_1+2}_{[\boldsymbol{m}],\ell}) \\ \vdots & \ddots & \vdots \\ \mathfrak{T}_\ell(\boldsymbol{T}^{n_1}_{[\boldsymbol{m}],\ell}) & \dots & \mathfrak{T}_\ell(\boldsymbol{T}^1_{[\boldsymbol{m}],\ell}) \end{bmatrix}\right), \tag{3.13}$$

where $K = n_2 \cdot \dots \cdot n_d$. The matrix is by design d-level circulant and can be diagonalized as in (3.11). Also, for an appropriately chosen index set S

$$\mathfrak{T}_d(\boldsymbol{T}_{(\boldsymbol{n},d)})_{S,S} = \boldsymbol{T}_{(\boldsymbol{n},d)} \tag{3.14}$$

holds. The set S subselecting the resulting matrix can be constructed iteratively by keeping track of inserted spurious columns and rows into $\mathfrak{T}_d(\boldsymbol{T}_{(\boldsymbol{n},d)})$ during the above embedding procedure, compared to the original $\boldsymbol{T}_{(\boldsymbol{n},d)}$. The resulting gain in computational efficiency is depicted in Figure 3.8. Again note that the problem size is not large in terms of k and already the matrix-free implementation can deliver several orders of magnitude better performance.

3.3.2.5 Block Multilevel Toeplitz Matrices

Here, we treat a special linear transform structure which turns out to be of great importance in Section 4.1, since it allows to formulate a matrix-free representation for the involved dictionary. Also, in Section 4.1.4.2 we present a more direct approach to this matrix structure, which also allows the implementation on a GPU.

Consider a matrix $\boldsymbol{H}$ which has a block structure, where each block $\boldsymbol{H}_{i,j}$ for $i,j = 0,\dots,M-1$ is a d-level Toeplitz matrix, which is why in general one could call the matrix $\boldsymbol{H}$ block multilevel Toeplitz. Here, we consider the special case $d = 1$. Hence, we can collect the unique defining elements of $\boldsymbol{H}$ in $\boldsymbol{h} \in \mathbb{R}^{M\times M\times 2N_1-1\times 2N_1-1}$ and then set

$$\boldsymbol{H}_{i,j} = \boldsymbol{T}_{([N_1,N_1],2)}(\boldsymbol{h}_{i,j}).$$

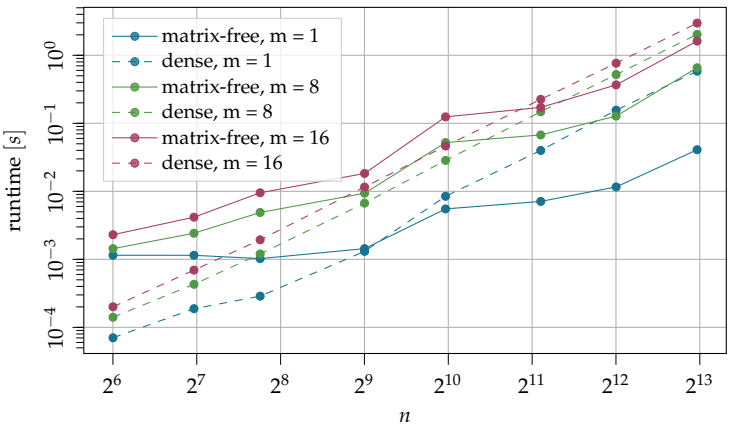


Figure 3.8 *Multilevel Toeplitz Matrices have a fast matrix-free representation.* – Comparison of the runtime between the dense and matrix-free implementation for $\boldsymbol{T}_{[k,k,k],3} \cdot \boldsymbol{x}$ for $\boldsymbol{x} \in \mathbb{C}^{k^3 \times m}$ with $k^3 = n$. ∎

With the methods of the previous section in (3.11) we can diagonalize each $\boldsymbol{H}_{i,j}$. To this end, we use (3.13) to embed each $\boldsymbol{H}_{i,j}$ into a 2-level circulant matrix

$$\begin{bmatrix} \mathfrak{T}_2(\boldsymbol{H}_{0,0}) & \dots & \mathfrak{T}_2(\boldsymbol{H}_{0,M-1}) \\ \vdots & \ddots & \vdots \\ \mathfrak{T}_2(\boldsymbol{H}_{M-1,0}) & \dots & \mathfrak{T}_2(\boldsymbol{H}_{M-1,M-1}) \end{bmatrix} = \boldsymbol{K}^{\mathrm{H}} \cdot \boldsymbol{D} \cdot \boldsymbol{K}. \tag{3.15}$$

Here, we have defined $\boldsymbol{\mathcal{F}} = \boldsymbol{F}_{2N_1-1} \otimes \boldsymbol{F}_{2N_1-1}$ essentially denoting a 2D Fourier transform, $\boldsymbol{K} = \boldsymbol{I}_M \otimes \boldsymbol{F}$s and

$$\boldsymbol{D} = \begin{bmatrix} \mathbf{diag}(\boldsymbol{\mathcal{F}} \operatorname{vec} \tilde{\boldsymbol{h}}_{0,0}) & \dots & \mathbf{diag}(\boldsymbol{\mathcal{F}} \operatorname{vec} \tilde{\boldsymbol{h}}_{0,M-1}) \\ \vdots & \ddots & \vdots \\ \mathbf{diag}(\boldsymbol{\mathcal{F}} \operatorname{vec} \tilde{\boldsymbol{h}}_{M-1,0}) & \dots & \mathbf{diag}(\boldsymbol{\mathcal{F}} \operatorname{vec} \tilde{\boldsymbol{h}}_{M-1,M-1}) \end{bmatrix},$$

where each $\tilde{\boldsymbol{h}}_{i,j}$ for $i, j = 0, \dots, M-1$ is chosen such that they contain the defining elements of the corresponding 2-level circulant matrix $\mathfrak{T}_2(\boldsymbol{H}_{i,j})$. The decomposition in (3.15) states that the block 2-level matrix $\boldsymbol{H}$ can be decomposed into a product consisting of two Fourier transforms by means of $\boldsymbol{K}$ and $\boldsymbol{K}^{\mathrm{H}}$ respectively and the block diagonal matrix $\boldsymbol{D}$. So, both $\boldsymbol{K}$ and $\boldsymbol{D}$ allow for efficient $\phi_{\boldsymbol{K}}$ and $\phi_{\boldsymbol{D}}$ such that by means of (3.6) the matrix $\boldsymbol{H}$ also allows to derive $\phi_{\boldsymbol{H}}$. The results in terms of performance are given in Figure 3.9.

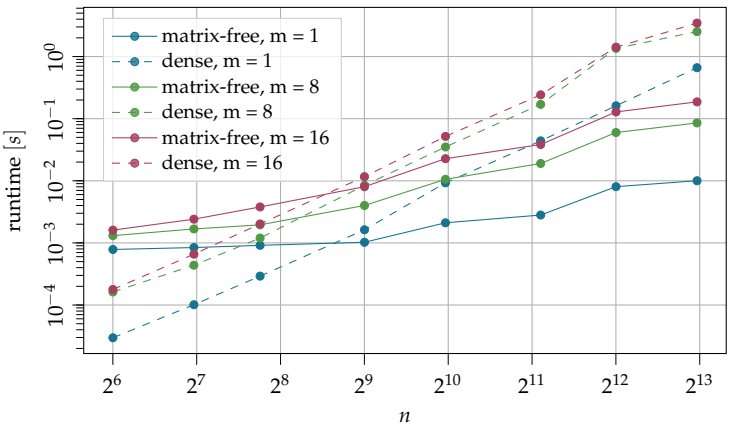


Figure 3.9 *Block Multilevel Toeplitz Matrices have a fast matrix-free representation.* – Comparison of the runtime between the dense and matrix-free implementation for $\boldsymbol{H} \cdot \boldsymbol{x}$ for $\boldsymbol{x} \in \mathbb{C}^{k^3 \times m}$ with $k^3 = n$. ■

3.3.3 Further Examples for Matrix-Free Representations

To further motivate the development of a software package in Section 3.4 we argue that quite a substantial amount of matrices occurring in practice have an efficient matrix-free representation. Some trivial examples are the identity matrix $\boldsymbol{I}_n \in \mathbb{R}^{n \times n}$, where in fact $\phi_{\boldsymbol{I}_n} = \beta_{\boldsymbol{I}_n} = \mathbf{id}$, where $\mathbf{id}$ is the identity mapping on $\mathbb{R}^n$, holds. Also, the zero matrix $\mathbf{0}_{n \times m}$ has as matrix-free representation simply the respective zero-mapping on $\mathbb{R}^n$ or $\mathbb{R}^m$ respectively.

Harmonic Analysis

When it comes to harmonic analysis several examples come to mind, like the Fast Hadamard Transform (FHT) [76] for spectral analysis over finite fields, the Discrete Cosine Transform (DCT) [79] or various versions of Wavelets [80] being the bridge between spectral and time analysis.

Sparse Matrices

When dealing with graph structures containing few edges, the incidence and adjacency matrix are naturally sparse and their eigenvalues and eigenvectors are of interest to derive properties of the graphs under study, see for instance [81]. Also, when discretizing the domain for solutions of partial differential equations [63] one usually has to solve (generalized) eigenvalue problems in order to derive solutions to these equations, which involve matrices with only a few non-zero entries. Hence, the efficient treatment of sparse matrices has sparked the development of various data structures for sparse matrices.

Inverse Matrices

If one is confronted with calculating expressions like $\boldsymbol{M}^{-1}\boldsymbol{x}$ one usually is inclined to fall into despair, since the computational cost of calculating $\boldsymbol{M}^{-1}$ is expected to be too high or numerical unstable. However, this is not necessarily the case, since one can also apply a suitable numerical algorithm to solve $\boldsymbol{M}\boldsymbol{y} = \boldsymbol{x}$ for $\boldsymbol{y}$, as it satisfies $\boldsymbol{y} = \boldsymbol{M}^{-1}\boldsymbol{x}$. Additionally, for solving a system of linear equations, there are many specialized and well studied numerical algorithms [72, 82], which only rely on $\phi_{\boldsymbol{M}}$ and $\beta_{\boldsymbol{M}}$. Thus, these algorithms can leverage the performance of the matrix-free representation.

Although the above approach is quite clear, it usually is not well known that a similar fact holds for the pseudo-inverse matrix $\boldsymbol{M}^{\dagger}$. In order to calculate $\boldsymbol{M}^{\dagger}\boldsymbol{x}$, one can solve

$$\min_{\boldsymbol{y}} \|\boldsymbol{M}\boldsymbol{y} - \boldsymbol{x}\|_2^2$$

for instance by means of the algorithm presented in [83], which again only relies on $\phi_{\boldsymbol{M}}$ and $\beta_{\boldsymbol{M}}$. Once the solution $\boldsymbol{y}^*$ is obtained one has that $\boldsymbol{y}^* = \boldsymbol{M}^{\dagger}\boldsymbol{x}$ and the explicit calculation of $\boldsymbol{M}^{\dagger}$ is avoided.

In addition, Figure 3.10 indicates that the numerical error, when calculating $\phi_{\boldsymbol{M}^{-1}}$ by solving a system of linear equations is lower compared to explicitly inverting $\boldsymbol{M}$ and then using the regular matrix-vector product.

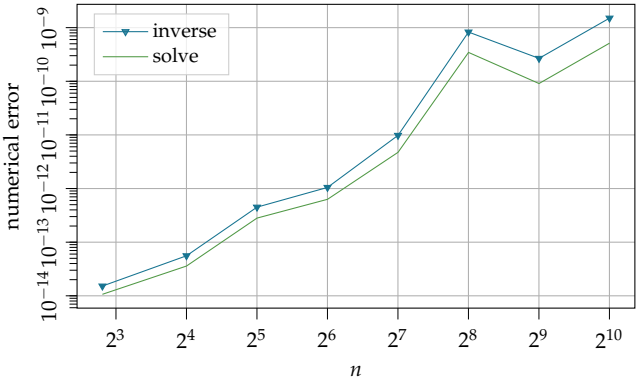


Figure 3.10 *Solving a system of linear equations can be more accurate than inverting the system matrix.* – Comparison of the numerical error of solving the system of linear equations (green) or first inverting $\boldsymbol{M}$ and then using $\boldsymbol{M}^{-1}\boldsymbol{x}$ (blue). ■

3.3.4 Conclusion

As we have seen, the matrix-free representation of linear transformations allows the derivation of efficient algorithms for various types of matrices and even compositions of these representations can be considered as an efficient representation for these. If applied carefully, these representations allow faster algorithms that are based on matrix-vector products and as such can yield memory and time efficient implementations in these scenarios.

However the reason why the dense representations often still are the initial approach is largely due to the fact that software and libraries used in scientific computing are designed around the use of the representation of linear transforms by 2D data structures. So, from a researcher's perspective the overhead in designing the algorithms for the presented matrix-free representations is a road paved with obstacles in the form of implementation complexity, bugs, maintenance issues and testing.

To address these obstacles we describe a software architecture for the construction and composition of linear transforms that use the matrix-free representation in the following section.

3.4 fastmat: A Library for Matrix-free Operators

In this chapter we outline the design principles, features and performance of an openly available software package called `fastmat` [O2, O3]. This is sparked by the insights gained in the previous sections which indicate that many practically relevant linear mappings have an efficient matrix-free representation and result in direct benefits for CS when employed in practice.

We first give some reasons why a coherent software framework for this alternative representation can come in handy. Based on these and the previously given motivations we derive some design criteria we adhered to when building the architecture and finally we present and benchmark a short and simple toy example for efficient SSR used in CS. More elaborate examples for the use of `fastmat` in practice are given later in Chapter 4. Finally, we conclude with comparing the proposed software to other similar implementations and even show how to combine it with other libraries in the Python [84] ecosystem.

3.4.1 Motivation

After we have dealt with the handling of the two representations of a matrix $\boldsymbol{M}$ – by means of referring to $\boldsymbol{M}$ and $\phi_{\boldsymbol{M}}$ or $\beta_{\boldsymbol{M}}$ – it is clear that depending on the situation either one of the two representations is more convenient or better reflects the intention of the author. For example, (3.9) and (3.8) differ in their transported meaning although they reflect the same concept of a fast convolution based on the Fourier transform. In other words, $\boldsymbol{\Gamma}(\boldsymbol{c}) \cdot \boldsymbol{x}$ and $\phi_{\boldsymbol{\Gamma}(\boldsymbol{c})}(\boldsymbol{x})$ reflect the same thing with different connotations on what the author would like to express. However, the notation relying on ϕ and β quickly becomes cumbersome and unintuitive to work with, due to the lack of freedom in defining these mapping and the missing flexibility of the rules to manipulate algebraic expression involving matrices.

This means that at this point the scientific notation necessarily deviates from the implementation that one resorts to when carrying out simulations or other numerical experiments, since the matrix-free representation allows for faster code and less memory usage. The more complicated the structure of the involved linear mappings becomes, the greater the divide between notation and implementation.

In other cases, one might be aware of the structure of the matrices but the adaption of existing code to this representation is too cumbersome, or

the programming skills of the scientist are not advanced enough to dare such an error prone endeavor in the first place. Then, one usually sticks with the inefficient implementation, which quickly limits the treatable problem size and hence the accuracy of the numerical evaluation.

Even existing implementations, aiming at a matrix-free representation, are often hard to maintain. Since not all corner cases can and will be considered, implementing tests poses too much work, code is often written in an ad-hoc-fashion for a single purpose, or the underlying design is not flexible enough to adapt for slightly different structural assumptions, which naturally pop up during scientific exploration.

This motivated us to design an open source[3] package called `fastmat` for the Python programming language that aims at providing a remedy to the previously described problems. This is achieved by providing a set of routines to build up lazily evaluated linear operators based on structural knowledge, which on the other hand can be treated like general matrices in order to facilitate code that comes close to the scientific notation used in publications. When providing this flexibility, we aim at maintaining most of the speedups and memory efficiencies facilitated by the matrix-free representation, while also designing the whole architecture for easy extendability. The following sections describe the main aspects in terms of design principles, architecture and resulting features.

3.4.2 Architecture

In this section we highlight only some aspects of the provided package and we refer to the documentation[4] or the respective software publication [O3] for a complete overview on the matter. Also note that the presented code examples do not reflect the actual implementation but rather give insights in the general approach one should take.

[3] Released under the rather permissive Apache-2.0 license, see `https://www.apache.org/licenses/LICENSE-2.0`.

[4] See `https://fastmat.rtfd.io`.

Main Dependencies

In order to allow an easy integration into the rest of the Python ecosystem for scientific computing we rely on the Numpy [85] library, which provides a multi-dimensional data structure and a vast amount of operations involving this data structure. These operations are often implemented in Fortran [86] or C [87] in order to avoid the costly influence of the Python interpreter during these array calculations, hence one can make use of very efficiently running algorithms with the convenience that is provided by the high-level features of Python. In addition to Numpy, there is SciPy [88] that provides even more algorithms for advanced linear algebra, optimization and signal processing. Finally, we depend on Cython [89] to statically compile most Python(-like) code into C, which allows to partially circumvent the Python interpreter and hence the associated overhead. Especially for small transforms the runtime is mostly affected by said interpreter, which drives the need to reduce its influence.

Objected Oriented Design

The concept of Object Oriented Programming (OOP) allows to naturally reflect the hierarchical structure of linear transforms in the codebase itself. As a starting point, we define a base class `fastmat.Matrix`, which represents a linear transform with *no* structural assumptions. As such it has two functions: First, it represents the Application Programmable Interface (API) provided by `fastmat` to define linear operators and the associated functionalities that are available for these. Second, it also implements a linear transform that is in fact represented by a matrix, which is stored as a 2D `numpy.ndarray`, so that

$$\phi_{\boldsymbol{M}}(\boldsymbol{x}) = \boldsymbol{M} \cdot \boldsymbol{x} \quad \text{and} \quad \beta_{\boldsymbol{M}}(\boldsymbol{x}) = \boldsymbol{M}^{\mathrm{H}} \cdot \boldsymbol{x},$$

where $\cdot$ represents the conventional matrix-vector product. These two functions are accessible to the user via `fastmat.forward(x)` and also via `fastmat.backward(x)`. Then, any other linear transform making structural assumptions that results in more efficiently running computations simply defines a new class that is derived from `fastmat.Matrix` and replaces `fastmat.forward` and `fastmat.backward`. This process is outlined in Listing 3.1. One can proceed similarly for other methods, like for the calculation of the eigenvalues or eigenvectors, which would be straightforward in the example given in Listing 3.1.

This approach allows the developer to decide which methods he deems worth of overriding and which he reuses from the base class implementation. Hence, one can focus on the methods that are needed for the

```
from fastmat.matrix import Matrix

class Doubler(Matrix):
  def forward(self, x):
    return (2 + 2j) * x

  def backward(self, x):
    return (2 - 2j) * x

  def __init__(self, numRow:int, numCol:int):
    self._initProperties(
      self, numRow, numCol, np.int8
    )
```

Listing 3.1 *It is easy to implement classes that derive from `fastmat.matrix`.* – Example for the definition of custom class derived from `fastmat.matrix`. The `_initProperties` method internally sets the minimum properties of the `Matrix` instance. ■

implementation at hand without the need to completely implement all existing parts of the API provided by `fastmat.Matrix`.

Other Architecture Aspects

Each `fastmat.Matrix` has an associated data type, which for instance allows linear transforms with integer-valued matrices to operate very efficiently on integer valued vectors. Both in terms of runtime and also in terms of required memory.

Additionally, through calibration we are able to roughly infer if it is more efficient to use the matrix-free or the dense representation of a given transform. Hence, we have means of internally determining what in a certain case is the optimal execution path, which can be handled without major intervention by the user.

In order to provide a basic interface for algorithms that depend on `fastmat.Matrix`, we also defined a small API for those in order to facilitate easy monitoring during execution and other conveniences. Further, there is a tool called (worker)-`bee` that is able to automatically carry out tests of all class implementations for various data types and matrix sizes. Additionally it allows the convenient generation of benchmarking results. All these features are also easily accessible by the user.

```
from fastmat.matrix import Matrix
import numpy as np

class Fourier(Matrix):
  def forward(self, x):
    return np.fft.fft(x)

  def backward(self, x):
    return np.fft.ifft(x)
```

Listing 3.2 *Fourier matrices use the* `fft` *and specialize* `fastmat.matrix`. – Simplified example for the implementation of the Fourier matrix. ■

3.4.3 Implemented Matrices

Conceptually, one can differentiate between three different types of linear transforms that we provide. We shortly outline them and give some examples by means of the already implemented transforms. Note that these are not fixed definitions and depending on the view taken with respect to a certain transform, one can put some of them into different or multiple of the provided categories.

Explicit Structures

In Example 3.1 we already encountered an explicit structure in the form of the DFT matrix. These structures usually are not composed of other structures, but are often depending only on the definition of the size, like the Fourier (`fastmat.Fourier`) matrix, or the Hadamard (`fastmat.Hadamard`) matrix. Otherwise, they can also depend on vectors or arrays, like diagonal matrices (`fastmat.Diag`). We also consider sparse (`fastmat.Sparse`) matrices or permutation (`fastmat.Permutation`) matrices to be members of this structure type. See Listing 3.2 for a sketch of an implementation of Example 3.1.

```
from fastmat.matrix import Matrix

class Hermitian(Matrix):
  def forward(x):
    return self._M.backward(x)

  def backward(x):
    return self._M.forward(x)

  def __init__(self, M):
    self._M = M
```

Listing 3.3 *Hermitian transposition simply switches* `fastmat.backward` *and* `fastmat.forward`. – Simplified example for the implementation of the Hermitian transpose of a matrix. ■

Expression Structures

Once one has some explicit structures at ones disposal, one often would also like to combine them to more involved expressions. Usually, one forms various products (`fastmat.Product`, `fastmat.Kron`) of matrices (see Section 3.3.1), sub-selects some columns or rows (`fastmat.Partial`), wants to access the inverse matrix or its Hermitian transpose (see Equation (3.8)). In Listing 3.3 we show how one could implement the Hermitian transpose of a given matrix $\boldsymbol{M}$.

Note that due to the fact that the resulting composite matrices are all instances of a common base-class, one can combine these composite instances easily in other compositions. See for instance Listing 3.4, where also the provided operator `fm.Matrix * fm.Matrix` for the matrix-matrix product is demonstrated.

Composite Structures

With these compositions at hand, one can also save a lot of implementational overhead, because some matrix structures expose their efficient transform by means of a smart factorization. For instance, if we consider the decomposition of multi-level circulant matrices in (3.11), which reads as

$$\boldsymbol{C}_{\boldsymbol{n},d} = \bigotimes_{i=1}^{d} \frac{1}{n_i} \boldsymbol{F}_{n_i}^{\mathrm{H}} \operatorname{diag}\left(\bigotimes_{i=1}^{d} \boldsymbol{F}_{n_i} \cdot \operatorname{vec} \boldsymbol{c} \right) \bigotimes_{i=1}^{d} \boldsymbol{F}_{n_i}. \tag{3.16}$$

```
import fastmat as fm

C = fm.Circulant(c)
T = fm.Toeplitz(t)
M = fm.Matrix(m)

P = fm.Inverse(fm.Kron(C * T, M))
```

Listing 3.4 *Easy compositions of compositions.* – We form a matrix-matrix product of a circulant and a Toeplitz matrix, which is contained in a Kronecker-product with an unstructured matrix and this Kronecker-product is inverted. ■

This formula motivates us to not implement it explicitly as above, but rather as outlined in Listing 3.5.

There we make use of `fastmat.Kron` and `fastmat.Fourier` to define a derivation of `fastmat.Product` to implement `fastmat.Circulant` without the need to explicitly define the `forward` and `backward`. Instead we have maximized the reuse of code already provided in the package. From an implementational point of view, this is preferable as long as one does not need more performance than what is provided by the classes that make up the circulant class.

These few concepts should suffice to present the general idea behind the architecture of the package and the implementations that this architecture allows. The next section briefly mentions some more features that are not explained in detail.

3.4.4 Additional Features

Due to the imposed structure of the individual classes often additional quantities can be calculated easily. The API allows to provide efficient methods for the calculation of column norms, row norms, singular values and vectors as well as eigenvalues and the respective vectors.

Since the authors are mostly concerned with CS as an application of `fastmat` in practice, we provide some compressed sensing algorithms that can exploit fast matrix multiplication, namely ISTA, FISTA, STELA and OMP. Hence, we explictily implemented OMP and FISTA according to Algorithm 3.1 and Algorithm 3.2, respectively. For the algorithms we also chose an OOP based approach to provide basic functionality like testing, benchmarking and logging during the execution for all derivations from a specifically for this purpose designed `Algorithm` base class.

```
import fastmat as fm

class Circulant(fm.Product):
  def __init__(self, c: np.ndarray):
    self._c = c
    self._F = fm.Kron(c.ndim * (fm.Fourier, ))
    self._chat = self._F * c.flat

    factors = [
      self._F.H,
      fm.Diag(self._chat.flat),
      self._F
    ]
    super().__init__(self, *factors)
```

Listing 3.5 *Circulant matrices are just a* `fastmat.Product` *with specifically structured factors.* – Based on the defining vector $\boldsymbol{c}$ we derive `fastmat.Circulant` from `fastmat.Product` and just explicitly define its factors. ∎

3.4.5 A Simple Representative Example

In order to illustrate how conveniently one can work with `fastmat` matrices to implement a CS scenario, we give a complete example in Listing 3.6. There we construct a signal that consists of $S = 3$ superpositioned and shifted $x_i \cdot \cos(2t - \tau_i)$ for unknown τ_i and x_i. Hence, $\boldsymbol{A} \in \mathbb{R}^{512\times512}$ has circulant structure and we compress this sampled signal additionally with a random Gaussian matrix $\boldsymbol{\Phi} \in \mathbb{R}^{256\times512}$ to get $\boldsymbol{z} \in \mathbb{R}^{256}$ from which we aim to reconstruct $\hat{\boldsymbol{x}}$ using Algorithm 3.1 running for $12S$ steps as indicated by Theorem 2.7.

The code already highlights the fact that it would be easy to employ a different dictionary $\boldsymbol{A}$, a different compression scheme $\boldsymbol{\Phi}$ or another reconstruction algorithm. All these building blocks can be exchanged independently, while the whole processing pipeline still performs well, since the matrix-free representation can be upheld.

```
import numpy.linalg as npl
import numpy as np
import fastmat as fm
import fastmat.algorithms as fma
# define the dimensions and the sparsity
n, S = 512, 3
# define the sampling positions
t = np.linspace(0, 20 * np.pi, n)
# construct the convolution matrix
c = np.cos(2 * t)
A = fm.Circulant(c)
# create the ground truth
x = np.zeros(n)
x[npr.choice(range(n), S, replace=0)] = 1
# compression as unstructured fm.Matrix instance
Phi = fm.Matrix(
  np.random.randn(256, 512)
)
y = A * x
z = Phi * A * x
# reconstruct it
omp = fma.OMP(Phi * A, numMaxSteps=36)
xhat = omp.process(b)
# check reconstructed signal
print(npl.norm(A * xhat - y))
```

Listing 3.6 *fastmat makes SSR for CS very convenient.* – A complete example for a CS scenario, where $\boldsymbol{A}$ is a circulant matrix and $\boldsymbol{\Phi}$ is a random Gaussian matrix and we use OMP to recover $\hat{\boldsymbol{x}}$. ■

3.4.6 Relation to other Software and Libraries

Since matrix-free representations have been around for several decades now, especially in high performance scientific computing, the SciPy [88] library has implemented a `LinearOperator` class, which works quite similar to `fastmat` in the sense that the `LinearOperator` class also provides the `forward` and `backward` routines. This allows many algorithms in `scipy.sparse.linalg` to take these linear operators as an input instead of a dense system matrix. Recently, this approach has been ported to the Graphical Processing Unit (GPU) by means of CuPy [90], adopting similar features.

The algorithms for solving a system of linear equations and the algo-

rithm for least-squares, as they are offered by SciPy, are used to implement `fastmat.Inverse` and `fastmat.PseudoInverse` respectively. However, what is missing in SciPy and Cupy is the possibility to conveniently generate these instances of `fastmat.LinearOperator`. To address this, the `fastmat.Matrix` class allows to generate a matrix-free representation that is used by SciPy in the form of `scipy.LinearOperator` directly, in order to make the algorithms in SciPy readily available to users aiming to implement their transforms in `fastmat`. This allows a tight integration with the rest of the computing ecosystem in Python.

Finally, there are also similar libraries that make use of the present matrix-free approach. A path similar to ours has been taken by the PyLops [91] developers, which directly instantiate from the already mentioned `scipy.LinearOperator` and provide methods to combine these instance to new linear operator represented by SciPy, hence allowing a more tightly integrated experience to SciPy. For more low-level programming languages, like C++, there also exist tool boxes to generate matrix-free representations, like the library provided by the Rice university [92].

3.5 Conclusion

The presented software has already been used successfully in several publications, where [O10], [93], [O8], [O9], [O11], [94] are mostly concerned with ultrasonic non-destructive testing. Additionally, in [O6] we used the implementation of the OMP algorithm for the sparse recovery step. In [O12] we also employed the OMP algorithm together with the provided circulant matrix structure origination from a Linear Feedback Shift Register in order to quickly carry out the simulations. Very recently, the flexibility of `fastmat` was exploited in a tomography technique [O15] using a sparse reconstruction step. Also, the software is used by the free Pygpc [95] package for the sparse reconstruction stage[5]. Additionally, [96] exploits the provided `fastmat.Kron` and `fastmat.Toeplitz` structures for their matrix-free computations on (for instance) high-resolution radar weather data[6].

Hence, the described `fastmat` library can readily be applied in a practical context and as such presents a viable option when matrix-free compressive sensing has to be made use of in order to avoid unnecessary computational overhead. The following chapter, which focuses on two

[5] See `https://github.com/pygpc-polynomial-chaos/pygpc`
[6] See: `https://github.com/ymohit/fkigp/`

different applications also draws heavily from the provided software, when producing numerical simulation results.

There are also several ways how to extend the current architecture. First, with the recent advances in computing power available on GPUs, one should take into consideration to allow that computations can be carried out on this very different hardware. In this case the CuPy library ise of high value, due to its similarity to the Numpy package in terms of its API. Second, there are already many Python libraries that implement specific linear transforms matrix-free, like the Radon transform[7] or Wavelets[8]. If one would provide a thin wrapping layer around these libraries, the number of available transforms could be increased easily.

[7] See `https://scikit-image.org/docs/dev/auto_examples/transform/plot_radon_transform.html`

[8] See `https://pywavelets.readthedocs.io/en/latest/`

Chapter 4

Imaging Applications

> "Es ist alles eine Frage des Ursprungs. Wo ist der Anfang? Wann ist der Anfang? Gibt es überhaupt einen Anfang? Die Welt ist voll von solchen Paradoxen. Wir entschließen uns nur, die meiste Zeit wegzusehen."
>
> *Dark*

Up until this point we have presented known and developed some new theoretical results for CS. We have derived CS as an alternative signal processing approach to traditional Nyquist sampling and have theoretically treated the aspects of signal acquisition, reconstruction guarantees and algorithmic concepts of this advanced sampling paradigm. Additionally, we now have software at our disposal that can handle large CS scenarios very easily. Hence, we are in the position to put the presented and newly developed theory to work in the sense that we tackle two different applications that make use of these previous results.

First, we present the problem of estimating the locations of material defects by means of ultrasonic non-destructive testing. For this example we will outline a comprehensive processing pipeline by conceptualizing an acquisition scheme based on a volumetric model for the observation, determining the resulting estimation accuracy and dealing with algorithmic aspects of a feasible implementation for the reconstruction step. Ultimately, we show that by means of the proposed architecture one gains a flexible system that can deliver better imaging quality than previously established processing schemes while the proposed sensing procedure also considers the possibility to implement. Summarizing, we contribute

to AOI-M, AOI-C, AOI-R and AOI-H and the presented results have been published to a great extend in [O11].

Second, we develop a compressive Ultra-Wideband (UWB) radar system that extends a state of the art UWB radar system by means of a random demodulator to realize compressive measurements directly in the analog domain. We demonstrate the expected reconstruction performance and also show how to make use of matrix-free representations to speed up the reconstruction step. As we derive, this proposed system allows to increase the allowed Doppler-shift during the acquisition stage compared to the previous state of the art. This renders it more robust to the effect of time varying channels. Hence, we make significant contributions to AOI-C and AOI-H and the results have been published in [O12] and also in the form of a patent application [P1].

4.1 Ultrasonic Non-Destructive Testing from a few Fourier Samples

In Ultrasonic Non-destructive Testing (US-NDT), a widespread approach is to take synthetic aperture measurements from the surface of a specimen to detect and locate defects within it. Based on these measurements, imaging is usually performed using the Synthetic Aperture Focusing Technique (SAFT). However, SAFT is sub-optimal in terms of resolution and requires oversampling in time domain to obtain a fine grid for the Delay-and-Sum (DAS) implementation. On the other hand, parametric reconstruction algorithms give better resolution, but their usage for imaging becomes computationally expensive due to the size of the parameter space and the large amount of measurement data in realistic 3-D scenarios when using oversampling.

In the literature, the remedies to this are twofold: First, the amount of measurement data can be reduced using state of the art sub-Nyquist sampling approaches to measure Fourier coefficients instead of time domain samples. Second, parametric reconstruction algorithms mostly rely on matrix-vector operations that can be implemented efficiently by exploiting the underlying structure of the model.

In this chapter, we propose and compare different strategies to choose the Fourier coefficients to be measured as outlined in Section 2.4.2.1 and hence we contribute to AOI-C. Their asymptotic performance is compared by numerically evaluating the CRB for the localizability of the defect coordinates. These sub-sampling strategies are then combined with ℓ_1-minimization scheme by means Equation (2.14) to compute 3D reconstructions from the low-rate measurements. Compared to conventional DAS, this allows us to make use of a physically motivated forward model matrix, which contributes to AOI-M. To enable this, the projection operations of the forward model matrix are implemented matrix-free by exploiting the underlying block 2-level Toeplitz structure as introduced in Section 3.3.2.5. Finally, we show that high resolution reconstructions from as low as a single Fourier coefficient per A-scan are possible based on simulated data as well as on measurements from a steel specimen where we also compare two different reconstruction approaches based on Algorithm 3.1 and Algorithm 3.2 respectively, hence we address AOI-R. We also show that the employed compression scheme can be readily implemented in the analog domain, which hence takes into account AOI-H.

Summarizing, based on the publications [O9, O11, O16, O17, O10, O8]

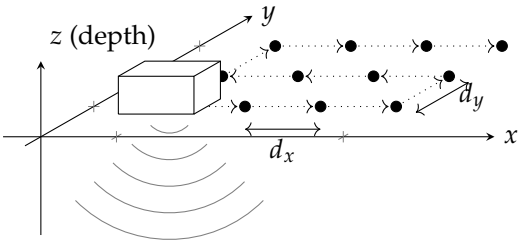


Figure 4.1 *Creating a synthetic aperture by moving a transducer* – Visualization how a regular 2D synthetic aperture is acquired by means of a transducer. ■

we present a modular processing pipeline, where we demonstrate how to attain different realizations of this pipeline and discuss the respective advantages and drawbacks. Additionally, we give certain design criteria which building blocks for the processing to use.

4.1.1 Introduction

4.1.1.1 State of the art

In ultrasound Nondestructive Testing (NDT) defects are detected and localized by inserting an ultrasonic pulse into a specimen and collecting the resulting echo signals [97]. A typical measurement setup consists of a single transducer or transducer array that is used both as transmitter and receiver. In the single channel case the transducer is moved on the specimen surface and measurements are collected at each scanning position to form a so-called synthetic aperture, see Figure 4.1[1].

The measurements in these setups are usually acquired in time domain and sampled at a frequency much higher than the Nyquist frequency to create a dense time grid that the subsequent DAS can use to achieve a well-resolved reconstruction. An image of the specimen interior is then commonly computed based on the SAFT [98] or its multi-channel extension [99]. More recently, model based approaches have been introduced, where the main idea is to treat the reconstruction as an inverse problem based on a physically derived model matrix that is solved using standard tools from linear algebra [100]. This enables us to include more complex mechanisms of wave propagation that cannot be captured by simple DAS such as attenuation and temporal dispersion [101] or acoustic shadow-

[1] I thank F. Krieg for the TiKZ code that produces this graphic.

ing [102]. Additionally, one can account for the pulse shape or elementary signature [103], which is usually modeled as the real part of a Gaussian windowed sinusoid [104].

The mentioned oversampling in time domain leads to a large amount of measurement data, which in some cases may result in a computationally expensive reconstruction, especially when it comes to 3D imaging [O9]. However, parametric models can be combined with CS techniques [11] to reduce the amount of measurement data. Initially, mostly random Gaussian measurement kernels as introduced in Section 2.4.1.1 have been investigated [105] for this compression step, since for those, theoretical guarantees like those in Equation (2.19) can be established ensuring that the reconstruction is robust and stable. However, as we argue in Section 2.4.2.1, the design of a generic CS hardware that implements these Gaussian measurements is a challenging task [106].

To reduce the amount of measurement data per incoming echo signal, prior work has shown that highly structured signals such as ultrasonic recordings can be completely recovered in frequency domain from samples taken at rates significantly below the Nyquist-rate as given in Theorem 2.1, provided that they are measured with an appropriate sampling architecture [15]. Based on the recovered Fourier coefficients, different reconstruction strategies can be employed. One approach is to use beamforming in frequency domain [107]. Another approach is to formulate the reconstruction as an inverse problem based on a physically motivated forward model and extend this model by the compression scheme. In the CS context, sub-sampled Fourier measurements can guarantee stable and robust reconstruction since a randomly generated partial Fourier matrix fulfills the D-RIP [21, 41, 40] as we have outlined in Theorem 2.9. It has been shown that high quality 3D images based on synthetic aperture measurements can be reconstructed from only a small number of Fourier samples collected at each measurement position [O10]. Additionally, the acquisition of Fourier coefficients is an attractive choice, as hardware architectures to measure them directly already exist [108, 109, 110, 111]. In multi-channel setups, the compression is often realized by sub-sampling the channels of a large array [112, 113, O16], which exploits the same subsampling principles as they are used in sparse array design.

Often, the sparsity is assumed in the parameter domain as the number of defects inside the specimen is small [114]. Hence, compressed observations can be post-processed by means of SSR to solve the parameter estimation problem at hand using methods like those in Algorithm 3.1 or Algorithm 3.2. These techniques have proven to produce images with

superior quality [103] and better resolution of closely spaced defects [115] compared to images using only DAS.

To reduce the computational effort that arises when aiming for high estimation quality, efficient implementations exploiting the structure of the involved linear mappings need to be employed [116]. Often these structure exploiting algorithms allow a high degree of parallelization so that further speed-ups are possible by the use of a GPU [O9, 117].

Additionally, a parametric model allows to analyze the influence of the parameters on the measurements. One way to do so is to study the estimation variance by means of the CRB [118], which we introduced in Section 2.4.4. In the ultrasound field, the CRB has been used to quantify time delay estimation jitter and the impact of the choice of bandwidth and center frequency of ultrasound pulse-echo measurements [119, 120], and the achievable resolution when locating single point-scatterers [121]. Moreover, the CRB has been used as a criterion for array design and sensor placement [122, 123] as well as to optimize spatial sub-sampling patterns in CS [O17].

4.1.1.2 Motivation

For data acquisition we focus on synthetic aperture ultrasound measurements in NDT. A single transducer is placed on the planar surface of a specimen and moved to different positions on a regular 2-D grid, see Figure 4.1. Specifically, we are interested in scenarios where a high resolution 3D reconstruction based on these measurements is required, for example to resolve closely spaced scatterers. Since the goal is to detect defects, we need to scan the whole region of interest leading to a large number of measurement positions and therefore a large amount of measurement data.

A Three-Stage Model for Inspection Measurements

In order to link our signal processing architecture with the task of inspection from acquired measurement, we introduce an abstract model that distinguishes the data processing into several steps. Inspection measurements can be described using a three-stage model, comprised of (1) data management (collection and storage of measurement data), (2) sense making (data analysis, analytics and feature extraction) and (3) decision making [124]. From stage to stage, data volume reduces while data value increases.

For manual inspections, a trained technician usually performs tasks of all model stages in full during inspection. However, expertise is only required for decision making in the final stage, leaving the (on-site) data-management and (off-site) sense-making stages up for automation. As a result, the required effort for on-site inspections can be reduced greatly. However, since expertise is no longer involved prior to the final decision making stage, data filtering also can no longer be applied in earlier stages. This not only leaves an overwhelming amount of unfiltered data to handle, but also clogs the computationally intense processes of the sense-making stage. To illustrate the amount of data gathering required, a small example size with a scanning grid of 100×100 locations and an A-scan time-length of 1000 samples is already sufficient. With 16 bit data quantization, one full synthetic aperture measurement requires 20 MB of data. Further assuming 100 different measurement positions for a single inspection, a total volume of 2 GB of data accumulates. To reduce the amount of measurement data, a natural approach in this scenario is to minimize the amount of measurement samples per scan position without losing relevant information. Instead of measuring on a dense grid in time domain (sampling frequencies above 100 MHz are common, although the ultrasound pulse rarely exceeds a bandwidth of, say, 10 MHz), we can and should use existing hardware architectures [109, 111] to sample the data more efficiently. Based on Corollary 2.1 obtain Fourier coefficients from samples taken at a much lower rate.

The adoption of CS enables the modification of the three-stage model on the hardware side so that it mimics the tasks of the technician. Complex post-processing routines are only necessary as a final step prior to decision making, meaning the operations performed by the sensors can be simplified and hence streamlined. Reducing the data rate directly at the measurement stage has the added benefits of diminishing power requirements and enabling data streaming to a possibly remote processing unit [125] at much lower rates than without the compression step. Further, progression toward the so-called NDE4.0 [126] increases the attractiveness of embedded sensors for Structural Health Monitoring (SHM) [127] scenario in which power consumption severely constrains active sensors. ∎

A naturally arising question is: how many and which measurements are necessary for a given sub-sampling methodology to result in robust and stable reconstruction? This question is a specific version of the setting in Remark 2.4. Considering the pulse-echo of a single volume-element within the specimen, an answer to this question is already indicated by the observation that the echo caused by said volume-element is systematically

represented in a large number of adjacent A-scans. Describing the linear measurement model in all three spatial dimensions reveals a structure presenting large amounts of spatial redundancy between A-scans.

In [O8], this time-domain model is extended by sub-selecting only a few Fourier coefficients in frequency domain from each A-Scan, effectively representing a compression of the measurement data. The parameter estimation problem of finding defect positions based on these compressed measurements is formulated as an SSR problem, hence a specific version of (2.5) or (2.4). As this exhibits a high dimensional parameter space, it quickly becomes computationally expensive or even intractable. Standard linear algebra solvers that rely on explicit representations of the system matrix, already break down on moderate problem sizes, since dense or even sparse representations of the operator matrix quickly become infeasibly large. As illustration, for the given example the operator matrix is of size $(1000 \cdot 100^2)^2$, even exceeding the memory capabilities of current mainframe systems. Instead, it is possible to derive a matrix-free representation of the operator, which exploits the embedded structure of the matrix in a computationally- and memory-efficient way such that the methods in Section 3.2 become applicable. The proposed implementation is designed to be flexible in terms of modeling parameters (e.g. pulse shape, dimensions of the specimen, grid dimensions of the Region of Interest (ROI)) as well as the measurement strategy (different compression strategies, and/or uncompressed measurements).

In addition, most of the existing algorithms which rely only on matrix-vector products and which are similar to those in Section 3.2 usually have side constraints that implicitly require more knowledge about the matrix. Therefore, implementations need to be tweaked to account for this. To give an example, the FISTA requires the largest singular value of the system matrix in order to select the correct step size, which has to be obtained also in the case of the matrix-free representation. Although an approximation using ARnoldi PACKage (ARPACK) is possible in our framework, this adds a large computational overhead that in some cases can and should be omitted. Lastly, the implementation should be easily usable with different computation setups available, e.g. likewise on a GPU or Central Processing Unit (CPU). In order to accomplish this, one can rely on the architecture presented in Section 3.4.

In order to precisely quantify the effects of our estimation procedure, we investigate the proposed methodology based on the asymptotic and real-world performance. The asymptotic performance is given in terms of the CRB, as introduced in Section 2.4.4, of the resolution of locating

single point-scatterers. This is helpful, as the CRB directly quantifies how parametrizing the measurement setup, e.g. how many Fourier coefficients are measured, influences the parameter estimation. The real-world performance is evaluated based on the reconstruction of simulated as well as real ultrasound measurement data.

4.1.1.3 Main contributions

We develop and investigate a complete framework for high resolution ultrasound imaging based on a small number of Fourier measurements and a 3D data model representing the propagation of the sound waves in the medium. To obtain these measurements, we propose several sampling strategies and compare their performance.

We first derive a theoretic scaling law for synthetic aperture ultrasound measurements using the strategy of sampling *uniformly at random* based on Theorem 2.9 as introduced in Section 2.4.2.2. Specifically, we show that the required number of Fourier coefficients in this case only depends on the worst case sparsity of all scans, i.e. the scanning position that sees the maximum number of echoes, arguing that it does not exploit the correlation between adjacent scans. In comparison, the novelty of the sampling strategies proposed in our work is twofold. First, we propose to incorporate prior knowledge about the spectrum of the inserted pulse into the design of the sampling pattern.

This sparks two possible strategies, which we term *maximal* and *random energy-based* sampling. They alleviate the need for additional random sign-flips prior to the Fourier sampling as dictated by Theorem 2.9 and therefore simplify the hardware requirements when implementing the Fourier subsampling in the analog domain. Second, we exploit the description of the data in form of the proposed 3D model by varying the sampling pattern at each scan position using the *random energy-based* strategy. The strong spatial correlation between the measurements at adjacent scan positions leads to a trade-off between temporal and spatial measurements. However, since we need to scan the specimen at a certain minimum density to ensure defect detection, this can mainly be exploited to reduce the number of temporal samples. In fact, we show numerically that when the number of spatial scanning positions is large, only taking one but varying Fourier coefficient per A-scan position does not substantially decrease the CRB compared to the uncompressed case in a single defect scenario (or a scenario where the distance between defects is sufficiently large).

The ultrasound measurements are modeled using a parametric forward model that conveys both the ultrasound propagation as well as the received pulse shape. The analytic signal is used instead of only modeling the real-valued RF signal.

The matrix-free implementation for the reconstruction is achieved by exploiting the block-wise 2-level Toeplitz structure of the forward model. This enables to implement the effect of both the subsampling and the model in the reconstruction using only FFTs and indexing operations. The implementation is done in Python using the `fastmat` package from Section 3.4. The reconstructions are carried out via the matrix-free version of FISTA as given in Algorithm 3.2. The resulting matrix-free operator can not only be used for high resolution parametric reconstruction but also, by applying its adjoint on the measurements, yields a single-step "compressed SAFT" reconstruction that considers our assumptions about the model and the compression scheme.

With this implementation at hand, using simulated as well as measurement data it is shown that the spatially randomized strategies allow to produce 3D images from a single Fourier coefficient per A-scan allowing precise localization and sizing of several test defects.

The remainder of this chapter is organized as follows: In Section 4.1.2 we derive the ultrasound propagation model used throughout this paper. In Section 4.1.3, we introduce the novel Fourier acquisition schemes and derive a theoretic scaling law based on CS theory. In Section 4.1.4, we discuss the reconstruction process and provide concrete matrix-free algorithms for the implementation of the model and compression operator. Using FISTA as our example algorithm, we discuss practical solutions to approximate the required largest singular value based on a matrix-free software implementation. In Section 4.1.6, we compare the hardware requirements as well as the computation complexity of the proposed CS architecture compared to state of the art systems. In Section 4.1.5, we derive the single-scatterer CRB for the model and investigate the influence of the compression on the localization capability of the measurements asymptotically. Then, in Section 4.1.7, we provide example reconstructions from numerical simulations as well as using realistic measurement data to back up the theoretic claims. Finally, Section 4.1.8 concludes the paper.

4.1.2 Observation Model

4.1.2.1 Ultrasound data model

We consider a pulse-echo setup, where a single transducer is used to insert an ultrasonic pulse $p : \mathbb{R} \to \mathbb{R}$ into the specimen, which we also assume to be the received pulse. This means we assume dispersion in the medium to be negligible. The specimen is considered to be homogeneous and isotropic with constant speed of sound $c \in \mathbb{R}^+$ and to possess a flat surface. It contains $D \in \mathbb{N}$ point-like defects located at unknown positions $(x_d, y_d, z_d) \in \mathbb{R}^3$ for $1 \leqslant d \leqslant D$ that are to be localized. Note that now the sparsity S has been turned into D representing the number of defect. We assume that we can omit reflections from known features of the specimen (such as the back wall) by appropriate windowing while multiple scattering is ignored, such that only the D reflections stemming from the interaction of the incident field with the defects remain.

We first introduce a continuous model for the observations. In order to circumvent the limitations of real-valued band-pass signals that are prone to amplitude modulation due to phase offsets in the carrier, analytic signals are employed instead. Although such offsets can be handled even when working with real-valued signals, this choice serves the dual purpose of allowing the estimation and compensation of phase offsets more naturally, as will become evident in the following definitions. If we define $f^{(a)}$ as the analytic signal of a function f via

$$f^{(a)}(t) = f(t) + \jmath \mathcal{H}\{f(t)\}, \tag{4.1}$$

where $\mathcal{H}\{\cdot\}$ is the Hilbert transform of f, the analytic noiseless signal $b^{(a)}_{x,y} : \mathbb{R} \to \mathbb{C}$ received by the transducer from position $(x, y) \in \mathbb{R}^2$ can be modeled as

$$b^{(a)}_{x,y}(t) = \sum_{d=1}^{D} \alpha_d \cdot g_{x,y}(x_d, y_d, z_d) \cdot p^{(a)}(t - \tau_{x,y}(x_d, y_d, z_d)). \tag{4.2}$$

Here, $\alpha_d \in \mathbb{C}$ are the complex reflectivities and for each (x, y) $\tau_{x,y} : \mathbb{R}^3 \to \mathbb{R}^+$ is the function modeling time of flight from the transducer at sample position $(x, y) \in \mathbb{R}^2$ to the d-th reflector at position $(x_d, y_d, z_d) \in \mathbb{R}^3$ and back. It can be computed as

$$(x_d, y_d, z_d) \mapsto \tau_{x,y}(x_d, y_d, z_d) = \frac{2}{c}\sqrt{(x - x_d)^2 + (y - y_d)^2 + z_d^2}. \tag{4.3}$$

Additionally, $g_{x,y} : \mathbb{R}^3 \to \mathbb{R}^+$ represents the transducer characteristic, which models its directivity of the transducer at position (x, y) towards

the direction of (x_d, y_d, z_d). In frequency domain, we obtain

$$B_{x,y}(\omega) = \sum_{d=1}^{D} \alpha_d \cdot g_{x,y}(x_d, y_d, z_d) \cdot P(\omega) \cdot \mathrm{e}^{-\jmath\omega\tau_{x,y}(x_d,y_d,z_d))}, \tag{4.4}$$

where the function $P : \mathbb{R} \to \mathbb{C}$ is the Fourier transform of the pulse function p. To simplify the notation we introduce the atomic functions $h_{x,y}(x_d, y_d, z_d, t) : \mathbb{R}^4 \to \mathbb{C}$ as

$$h_{x,y}(x_d, y_d, z_d, t) = g_{x,y}(x_d, y_d, z_d) p^{(a)}(t - \tau_{x,y}(x_d, y_d, z_d)) \tag{4.5}$$

and we can now write concisely

$$b_{x,y}^{(a)}(t) = \sum_{d=1}^{D} \alpha_d \cdot h_{x,y}(x_d, y_d, z_d, t). \tag{4.6}$$

Next, we transform the continuous model into a discrete one. This consists of several steps. First, we naturally have to assume a discrete and finite set of observation locations. Second, we discretize the received signals by means of Nyquist rate sampling. Note that this does not mean we actually need to have access to this sampled observation, but instead we use it as a discrete and finite, hence convenient, representation of the continuous signal. As outlined in Section 2.1.1 it allows us to write linear transforms on the signal as matrices. Finally, we also make the same assumptions about the reconstructed signal. It is composed of defects residing on the same grid as we used for the observations. These regularity assumptions about the grid are necessary for efficient recovery to be possible as we see later on.

To define the observation grid we take synthetic aperture measurements at positions $(x, y, z = 0) \in \mathbb{R}^3$ located on the surface of the specimen lying on an equidistant grid defined as

$$\begin{aligned} G_{2D} = \{(x,y) | x = n_x \cdot \Delta x, n_x \in \{0, \ldots, N_x - 1\}, \\ y = n_y \cdot \Delta y, n_y \in \{0, \ldots, N_y - 1\}\}, \end{aligned}$$

where $\Delta x = \Delta y$ is the grid spacing and $N_x \in \mathbb{N}$ and $N_y \in \mathbb{N}$ are the number of samples in each spatial dimension.

After discretizing the observations $b_{x,y}^{(a)} : \mathbb{R} \to \mathbb{C}$ with a sampling rate $t_s = f_s^{-1}$ to vectors $\boldsymbol{b}_{x,y} \in \mathbb{C}^{N_t}$, resulting in N_t samples, the full set of $N_x \times N_y$ measurements can be combined into a single vector $\boldsymbol{b} \in \mathbb{C}^{N_x N_y N_t}$.

The same discretization process as for the observation and reconstruction locations as well as the sampling of the functions along time can be

applied to the atomic functions $h_{x,y}$ and we define a matrix

$$\boldsymbol{H}_{i,j} = h_{n_{x,i}\Delta x, n_{y,i}\Delta y}(n_{x,j}\Delta x, n_{y,j}\Delta y, n_{z,j}\Delta z, n_{t,i}t_s), \tag{4.7}$$

where $i = n_{t,i} \cdot N_x \cdot N_y + n_{y,i} \cdot N_x + n_{x,i}$ and $j = n_{z,j}N_xN_y + n_{y,j}N_x + n_{x,j}$ realize the same vectorization of $h_{x,y}$ as with $b^{(a)}_{x,y}$. In other words, the column $\boldsymbol{H}_{\cdot,j}$ contains the vectorized and discretized volumetric observation of a single reflector at (vectorized) position j. This is expressed in discrete time domain concisely via

$$\boldsymbol{b} = \boldsymbol{H}\boldsymbol{a} \tag{4.8}$$

and as such it is the basis for the following introduction of the sampling scheme. Note that compared to (4.6) the vector $\boldsymbol{a} \in \mathbb{C}^N$ now contains D non-zero elements with values α_d at unknown positions.

4.1.2.2 Data Acquisition

The vector $\boldsymbol{b} \in \mathbb{C}^N$ contains the discrete time samples of A-scans from all different measurement positions stacked on top of each other. However, we wish to consider compressed observations of the A-scans. So, instead of measuring $\boldsymbol{b}$ directly, we employ a compression step and measure a subsampled version of it. Following (2.1) in Section 2.1.2 the compressed measurement is given by

$$\boldsymbol{y} = \boldsymbol{\Phi} \cdot \boldsymbol{b} + \boldsymbol{n} = \boldsymbol{\Phi}\boldsymbol{H}\boldsymbol{a} + \boldsymbol{n}. \tag{4.9}$$

Here, $\boldsymbol{n} \in \mathbb{C}^N$ represents measurement noise with currently unspecified distribution. If we consider a single A-scan we consequently have

$$\boldsymbol{y}_{n_x,n_y} = \boldsymbol{\Phi}_{n_x,n_y}\boldsymbol{b}_{n_x,n_y} + \boldsymbol{n}_{n_x,n_y}.$$

For the compression matrix we assume it has the structure

$$\boldsymbol{\Phi}_{n_x,n_y} = \boldsymbol{S}_{n_x,n_y}\boldsymbol{F}_{N_t}\boldsymbol{\Sigma}_{n_x,n_y}$$

where $\boldsymbol{F}_{N_t}$ is the DFT matrix, $\boldsymbol{S}_{n_x,n_y} \in \mathbb{R}^{n_f \times N_t}$ is a selection matrix selecting n_f out of the N_t Fourier coefficients, so $\boldsymbol{S}$ is a row-subselected identity matrix. Moreover, $\boldsymbol{\Sigma}_{n_x,n_y}$ is a full rank diagonal matrix, a so called mixing matrix as outlined in [40], [41] and already presented in Section 2.4.2.1. Summarizing, the compression consists of first mixing followed by a DFT and then finally subselecting a few of the thereby computed Fourier coefficients. This merely depicts the mathematical model and we deal with possible hardware implementations in Section 4.1.6.

Considering the full vectorized, discretized and compressed observation, we can write

$$\boldsymbol{y} = \mathbf{blkdiag}\{\boldsymbol{\Phi}_{1,1}, \ldots, \boldsymbol{\Phi}_{N_x,N_y}\} \cdot \boldsymbol{b} + \boldsymbol{n} \tag{4.10}$$

as a linear model for the compressed observations. Here **blkdiag** denotes the block diagonal operator, which aligns the matrices in its argument as blocks on the diagonal of a matrix containing zeros for all other entries.

Remark 4.1. We would like to stress the fact that the following three sections on the data acquisition, algorithms, and hardware implementations are independent on some parts of the specific model presented above. It would be straightforward to use a frequency dependent transducer characteristic g, adopt to a slightly altered geometric setup by defining a different time of flight τ or define different grids for the sampling and reconstruction positions.

However, the structure of $\boldsymbol{H}$ we derive in Section 4.1.4 depends on the sampling and reconstruction positions defined via G_{3D}. For different acquisition schemes one would have to study the altered properties of $\boldsymbol{H}$ in order to derive efficient reconstruction algorithms.

We assume that the pulse shape p follows a known model. Estimating it by a measured pulse shape (e.g. from a backwall echo) from a reference measurement is another valid option. Further, the choice $p(t) = \delta(t)$ leads to $\boldsymbol{H}$ becoming a discrete DAS-operator (cf. Section 4.1.4.1). Still, already the "simple" model of this section represents realistic measurement data reasonably well, as validated in Section 4.1.7.2.

The spectrum of any realistic $p(t)$ decays exponentially for $|f| \to \infty$. For that reason, the DFT is a valid proxy to model the Fourier coefficients, since no (practically relevant) aliasing can occur.

The matrix $\boldsymbol{\Sigma}_{n_x,n_y}$ is necessary for some of the later presented sampling strategies. In some sense it results in the vectors $\boldsymbol{\Sigma}_{n_x,n_y}\boldsymbol{b}_{n_x,n_y}$ entries being centered random variables, which allows proofing reconstruction results in this setting, which are based on concentration of measure results from probability, see Section 2.4.2.1 for further details. ∎

4.1.3 Fourier Subsampling

Based on the proposed data acquisition scheme, we can distinguish different subsampling strategies, which we will discuss in the following. Here, we first focus on the mathematical formulation and an analysis from the signal processing perspective. Hardware considerations are deferred to Section 4.1.6.

4.1.3.1 Strategies

Random Sampling

Classically, the theory of compressed sensing started out by deriving reconstruction results for random matrices [11] as outlined in Section 2.4.1.1. In the proposed Fourier subsampling approach this translates to the diagonal entries of $\mathbf{\Sigma}_i$ following an independent Rademacher distribution, so $\mathbf{\Sigma}_i = \operatorname{diag}\{\boldsymbol{\xi}_i\}$ for a Rademacher vector $\boldsymbol{\xi}_i \in \{-1, 1\}^{N_t}$. For a Rademacher vector, the entries are drawn independently from the uniform distribution on $\{-1, 1\}$.

This mixing matrix can be justified several ways. First, by invoking the results in Corollary 2.1, which show that the matrices $\mathbf{\Sigma}_{x,y}$ allow to derive that the matrix $\mathbf{\Phi}$ has favorable geometric properties. Second, in cases where the spectrum of the received signal cannot be determined beforehand reliably enough, the mixing converts the signal into a random sequence which uses the whole bandwidth such that each Fourier coefficient contains roughly the same information about the signal.

Adding to the randomness, each $\boldsymbol{S}_{n_x,n_y}$ selects from the uniform distribution on all subsets of $\{1, \dots, N_t\}$ with magnitude n_f. If we select this subsampling strategy, we indicate it with a subscript of the complete compression matrix via $\mathbf{\Phi}_{\text{rnd}}$.

Maximal Sampling

The random sampling approach neglects any prior knowledge one has about the inserted pulse and its spectrum. One way to improve this would be to consider $\hat{\boldsymbol{p}} \in \mathbb{C}^{N_t}$ as the DFT of the inserted pulse and to define

$$J_q = \underset{q,n}{\operatorname{argmax}} \{|\hat{\boldsymbol{p}}_n|\}$$

as the index set of $\hat{\boldsymbol{p}}$ that refers to the q entries with largest amplitude. Now we set $\mathbf{\Sigma}_{n_x,n_y} = \boldsymbol{I}$ for all $i \leqslant N_x \cdot N_y$ and each $\boldsymbol{S}_{n_x,n_y}$ such that it

subselects the entries in J_{n_f}. This subsampling strategy is denoted as $\mathbf{\Phi}_{\max}$.

Random Energy-based Sampling

The choices $\mathbf{\Phi}_{\text{rnd}}$ and $\mathbf{\Phi}_{\max}$ represent two very different approaches. The first focuses entirely on randomness and does not assume prior knowledge about the pulse, whereas the latter assumes perfect spectral knowledge about the pulse only and as such does not need to make use of random subsampling and can resort to a deterministic selection of the Fourier samples. However, in reality one mostly has approximate knowledge of the pulse. So the former approach would be too pessimistic and the latter too optimistic.

As a trade-off between the two strategies, we impose a random sampling based on the frequency power distribution of the inserted pulse. This is achieved by selecting a subset of Fourier coefficients by means of$\boldsymbol{S}_{n_x,n_y}$ in such a way that the probability that the ith frequency bin is contained in the subset is proportional to the power of the pulse in that frequency bin. Since we are drawing from all possible Fourier bins without replacement, this has to be an iterative process.

Let q_i for $i \leqslant N_t$ be the normalized power of the pulse in the i-th DFT-coefficient, so we have

$$q_i = |\hat{\boldsymbol{p}}_i| / \|\hat{\boldsymbol{p}}\|_1.$$

Now, we iteratively construct a set of indices J based on $\boldsymbol{q} \in \mathbb{R}^{N_t}$. Assume we have already drawn $n < n_f$ indices from the set $\{1, \ldots, N_t\}$ into the set J_n. Then we set $\boldsymbol{q}_{J_n} = 0$ and then normalize $\boldsymbol{q}$ such that the remaining entries sum up to 1. Then we draw another index j_{n+1} and set $J_{n+1} = J_n \cup \{j_{n+1}\}$ and iterate until $n = n_f$. We again set $\mathbf{\Sigma}_i = \boldsymbol{I}$ and denote the compression matrix selected according to this strategy with $\mathbf{\Phi}_{\text{nrg}}$.

Independent A-Scans

First, one notices that during the treatment of the individual A-scans it is possible to choose $\boldsymbol{S}_{n_x,n_y} = \boldsymbol{S}_0$ for all n_x, n_y. This implies that we collect the same Fourier coefficients for all A-scan positions. In case we do *not* keep them fixed, we add a subscript to the compression matrix as $\mathbf{\Phi}_{\ldots,f}$. The same distinction can be made for the choice of the mixing done by $\mathbf{\Sigma}_{n_x,n_y}$. If we vary them across A-scans we indicate this via $\mathbf{\Phi}_{\ldots,m}$. Note that $\mathbf{\Phi}_{\ldots,m,f}$ is also possible. In case of the same mixing and subsampling

		$\boldsymbol{S}_{n_x,n_y} = \boldsymbol{S}_0$	
		yes	no
$\boldsymbol{\Sigma}_{n_x,n_y} = \boldsymbol{\Sigma}_0$	yes	$\boldsymbol{\Phi}_{\text{max}}$, $\boldsymbol{\Phi}_{\text{nrg}}$, $\boldsymbol{\Phi}_{\text{rnd}}$	$\boldsymbol{\Phi}_{\text{nrg},f}$, $\boldsymbol{\Phi}_{\text{rnd},f}$
	no	$\boldsymbol{\Phi}_{\text{rnd},m}$	$\boldsymbol{\Phi}_{\text{rnd},m,f}$

Table 4.1 *Different aspects of the sampling strategies can be combined freely* – Sampling strategies considered in this work ■

for each A-scan, we can also write Equation (4.10) more concisely

$$\boldsymbol{\Phi} = (\boldsymbol{S}\boldsymbol{F}_{N_t}\boldsymbol{\Sigma}) \otimes \boldsymbol{I}_{N_x} \otimes \boldsymbol{I}_{N_y},$$

where $\otimes$ defines the Kronecker product, to describe the compression matrix. In Table 4.1 we summarized all possible and discussed combinations.

4.1.3.2 Analytic vs. Real Representations

Generally, we assume $\hat{\boldsymbol{p}}$ in case of maximal sampling and energy-based sampling to be the DFT of the analytic signal, hence only the positive half of the spectrum is nonzero and therefore $[\hat{\boldsymbol{p}}]_n = 0 \; \forall n > \lfloor N_t/2 \rfloor$. Under this assumption, the selection matrices $\boldsymbol{S}_i$ will only pick from the first $\lfloor N_t/2 \rfloor$ rows of $\boldsymbol{F}$. In [O8], we used a real-valued model for $\boldsymbol{H}$ and therefore symmetrized the sampling patterns by additionally choosing $N_t - J_{n_f}$ such that in total $2n_f$ Fourier coefficients are sampled. The symmetrized sub-sampling matrices are denoted via $\boldsymbol{\Phi}^{(\text{Re})}$.

4.1.3.3 Performance Guarantees for Random Subsampling

Now we study the ramifications of the proposed Fourier subsampling strategies. In fact, we are able to establish some analytic results for one of the presented approaches. In this chapter we focus on the case where $\boldsymbol{\Phi} = \boldsymbol{\Phi}_{\text{rnd}}$, so each A-scan is pre-multiplied with the same $\boldsymbol{\Sigma}$ and we pick the same Fourier coefficients by means of $\boldsymbol{S}$, so that

$$\boldsymbol{\Phi} = (\boldsymbol{S}\boldsymbol{F}_{N_t}\boldsymbol{\Sigma}) \otimes \boldsymbol{I}_{N_x} \otimes \boldsymbol{I}_{N_y}.$$

As we have seen in Section 2.2.2, the recovery performance of sparsity exploiting algorithms and the employed compression strategy can be measured in terms of the RIP, if a signal $\boldsymbol{b}$ is sparse in an orthonormal basis $\boldsymbol{H}$, which means that in $\boldsymbol{b} = \boldsymbol{H}\boldsymbol{a}$ the vector $\boldsymbol{a}$ is sparse. If on the other hand

$\boldsymbol{a}$ is sparse, but $\boldsymbol{H}$ is not a basis anymore, but an overcomplete dictionary as in our case, then the so-called $\boldsymbol{A}$-RIP introduced in Section 2.2.5 yields the natural framework for reconstruction guarantees, which we make use of in the following by considering the $\boldsymbol{A}$-RIP for $\boldsymbol{A} = \boldsymbol{H}$.

If we consider a single A-scan vector $\boldsymbol{b}_{n_x,n_y}$ recorded at an arbitrary but fixed measurement position (x, y) the pulse-echo model in Equation (4.2) results in $\boldsymbol{b}_{n_x,n_y}$ being a linear superposition of shifted versions of the inserted pulse p. Hence, it follows that $\boldsymbol{b}_{n_x,n_y} = \boldsymbol{G}\boldsymbol{\alpha}_{n_x,n_y}$, where the columns of $\boldsymbol{G}$ are the discretized and shifted versions of the inserted pulse, rendering $\boldsymbol{G} = \boldsymbol{\Gamma}(\boldsymbol{p})$ a circulant matrix (see Section 3.3.2.1), when $\boldsymbol{p}$ is the sampled pulse waveform. In other words: the matrix $\boldsymbol{G}$ is the sparsifying dictionary for individual A-scans. Now, according to our compression scheme, we have $\boldsymbol{y}_{n_x,n_y} = \boldsymbol{S}\boldsymbol{F}_{N_t}\boldsymbol{\Sigma}\boldsymbol{G}\boldsymbol{\alpha}_{n_x,n_y}$ for our compressed observations at a single scanning position x, y.

Since the total number of defects in the specimen is assumed to be small, i.e. $\boldsymbol{a}$ in Equation (4.2) being D-sparse, we have that each $\boldsymbol{\alpha}_{n_x,n_y}$ is sparse as well. This implies that each $\boldsymbol{b}_{n_x,n_y}$ is sparse in the dictionary $\boldsymbol{G}$. Let now $d_{\max} = \max_{n_x,n_y} \|\boldsymbol{\alpha}_{n_x,n_y}\|_0$ be the maximum encountered sparsity-level over all A-scans. However, the sparsity is not prevalent with respect to a basis but to the overcomplete dictionary $\boldsymbol{G}$. This sparks the need for a modified reconstruction guarantee presented in the next definition.

The following result calculates the $\boldsymbol{H}$-RIC of the matrix $\boldsymbol{A} = \boldsymbol{C} \otimes \boldsymbol{I}_{N_x N_y}$ for an arbitrary Kronecker factor $\boldsymbol{C}$ and as such it delivers necessary conditions for efficient, stable and robust recovery to happen when compressing signals with $\boldsymbol{C} \otimes \boldsymbol{I}_{N_x N_y}$ and when the signals are sparse with respect to the dictionary $\boldsymbol{H}$ in (4.7).

Theorem 4.1. *For each $k \in \mathbb{N}$ it holds that the $\boldsymbol{G}$-RIC $\delta_k^{\boldsymbol{G}}$ of the matrix $\boldsymbol{A}$ and the $\boldsymbol{H}$-RIC of $\boldsymbol{C} = \boldsymbol{A} \otimes \boldsymbol{I}_{N_x N_y}$ are equal.* ∎

Proof. From the definition of $\delta_k^{\boldsymbol{G}}$ and the vectors $\boldsymbol{\alpha}_{n_x,n_y}$ we have for every (n_x, n_y) that

$$(1 - \delta_{d_{\max}}^{\boldsymbol{G}}) \left\| \boldsymbol{\alpha}_{n_x,n_y} \right\|_2^2 \leqslant \left\| \boldsymbol{A}\boldsymbol{\alpha}_{n_x,n_y} \right\|_2^2 \leqslant (1 + \delta_{d_{\max}}^{\boldsymbol{G}}) \left\| \boldsymbol{\alpha}_{n_x,n_y} \right\|_2^2.$$

Together with the definition of $\boldsymbol{G}$ and $\boldsymbol{y}_{n_x,n_y}$ and the properties of the Kronecker product the statement directly follows. ∎

The theorem above illuminates how, given the proposed sensing scenario, only the properties of the A-scans and their sparse representation influence the recovery performance. Additionally, only the most complex

A-scan in the sense that it is the least sparse one determines the worst case performance. Now, since randomly sub-selected Fourier matrices are known to have a low $\boldsymbol{G}$-RIC (as outlined in Corollary 2.1), Theorem 4.1 shows how to asses the recovery performance and infer the number of necessary measurements. Moreover, one is able to determine the number of Fourier coefficients n_f such that the $\boldsymbol{G}$-RIP of the appropriate order holds with a high probability, which delivers stable, robust and efficient recovery if we employ ℓ_1-minimization as given in Equation (2.14). This we already formalized in Theorem 2.6, where it is shown that $\delta^{\boldsymbol{G}}_{d_{\max}} < 0.08$ is a sufficient condition for stable recovery to happen in every A-scan. Combining the requirements in Theorem 2.6 with the results in Section 2.4.2.2 we can finally state that n_f has to satisfy

$$n_f \geqslant \hat{C} \log N_t d_{\max} \log(d_{\max}) \log^2(50 d_{\max}). \tag{4.11}$$

We would like to stress the fact that in the above analysis we derived a performance bound for the 3D reconstruction process, while only dealing with the restricted isometry constants associated to the dictionary of the single dimensional A-scans. In other words, since we are only compressing / subsampling in one dimension, this dimension alone determines the reconstruction performance. Further, the choice of n_f in a practical setup can be based on a worst-case number of defect echoes that are expected to appear in a single A-scan.

Remark 4.2. On average $\boldsymbol{\Phi}_{\text{rnd}}$ selects around n_f / N_t parts of the energy that is contained in the reflected waves. As such of the strategies considered in this work, it has the worst SNR. However due to the random mixing done by the $\boldsymbol{\Sigma}_{n_x,n_y}$ it maximizes the bandwidth of the measurement, since it is approximately the same as the inserted pulse.

The strategy using $\boldsymbol{\Phi}_{\max}$ maximizes the SNR, since it collects the most energy from the pulse by sampling at the around the peaks in the spectrum. However, due to the typical shape of an ultrasound pulse in time- and frequency domain, the resulting samples are closely spaced, which directly results in a low bandwidth of the acquired signal.

As we see in Section 4.1.7, the two strategies discussed above perform as expected and a trade-off between the two is represented by $\boldsymbol{\Phi}_{\text{nrg}}$. So depending on the goal during reconstruction in terms of depth-resolution, the different strategies cover the whole range from high SNR and poor bandwidth to poor SNR and high bandwidth.

The problem of estimating the model order $d_{\max}$ to set n_f before carrying out the reconstruction is hard to overcome in a CS setting. There is a large literature on sparsity order estimation with various advantages and

drawbacks, see [O1], [53], [54], [58] and we outline one specific approach in Section 2.5. Often a satisfactory method for model order selection depends very much on the specific applications' side constraints. In our case, for instance, on the size of a typical defect, the number of defects and their shape.

Note for the recovery guarantees that they only represent scaling laws and not explicit bounds on the number of measurements due to the fact that there is a (here) unspecified constant factor involved as we already indicated for instance in Figures 2.1 and 2.2. However, even if one would compute it explicitly using the results in [40] and [41] these estimates would be too conservative for practical considerations. In case when one has empirical evidence that for a certain sparsity order $d_{\max}$, problem size N and number of measurements n_f the recovery is satisfactory, one can infer the necessary number of measurements if the problem size or sparsity change. ∎

4.1.4 Reconstruction

Due to the more involved sampling process compared to traditional Nyquist sampling we have already argued in Section 2.2 that the reconstruction step is similarly more complex and especially non-linear. Due to the large problem size usually encountered in US-NDT we have to take several additional measures to render the reconstruction stage an efficient process.

4.1.4.1 Algorithms

Now that we have established the data model and studied the compression schemes, it first is necessary to formulate viable algorithms in order to reconstruct $\boldsymbol{a}$ in Equation (4.9) for all discussed compression strategies. Due to the size of $\boldsymbol{\Phi H}$, we resort to the matrix-free representations as introduced in Chapter 3 in the form of $\phi_{\boldsymbol{\Phi H}}$ and $\beta_{\boldsymbol{\Phi H}}$. In the following we present three different approaches for the imaging process based on the compressed data.

Synthetic Aperture Focusing Technique

As a first simple strategy we apply the backward projection to the observation data which yields the estimate $\boldsymbol{a}_{\text{SAFT}} = \beta_{\boldsymbol{\Phi H}}(\boldsymbol{y})$ which is traditionally called SAFT when $\boldsymbol{\Phi}$ is the identity matrix and $h(t) = \delta(t)$ (cf. for example [128]). However, our approach in formulating the data model and compression scheme directly in terms of matrix vector products yields a more general implementation that can also cope with compressed data and more involved physically motivated forward models. As another special case, our formulation also yields a (compressed) synthetic aperture formulation of the *excitelet* reconstruction [103].

Sparse Signal Recovery

Since SAFT does not exploit sparsity, we will see in Section 4.1.7 that the imaging can substantially be improved by sparsity aware algorithms. In Section 2.2 we essentially presented two strategies for estimating sparse vectors from compressed measurements. If we reformulate (2.12) and (2.14) to our problem at hand, we get

$$\min_{\boldsymbol{a}\in\mathbb{C}^M} \|\boldsymbol{a}\|_0 \quad \text{subject to} \quad \|\boldsymbol{\Phi H a} - \boldsymbol{b}\|_2 \leqslant \eta,$$

which can be solved by means of Algorithm 3.1 using the matrix-free representation of $\boldsymbol{\Phi H}$ and

$$\min_{\boldsymbol{a}\in\mathbb{C}^M} \|\boldsymbol{\Phi H a} - \boldsymbol{b}\|_2^2 + \lambda(\eta)\|\boldsymbol{a}\|_1,$$

which can efficiently solved by Algorithm 3.2 respectively also resorting to the matrix-free representation.

In the following we will focus on FISTA as a representative example for many so-called proximal gradient algorithms used in compressed sensing, since it provides a good trade-off between simplicity and performance. However, we would like to stress that in general the following considerations provide a blueprint on how to apply matrix-free reconstruction algorithms that use forward and backward projections to the problem at hand.

4.1.4.2 Fast Transforms

To devise efficient implementations for $\phi_{\boldsymbol{\Phi H}}$ and $\beta_{\boldsymbol{\Phi H}}$ it is necessary to consider the structure of the matrices $\boldsymbol{\Phi}$ and $\boldsymbol{H}$ in order to make the software presented in Chapter 3. We will treat both matrices separately and then make use of Section 3.3.1 to get an efficient matrix-free representation for the complete system.

As it turns out, we can show that $\boldsymbol{H}$ in Equation (4.9) is a block Toeplitz matrix as defined in Section 3.3.2.5. The key observation is that moving any scatterer in the x-y-plane while also displacing the measurement position by the same amount, does not change the data captured by the transducer up to some boundary effects. This is formalized in the next result.

Theorem 4.2. *The matrix* $\boldsymbol{H} \in \mathbb{R}^{N_z N_x N_y \times N_t N_x N_y}$ *from Equation (4.9) is block 2-level Toeplitz, where for the generating elements* $\boldsymbol{h}$ *it holds that* $\boldsymbol{h} \in \mathbb{R}^{N_z \times N_t \times 2N_x - 1 \times 2N_y - 1}$. ∎

Proof. We consider a column $\boldsymbol{H}_{z,x_1,y_1}$ of $\boldsymbol{H}$ with $(x_1, y_1, z) \in G_{3D}$. If we now pick an arbitrary $(x_2, y_2, ct) \in G_{3D}$, we see that

$$[\boldsymbol{H}_{z,x_1,y_1}]_{t,x_2,y_2} = \gamma(z, t, x_1 - x_2, y_1 - y_2),$$

for some function γ depending on the transducer characteristic g and the time of flight τ. And due to the specific structure of τ we have that $\tau_{x,y}(x_d, y_d, z) = \tau_{0,0}(x_d - x, y_d - y, z)$, so $\boldsymbol{H}$ is 2-level Toeplitz because of the translational invariance with respect to x, x_d and y, y_d. Finally, the asserted structures of $\boldsymbol{H}$ and $\boldsymbol{h}$ follow easily. ∎

Note that above result implicitly also states how the defining array $\boldsymbol{h} \in \mathbb{C}^{N_t \times N_z \times 2N_x - 1 \times 2N_y - 1}$ to generate the block 2-level Toeplitz Matrix $\boldsymbol{H}$ should be computed. Now, by accounting for the block Toeplitz structure and explicitly calculating the influence of the 2-level Toeplitz blocks in Equation (3.15), we get $\phi_{\boldsymbol{H}}$ and in a similar fashion $\beta_{\boldsymbol{H}}$ by noticing that

$$\phi_{\boldsymbol{H}}(\boldsymbol{a}) = \boldsymbol{H} \cdot \boldsymbol{a} = \left[\sum_{j=1}^{N_t} \boldsymbol{H}_{i,j} \boldsymbol{a}_j \right]_{i=1}^{N_z} = \left[\sum_{j=1}^{N_t} \phi_{\boldsymbol{H}_{i,j}}(\boldsymbol{a}_j) \right]_{i=1}^{N_z}, \tag{4.12}$$

where the $\boldsymbol{H}_{i,j}$ are 2-level Toeplitz and each $\boldsymbol{a}_j$ is a subvector of $\boldsymbol{a}$ of size $N_x N_y$.

However, consider the specific example from Section 4.1.1.2 where we take measurements on a scanning grid of size 100×100 and measurement

Data: Input data $\boldsymbol{x} \in \mathbb{C}^{N_z \times N_x \times N_y}$
Result: Transformed data $\phi_{\boldsymbol{\Phi H}}(\boldsymbol{x}) = \boldsymbol{y} \in \mathbb{C}^{n_f \times N_x \times N_y}$
[1] Zeropad $\boldsymbol{x}$ to $\boldsymbol{x}_0 \in \mathbb{C}^{N_z \times 2N_x-1 \times 2N_y-1}$;
[2] Apply a 2D-FFT along dimensions 2 and 3 to $\boldsymbol{x}_0$ to get $\hat{\boldsymbol{x}}_0$;
[3] Set $\hat{\boldsymbol{y}}_0 = \boldsymbol{0} \in \mathbb{C}^{N_z \times 2N_x-1 \times 2N_y-1}$;
[4] **for** $i_t = 1, \ldots, N_t$ **do**
[5] | **for** $i_z = 1, \ldots, N_z$ **do**
[6] | | Calculate $\boldsymbol{h}_{i_t,i_z} \in \mathbb{C}^{2N_x-1 \times 2N_y-1}$;
[7] | | Apply a 2D-FFT to $\boldsymbol{h}_{i_t,i_z}$ to get $\hat{\boldsymbol{h}}_{i_t,i_z}$;
[8] | | $\hat{\boldsymbol{y}}_{0,i_t} \mathrel{+}= \hat{\boldsymbol{h}}_{i_t,i_z} \odot \hat{\boldsymbol{x}}_{0,i_t}$;
[9] | **end**
[10] **end**
[11] Apply a 2D-iFFT to $\hat{\boldsymbol{y}}_0$ along dimensions 2 and 3 to get $\boldsymbol{y}_0$;
[12] Revert the zero-padding on $\boldsymbol{y}_0$ to get $\boldsymbol{y} \in \mathbb{C}^{N_z \times N_x \times N_y}$;
[13] **for** $i_x = 1, \ldots, N_x$ **do**
[14] | **for** $i_y = 1, \ldots, N_y$ **do**
[15] | | $\boldsymbol{y}_{\cdot,i_x,i_y} \leftarrow \boldsymbol{\xi}_{i_x,i_y} \odot \boldsymbol{y}_{\cdot,i_x,i_y}$;
[16] | **end**
[17] **end**
[18] Apply a 1D-FFT along the first dimension to get $\hat{\boldsymbol{y}} \in \mathbb{C}^{N_z \times N_x \times N_y}$;
[19] **for** $i_x = 1, \ldots, N_x$ **do**
[20] | **for** $i_y = 1, \ldots, N_y$ **do**
[21] | | $\boldsymbol{y}_{\cdot,i_x,i_y} \leftarrow \boldsymbol{S}_{i_x,i_y} \hat{\boldsymbol{y}}_{\cdot,i_x,i_y}$;
[22] | **end**
[23] **end**
[24] Return $\boldsymbol{y}$.

Algorithm 4.1: *The computation of $\phi_{\boldsymbol{\Phi H}}$ can be more memory efficient at the cost of more computations* – An explicit version Equation (3.15), which also does on-the-fly calculations of the generating elements of $\boldsymbol{H}$.

has 1000 time samples. Then, assuming that the reconstruction grid is identical to the measurement grid, the 1000^2 generating elements each have size 199×199, which would require $\approx 158\,\text{GB}$. So, even for moderately sized problems and while exploiting the inherent Toeplitz structure, we usually cannot fit the generating sequences of each $\boldsymbol{H}_{i,j}$ into memory at once. In these cases the generating elements have to be recomputed during each transformation step. To this end, we propose to recalculate each $\boldsymbol{h}_{i,j}$ on the fly only when needed in (4.12) during the multiplication of $\boldsymbol{H}$. This modified version of Equation (3.15) is given in Algorithm 4.1. The hereby reduced memory footprint at the cost of more computations is

especially beneficial when Algorithm 4.1 is realized on a GPU as depicted by Listing 4.1.

Next, we proceed with the analysis of the structure of the compression matrix $\boldsymbol{\Phi}$ in order to devise $\phi_{\boldsymbol{\Phi}}$ and $\beta_{\boldsymbol{\Phi}}$. In its most general form we have by Equation (4.10) that

$$\boldsymbol{\Phi} = \mathbf{blkdiag}\{\boldsymbol{\Phi}_{1,1}, \ldots, \boldsymbol{\Phi}_{N_x,N_y}\},$$

i.e., the matrix $\boldsymbol{\Phi}$ is a block-diagonal matrix, where each block consists of a product

$$\boldsymbol{S}_{n_x,n_y} \boldsymbol{F}_{N_t} \mathbf{diag}(\boldsymbol{\xi}_{n_x,n_y}).$$

In terms of matrix vector products, this means that multiplication with $\boldsymbol{\Phi}$ is a blockwise procedure, where each block is processed first by a pointwise multiplication ($\odot$) with $\boldsymbol{\xi}_i$, an DFT and lastly a subselection of the vector in the frequency domain. Thus, it is trivial to implement these matrix vector products efficiently. Finally, we invoke (3.6) to get $\phi_{\boldsymbol{\Phi H}} = \phi_{\boldsymbol{\Phi}} \circ \phi_{\boldsymbol{H}}$ and $\beta_{\boldsymbol{\Phi H}} = \beta_{\boldsymbol{H}} \circ \beta_{\boldsymbol{\Phi}}$, where $\circ$ denotes function concatenation. So given two distinct implementations for the matrix vector products for $\boldsymbol{\Phi}$ and $\boldsymbol{H}$ we only have to concatenate them by applying them one after another.

Algorithm 4.1 schematically displays how to carry out the multiplication with $\boldsymbol{\Phi H}$ efficiently. The algorithm for $(\boldsymbol{\Phi H})^{\mathrm{H}}$ can be derived in a similar manner.

Remark 4.3. It is worth noting that Algorithm 4.1 should be implemented in a blocked manner. This means that the loops for i_x and i_y should be replaced with loops over blocks containing several $i_{x_i}, \ldots i_{x_i+b}$ and $i_{y_i}, \ldots i_{y_i+b}$ at once, since most high-level programming languages allow for faster processing of these blocks, especially when working on a GPU. This also steers the amount of system memory the transformation occupies and can be tuned to the system specifications at hand. In the extreme case when one has enough system memory available to store $\boldsymbol{h}$ entirely, one should do so in order to maximize performance.

Additionally, the problem dimensions N_x and N_y do not influence the FFT performance that much, since for badly conditioned FFT sizes in terms of prime factors, once can exploit Bluestein's algorithm [129], which expresses the Fourier transform as a cyclic convolution where one can use zero-padding. Most modern FFT implementations have heuristics in place to decide whether to use this alternative approach or not.

Finally, we refer to Listing 4.1 that outlines how to use the `fastmat` API to move the fast transforms to the GPU. In this snippet we use

```
import fastmat as fm
import cupy as cp
import numpy as np

class FourierGPU(fm.Matrix):
  def _forwardGPU(x: cp.ndarray) -> cp.ndarray:
    # this does the FFT on the GPU by using CuPy
    return cp.fft.fft(x)

  def forward(x: np.ndarray) -> np.ndarray:
    #  copy to the GPU
    x = cp.array(x)

    # call the GPU routine
    y = self._forwardGPU(x)

    # copy back to CPU and return
    return y.get()
```

Listing 4.1 *fastmat classes can easily wrap GPU code* – How to move the DFT GPU as an isolated calculation. ■

CuPy [90] that supplies an array data structure for GPUs that is exposed to regular Python code. The essential idea is to copy the input vector x to the GPU, where the transform is carried out and the result is copied back to the CPU memory. This way the GPU calculations are isolated to the transform only and the rest of the computing path is not affected. Naturally, this introduces a substantial overhead if the memory transfer has to happen too often. But, as Figure 4.5 and Figure 4.6 in Section 4.1.6.2 indicate, as soon as the transform size is large enough, the overhead can be compensated for by the substantially decreased runtime. ■

4.1.4.3 The Largest Singular Value

As we see in Algorithm 3.2 it is necessary to compute or at least estimate $\sigma_{\max}$ for the matrix $\boldsymbol{\Phi H}$. When estimating the singular value, one should take care that it is not underestimated, because in this case one expects the objective function minimized by FISTA to be smoother than it actually is. This in turn leads to more aggressive (and in this case unjustified so) iteration steps, which ultimately result in divergence of the algorithm.

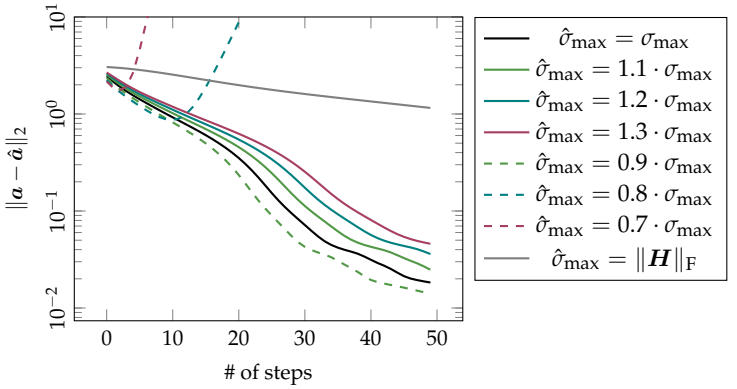


Figure 4.2 *Wrong estimates of $\sigma_{\max}$ can have detrimental on the estimate and lead to divergence* – Reconstruction error vs. number of FISTA steps using different stepsizes. ■

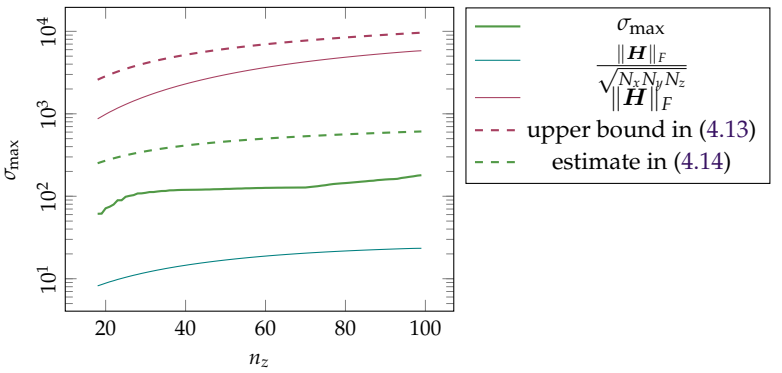
Consequently it is better to overestimate $\sigma_{\max}$ to avoid divergence at the cost of slower convergence.

To illustrate the influence of the stepsize, we carry out a numerical simulation where a Toeplitz matrix $\boldsymbol{H}$ and a sparse vector $\boldsymbol{a}$ are generated randomly, FISTA is run to recover $\boldsymbol{a}$ and we study the effect of over/underestimating $\sigma_{\max}$ of $\boldsymbol{H}$. In particular, we chose $\boldsymbol{H} \in \mathbb{R}^{N \times N}$, $N = 100$, and the sparse vector $\boldsymbol{a}$ to have 10% of its entries non-zero, with random positions for the non-zero elements in $\boldsymbol{a}$ and random amplitudes. Figure 4.2 shows the reconstruction error of FISTA as given in Algorithm 3.2 with $\lambda = 1$ over 50 steps, where we over- and underestimate the largest singular value of $\boldsymbol{H}$ by 10%, 20% and 30%, respectively. Each stepsize configuration is averaged over 100 trials. The reconstruction error is measured as $\|\boldsymbol{a} - \hat{\boldsymbol{a}}\|_2$, where $\hat{\boldsymbol{a}}$ is the estimate of FISTA at each step.

It can be seen that already slightly underestimating $\sigma_{\max}$, therefore, chosing the stepsize too large, leads to divergence and a bad reconstruction. This behavior can be weakened by increasing λ, i.e., enforcing a sparser solution. On the other hand, FISTA converges slower for stepsizes smaller than the optimum.

Figure 4.3 shows the largest singular value of $\boldsymbol{H}$ and the corresponding upper and lower bound for it computed for varying N_z as well as the approximation defined in (4.14) (dashed red line). The approximation is shown as a dashed green line in Figure 4.3.

However, in a matrix-free setting this is no trivial endeavor, since



c_0	f_s	Δx	Δy	α
5920 m/s	14 MHz	0.50 mm	0.50 mm	60°
Δz	N_x	N_y	N_z	f_c
422.85 µm	25	25	n_z	4 MHz

Figure 4.3 *The geometric mean seems to be a valid proxy for* $\sigma_{\max}$ – Largest singular value vs. N_z for $\boldsymbol{H}$ computed using the parameters in the table ■

algorithms that rely on having the whole matrix in system memory cannot be used.

Backtracking

A first approach would be to use a simple backtracking version of FISTA [30, p. 194], which does not need the singular value as an input, but rather a crude lower bound σ_{low} together with a scaling parameter $\eta > 1$. Then, in each step one determines an appropriate stepsize based on a local estimate σ_{loc} within

$$\sigma_{\text{low}} \leqslant \sigma_{\text{loc}} \leqslant \eta \cdot \sigma_{\max}.$$

Ultimately, η influences the speed of convergence such that higher values of η result in slower convergence due to the overestimation of the singular value. Additionally, this comes at the cost of evaluating the objective function and a suitable quadratic approximation numerous times during the iteration, leading to an impractical amount of computations that are necessary for convergence, since we have to call $\phi_{\boldsymbol{\Phi H}}$ twice per backtracking step. Hence, in large scale scenarios it might be advantageous to have means for acquiring the largest singular value. In the following we present two alternatives.

Estimation

Another approach would be to find a suitable estimate $\hat{\sigma}_{\max}$ that is cheap to compute and which still allows proper convergence [O9]. In this case one has to ensure that the estimated singular value is bounded from below by the true $\sigma_{\max}$. For the product $\boldsymbol{\Phi H}$ at hand we proceed as follows.

A simple bound for the largest singular value is given by

$$\frac{1}{\sqrt{N}}\|\boldsymbol{\Phi H}\|_F \leqslant \sigma_{\max} \leqslant \|\boldsymbol{\Phi H}\|_F, \tag{4.13}$$

which would imply to estimate $\hat{\sigma}_{\max} = \|\boldsymbol{\Phi H}\|_F$, which in terms of convergence provides poor results, since it forces FISTA to take too conservative steps. As such, we propose to use the geometric mean of the upper and lower bound

$$\hat{\sigma}_{\max} = \frac{\|\boldsymbol{\Phi H}\|_F}{N^{(1/4)}}, \tag{4.14}$$

for which it is necessary to compute $\|\boldsymbol{\Phi H}\|_F$. However, since we have no direct access to the entries of $\boldsymbol{\Phi H}$ we need to estimate this as well. To this end, we first notice that

$$\|\boldsymbol{\Phi H}\|_F^2 = \sum_{z_d}\sum_{x_d}\sum_{y_d}\left\|\boldsymbol{\Phi H}_{z_d,x_d,y_d}\right\|_2^2.$$

this means we are summing over the squared norms of all atoms, where each atom belongs to a single defect position (x_d, y_d, z_d). Now, neglecting boundary effects within the individual atoms we can simplify to

$$\|\boldsymbol{\Phi H}\|_F^2 \approx N_x N_y \sum_{z_d}\left\|\boldsymbol{\Phi H}_{x_d^0,y_d^0,z_d}\right\|_2^2$$

for some representative target position $(x_d^0, y_d^0, \cdot)$. This leaves us with calculating the innermost summand for which we have

$$\left\|\boldsymbol{\Phi h}(x_d^0, y_d^0, z_d)\right\|_2^2 \approx \sum_{n_x=1}^{N_x}\sum_{n_x=1}^{N_x} g_{n_x\Delta x, n_y\Delta y}(x_d^0, y_d^0, z_d)^2 \left\|\boldsymbol{\Phi}_{n_x,n_y}\boldsymbol{h}_{n_x,n_y}(x_d^0, y_d^0, z_d)\right\|^2.$$

Finally, depending on the actual subselection strategy employed by $\boldsymbol{\Phi}$ we can approximate the largest singular value by the above estimate of the Frobenius norm.

Iterative Approximation

To circumvent possible problems from an inaccurate estimate of the singular value, iterative algorithms [73] to approximate it have been developed and finally implemented in ARPACK [130]. These again only rely on $\phi_{\boldsymbol{\Phi H}}$ and $\beta_{\boldsymbol{\Phi H}}$ and for a specific scenario $\boldsymbol{\Phi H}$ one can cache this approximate result. However one cannot guarantee that the singular value is not under-estimated in magnitude.

Remark 4.4. In cases where the evaluation of $\phi_{\boldsymbol{\Phi H}}$ takes up to several minutes it is not advisable to use a backtracking scheme in FISTA. Instead one should use these additional projections during an Arnoldi iteration and approximate and store the largest singular value directly. Although one cannot guarantee that ARPACK estimates $\sigma_{\text{est}} > \sigma_{\max}$ it provides a tolerance $\delta_\sigma \geqslant |\sigma_{\text{est}} - \sigma_{\max}|$. So ultimately one can ensure that $\sigma_{\max} \leqslant \sigma_{\text{est}} + \delta_\sigma$, which can in turn be used in FISTA safely.

Additionally, the approach we took for the estimation of $\sigma_{\max}$ using Equation (4.14) can easily be generalized and altered to account for different data and compression models. ∎

4.1.5 Asymptotic performance

As one of the main questions in NDT is how accurately we can localize a defect within a specimen, we investigate this accuracy in terms of the CRB from Section 2.4.4 for a single scatterer. In particular, we investigate the loss in localization accuracy as a function of the number of Fourier coefficients per A-scan n_f in order to compare the proposed sub-sampling strategies. To this end, we now have to shift our perspective from (4.10), which has consumed the parameters of interest into a matrix-vector product, we go back to the original parametric model in (4.2), since this formulation allows us to directly model the influence of certain parameters on the observed measurements.

The CRB as defined in Section 2.4.4 is computed by means of the FIM, which in turn depends on our assumed noise statistics, which we specify now. Let $\boldsymbol{n}$ in Equation (4.9) be zero-mean circularly symmetric white complex Gaussian noise with variance σ^2. Then, it follows that $\boldsymbol{y} \sim \mathcal{CN}(\boldsymbol{\Phi b}, \sigma^2 \boldsymbol{I})$. The measurements depend on the set of parameters

$$\boldsymbol{u} = [\boldsymbol{p}^{\mathrm{T}}, \alpha, \varphi, \sigma_n^2]^{\mathrm{T}} \in \mathbb{R}^6$$

where $\boldsymbol{p} = [x_1, y_1, z_1]^{\mathrm{T}} \in \mathbb{R}^3$ comprises the scatterer location, α is the scatterer amplitude and φ is the scatterer phase, such that in Equation (4.2)

we would get $\alpha_1 = \alpha e^{\jmath\varphi}$. We further assume a Gaussian pulse with

$$p(t) = \exp\left(-\xi^2(t-\tau_{x,y})^2 + \jmath\omega_c(t-\tau_{x,y})\right),$$

and a transducer directivity of the form

$$g(x_d, y_d, z_d) = \exp\left(-\frac{(x_d-x)^2+(y_d-y)^2}{\tan^2(\theta) z_d^4}\right),$$

where we assume that the center wave number ω_c, bandwidth ξ and opening angle θ are known and hence do not have to be estimated. Since $\boldsymbol{y}$ follows a Gaussian distribution and together with the above specifications of $\boldsymbol{u}$, p and g, we specify (2.25) to our scenario and get

$$\boldsymbol{J}(\boldsymbol{u}) = \frac{2}{\sigma_n^2}\Re\left\{\left(\frac{\partial \boldsymbol{b}}{\partial \boldsymbol{u}^{\mathrm{T}}}\right)^{\mathrm{H}} \boldsymbol{\Phi}^{\mathrm{H}}\boldsymbol{\Phi}\frac{\partial \boldsymbol{b}}{\partial \boldsymbol{u}^{\mathrm{T}}}\right\} \in \mathbb{R}^{5\times 5}. \tag{4.15}$$

In Appendix C.1 we present a detailed derivation of explicit formulas for (4.15).

A direct observation from (4.15) is that if for a given measurement strategy encoded by $\boldsymbol{\Phi}$ the total number of samples resulting from this strategy is too small, the FIM becomes singular. However, due to the strong spatial correlation of neighboring A-scans, we can distribute the minimum number of Fourier coefficients over the complete set of spatial scan positions, which is usually much larger in magnitude than the required Fourier samples.

Further, neighboring A-scans are highly correlated, since the physical phenomena are rather smooth on this scale. Due to this, measurements at position (x, y) also yield a certain amount of information about $(x \pm \Delta x, y \pm \Delta y)$ for small Δx and Δy. Hence, choosing different mixing patterns $\boldsymbol{\sigma}_{n_x,n_y}$ or subselections $\boldsymbol{S}_{n_x,n_y}$ yields more information about the specimen for closely spaced sampling positions. Figure 4.4 shows $\boldsymbol{c}$ for varying n_f, normalized by the CRB obtained by sampling the full spectrum, $C_{\mathrm{N_t},\cdot}$. Since subsampling can only increase the CRB, we have that $C_{\mathrm{N_t},\cdot} \leqslant C_{\cdot}$.

Simulations are performed using the parameters in Table 4.3 placing a single point source at a depth of 33.30 mm beneath the center of the scanning grid. The randomized strategies each were averaged over 50 realizations of $\boldsymbol{\Phi}$nrg and $\boldsymbol{\Phi}_{\mathrm{rnd}}$ respectively. The transparent areas represent the range between the lowest and highest CRB values obtained from (4.15) for the randomized strategies. From Figure 4.4, the following can be noted. For larger n_f, the knowledge based sampling performs better than

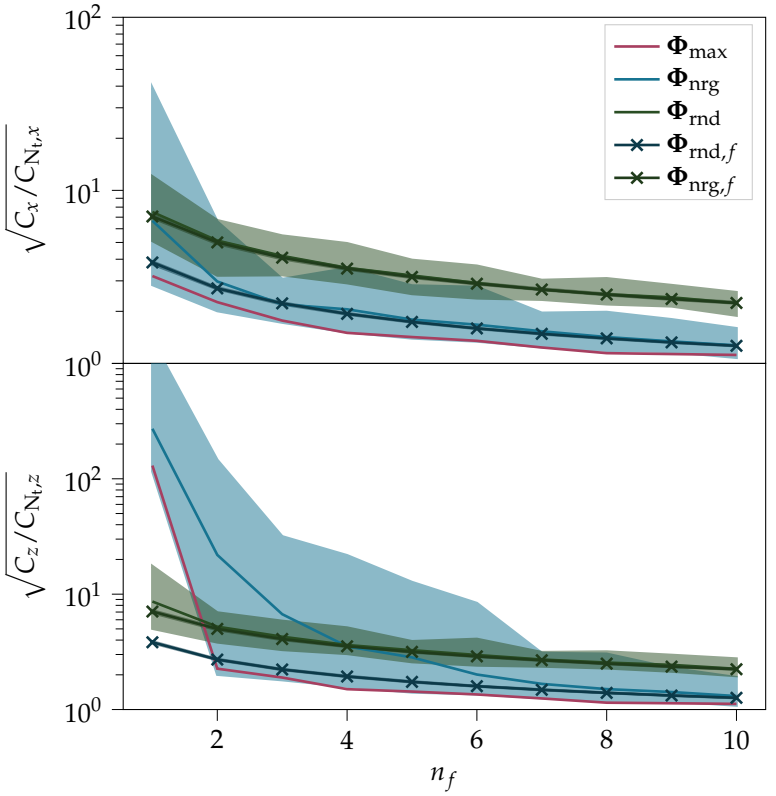


Figure 4.4 *The number of collected Fourier coefficients n_f has more influence on the estimation accuracy of the defects position in z-direction than in x- or y-direction.* – Asymptotic performance of the different sub-sampling strategies for varying n_f. ■

using $\mathbf{\Phi}_{rnd}$, since the bandwidth of the pulse is exhausted completely after a certain value for n_f. Further, $\mathbf{\Phi}_{max}$ provides a lower bound to $\mathbf{\Phi}_{nrg}$ in the considered scenario. This is due to the assumed Gaussian pulse shape, which leads to sampling more energy being better than sampling a higher bandwidth. For small n_f, the loss in performance is the highest for $\sqrt{C_z}$, which is expected since the compression along individual A-scans deteriorates the delay estimation of the echoes the most. This in turn influences the estimation of the scatterers' depths the most. Varying the coefficients for each scan position leads to a significant performance gain when using $\mathbf{\Phi}_{rnd,f}$ or $\mathbf{\Phi}_{nrg,f}$ and reduces the overall loss compared to $C_{N_t,\cdot}$ to a factor of 2-3.

4.1.6 Implementation considerations

In the following section, we analyze the proposed sampling and modeling strategies from Section 4.1.2 and Section 4.1.3 in terms of their hardware implementation effort in hardware as well as their computational complexity. Aside from the final measurement quality, implementation effort is a crucial aspect when choosing an architecture for a particular measurement application. The three-stage model introduced in Section 4.1.1.2 offers an intuitive abstraction, well suited for the following discussion:

To recap, the first *Data management* stage comprises of the acquisition frontend, generating a raw data stream of digital representations for the analogue pulse-echo signals followed by the interfaces for *Data handling and storing* into non-volatile memory. Representing the sense making stage, an off-site computation unit analyzes the stored raw data and distills it into interpretable information, which is usually the data of interest that motivates the deployment of the system. In our case, this is constituted by the scatterers' positions and their respective amplitudes. The final Decision making stage will not be discussed, since it bears no relevance to implementation aspects.

Based on a common example scenario, Table 4.2 summarizes key design parameters and performance indications when comparing the following three measurement architectures:

(I) A state-of-the-art system based on oversampling and SAFT, see Section 4.1.4.1.

(II) A critically sampling system using the physically motivated forward model $\boldsymbol{H}$ defined in this work, see Section 4.1.2.

(III) A sub-Nyquist sampling system extending Item (II) by the sub-Nyquist sampling strategies $\boldsymbol{\Phi}$ from Section 4.1.3.

A single architecture, satisfying the widely diverse and often conflicting requirements for the set of all measurement applications, usually cannot be found. Some handheld measurement units require battery operation on mobile network connections, while other units, irreplaceably built into long-lasting structures, deliver their raw data over wired data transmission networks. These conflicting scenarios and hence demands require making compromises between cost, size, power- and energy usage, ruggedness, data path or accuracy.

Notably, the major benefit of our proposed compressive architecture over common SAFT-based implementations (aside from imaging quality)

is that it allows trading the amount of collected raw data against computation complexity later in the process. In choosing the number of obtained Fourier coefficients per A-scan (in Section 4.1.7 we show that already a single one can be sufficient), our proposed compressed architecture relates the raw data rate to the actual amount of relevant signal information, rather than some artificial grid constraint (as is the case for oversampling SAFT). This makes it especially useful for applications suffering from *data handling* bottlenecks [124].

4.1.6.1 Effort for data acquisition

For the implementation of the signal acquisition, multiple strategies exist, where each has individual benefits and drawbacks which also depend on the application. Following, a general overview is provided to allow for good architectural decisions.

Direct Time-Domain Sampling

A standard ADC linearly represents the time-domain pulse-echo response signal as a digital vector. To fully represent all signal information in the raw data stream, the sampling rate f_s must be at least the critical rate $f_{\text{crit}} = 2f_{\text{max}}$, where f_{max} is the highest signal frequency component as stated by Theorem 2.1. To keep computation cost low, a simple delay-based SAFT model (disregarding more complex propagation effects) is commonly chosen. By using an oversampling factor $k_O = f_s / f_{\text{crit}}$, good depth-resolution with acceptable visual artifacts is achieved [128]. For common choices like $k_O = 16$ the amount of added redundant data is immense.

This work's generalized matrix formulation $\boldsymbol{H}$ adds an improved physical propagation model, increasing visual quality considerably at the price of higher computation cost during the sense making stage (see Section 4.1.6.2). However, data amount is reduced substantially, since choosing $k_O = 1$ does not negatively affect the resulting imaging quality as we see in Section 4.1.7.

Albeit very poor information efficiency and high demands on the ADC frontend and *data handling* stage, oversampling SAFT may still be advantageous when available computation resources are limited, i.e. for battery-powered on-site inspection systems.

Scenario	Volume Surface	all methods 100×100 Points		
	Max. round-trip-time	all methods 25 µs		
	Pulse bandwidth	all methods 10 MHz		
	Implemented system	(I) SAFT	(II) $\boldsymbol{H}$	(III) $\boldsymbol{\Phi H}$
Data Management	ADC (16 bit) rate [S/s]	$k_O \cdot 20\,\mathrm{M}$	$20\,\mathrm{M}$	$\leqslant$ **20 M***
	ADC data stream [bit/s]	$k_O \cdot 320\,\mathrm{M}$	$320\,\mathrm{M}$	$\leqslant$ **320 M***
	Samples per A-scan N_t	$k_O \cdot 500$	500	$\geqslant$ **1**
	A-scan data size**	$k_O \cdot 1\,\mathrm{kB}$	1 kB	**8 B**
	Volume data size**	$k_O \cdot 10\,\mathrm{MB}$	10 MB	**0.08 MB**
	...as normalized ratio***	$\approx 10^{3.3}$	$\approx 10^{2.1}$	**1**
Sense making	Computation Complexity	$+$	$\circ$	$-$
	...as normalized ratio***	**1**	$\approx 10^{2.8}$	$\approx 10^{4.8}$
	Depth-axis resolution	fixed	**variable**	**variable**
	Considers physical model	no	**yes**	**yes**
	Approximates inverse problem	no	no	**yes**
	Point focus quality	$-$	$+$	$(\circ \ldots +)$

k_O = *Oversampling factor* ; k_F = *FISTA iterations*

* *equal to when using FFT on raw samples,* $<$ *when employing [111]*

** *Data storage, complex single-precision float (8 byte)*

*** *Relative, given for* $k_O = 16$ *and* $k_F = 50$ *and* $\Delta z = t_s \cdot c$.

Table 4.2 *Depending on the application at hand the compression and reconstruction schemes perform differently.* – Implementation effort and performance of the system (III) proposed in this work compared to the two example systems (I) and (II).

■

Digital Fourier Coefficient Sampling

Additionally, we show how to extend (II) by a sub-sampling/compression matrix to form the CS system (III), allowing for source-compression by keeping only a few Fourier coefficients of each A-scan. In reconstructing the signal describing the 3D volume from all A-scans jointly, spatial

redundancy is exploited such that keeping as low as one coefficient per A-scan is sufficient in scenarios, when sampling locations yield correlated information. While this greatly improves *data handling*, the computation-intense algorithms from Section 4.1.4 must now be applied during the sense making stage. If one has access to the sampled values produced by the transducer, the Fourier coefficients can be computed directly from the ADC output using the FFT or the Goertzel [131] algorithm. In this case the implementation effort up to the ADC is identical to the critical sampling case in "Direct Time-Domain Sampling". This approach works best when collected data is processed off-site, where bulk computation power is readily available, and the measurement device can afford to compute a few Fourier coefficients on-the-fly.

Analogue Fourier Coefficient Sampling

Retrieving the Fourier coefficients for system (III) can also can be implemented using sub-Nyquist sampling, as proposed in [111, Sec. 3]. This requires the signal to be filtered by a Sum-of-Sincs filter in analog domain, which is sparsely sampled with an ADC operating below the critical sampling rate, since if we wish to acquire n_f samples in frequency domain, the authors of [111] show that then also only n_f appropriately filtered samples in time-domain suffice. This approach allows a "true" Compressed Sensing implementation, since we do not need direct access to a sampled version of the signal acquired by the transducer, but rather we directly implement the compression as modeled by (2.1). From the collected samples, the desired Fourier coefficients (denoted as the set $\mathcal{K}_n$ in [111, Sec. 3]) are retrieved by solving a system of linear equations.

Multi-channel setups, such as [O16], may easily be supported by moving from sparse- to interleaved sampling of multiple analogue channels by combining a single ADC with a multiplexer. This makes this approach attractive for low-power single- or multichannel applications. Further combined with integrated Complementary Metal Oxide Semiconductor (CMOS) technology, low-cost and low-footprint multi-channel frontends can be achieved. This allows for the measurement strategies $\mathbf{\Phi}_{\text{nrg}}$ and $\mathbf{\Phi}_{\text{max}}$ to be implemented. To realize $\mathbf{\Phi}_{\text{rnd}}$ the multiplication of $\mathbf{\Sigma}_i$ additionally needs to be implemented in the analog domain.

4.1.6.2 Effort for Data Processing (Computation Complexity)

In the *Sense making* stage, the raw data is condensed to interpretable information by a processing unit, where computation effort is a crucial quantity of interest. Since the main computing effort for SAFT, FISTA and ARPACK lies in $\phi_{\boldsymbol{\Phi H}}$ and $\beta_{\boldsymbol{\Phi H}}$, we study the complexity of Algorithm 4.1 in relation to the spatial grid sizes N_x, N_y and $N_t = N_z$. The cost of $\phi_{\boldsymbol{\Phi}}$ is dominated by the cost of the $N_x \cdot N_y$ executed FFTs, yielding a complexity of $\mathcal{O}(N_t \log(N_t) N_x N_y)$. For $\boldsymbol{H}$, we see that we have to compute the generating elements[2] in $\mathcal{O}(N_t^2 N_x N_y)$ and then compute all 2D convolutions in $\mathcal{O}(N_t N_z N_y \log(N_y) N_x \log(N_x))$. The total computation complexity for the SAFT (cf. Section 4.1.4.1) matrix-free matrix-vector product $\phi_{\boldsymbol{H}}(\boldsymbol{x})$, which also is equal to the complexity of the product $\phi_{\boldsymbol{\Phi H}}(\boldsymbol{x})$, is $\mathcal{O}(S) = \mathcal{O}(N_t N_z N_y \log(N_y) N_x \log(N_x))$. Calculating $\sigma_{\max}$ requires k_A ARPACK iterations yielding a complexity of $\mathcal{O}(k_A S)$ for a single execution of the routine. Since $\sigma_{\max}$ may be reused, the complexity for one reconstruction is $\mathcal{O}(k_F S)$, scaling only with the k_F iterations of FISTA.

We carried out an empirical study on the influence of the grid sizes in z-direction independently from x and y in Figure 4.5[3]. As expected, the transform scales quadratically with N_z and N_t. Varying N_x or N_y influences the FFT size, exhibiting some ripple in the run-time plots of Figure 4.5, depending on the prime factorization of $2N_x - 1$ and $2N_y - 1$.

The comparison in Figure 4.6[4] makes use of a previous "raw" CUDA-C [132] implementation and this already shows the advantage of exploiting structure in $\boldsymbol{H}$ to get a matrix-free formulation in terms of computation time. Hence, implementing this matrix-free algorithm naïvely on a GPU allows to carry out the reconstruction in even larger SSR scenarios on standard work station hardware.

In case of the simple delay-based oversampling system (I) , which required no convolutions, as outlined in Section 4.1.6.1, a very low computation complexity of $\mathcal{O}(k_O N_z N_y N_x)$ can be attained. This is due to the fact that only the echo delays are considered in the model, as is often used for the analysis of oversampling SAFT data (see section on "Direct time-

[2] Note that this can be pre-computed if $\boldsymbol{H}$ fits into the memory. If they need to be computed on-the-fly, the computation complexity of the generating elements using the model from Section 4.1.2.1 is neglibible compared to the computation of the actual matrix-vector product.

[3] The benchmark is performed on a PC equipped with a Intel Xeon E5-1620 v4 CPU 3.50 GHz, 32 GB of memory, and a Nvidia GeForce GTX 980 GPU.

[4] The benchmark was performed on an NVIDIA Titan XP, an Intel(R) Core(TM) i7 CPU 975 at 3.33 GHz was used as a CPU.

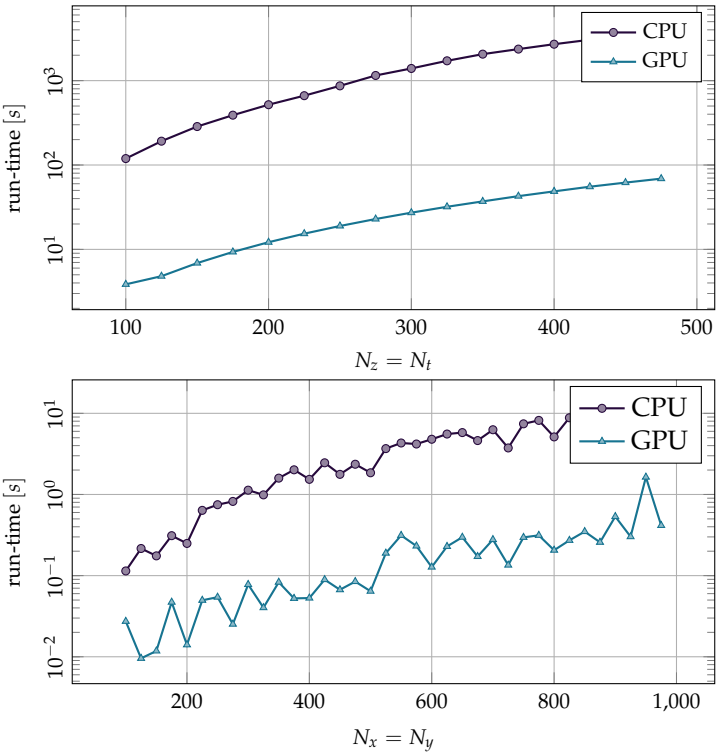


Figure 4.5 *The GPU-based implementation substantially outperforms the CPU-based one.* – Empirical study of the influence of the scene dimensions on the run-time of $\phi_{\boldsymbol{\Phi H}}$. For varying N_z we choose $N_x = N_y = 100$, for varying $N_{x,y}$ we choose $N_z = 5$. ■

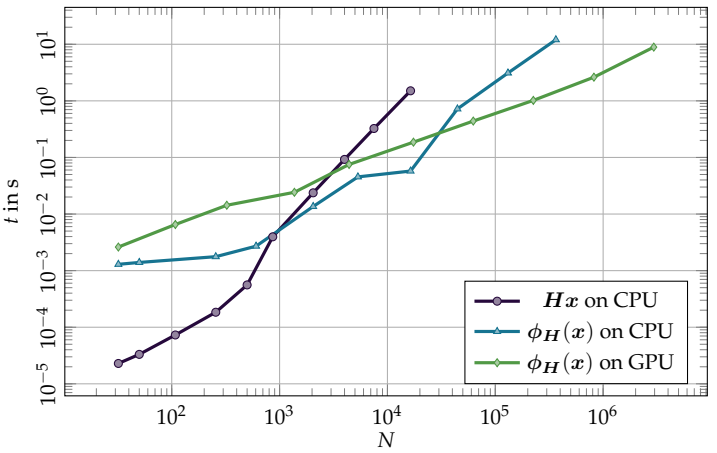


Figure 4.6 *At a certain problem size the overhead introduced by the GPU is compensated for.* – Comparison of the matrix-free projection to the standard matrix-vector product not exploiting any structure in $\boldsymbol{H}$. ■

domain sampling", the model formulation can be carried out without the necessity for FFT operations.

Compared to this model, the complexity of system (II) is higher by a factor of $\mathcal{O}(\frac{N_t}{k_O}\log(N_y)\log(N_x))$. In the case of system (III), $\boldsymbol{\Phi}\boldsymbol{H}$ is applied twice in each FISTA iteration, increasing the computation effort by another factor of $2k_F$ as k_F denotes the number of iterations of FISTA. In exchange, system (III) actually tackles the inverse problem, as opposed to the other systems that solely perform projections.

In total, Table 4.2 reports the total computation complexity to be $\approx 10^3$ times higher for (II) and $\approx 10^5$ times higher for (III) compared to simple oversampling SAFT for the example scenario. However, this comes along with a significant reduction of total collected raw data amount and an improvement of reconstruction quality (see e.g. Figure 4.11).

Remark 4.5. For the example setting, Table 4.2 reports a data reduction of ≈99 % when compared against system (II) and ≈99.90 % when compared against system (I), solely from reducing spatial redundancy by implementing (III).

When applying the physically motivated forward model $\boldsymbol{H}$, it is possible to choose the depth-axis resolution Δz independent of the particular sampling rate f_S. This is opposed to the state-of-the-art oversampling SAFT system, where a high f_S is required to achieve small Δz, and as a

consequence, measurement data must be reacquired.

The implementation of the strategies $\boldsymbol{\Phi}_{\text{nrg}}$ and $\boldsymbol{\Phi}_{\text{max}}$ is slightly easier compared to $\boldsymbol{\Phi}_{\text{rnd}}$, especially in terms of hardware components in the analog domain, since they do not require the additional element-wise multiplication due to $\boldsymbol{\Sigma} = \boldsymbol{I}$.

The benefits of the proposed CS-architecture (III), among which is that it actually approximates a sparse solution of the inverse problem, appear to justify the large computation complexity, considering the recent advances in CPU and GPU performance. ∎

4.1.7 Numerical Simulations

In the following, we will present several reconstructions using FISTA. The regularization parameter λ is chosen as

$$\lambda = \mu \|\beta_{\boldsymbol{\Phi}\boldsymbol{H}}(\boldsymbol{y})\|_\infty,$$

with $0 < \mu < 1$. The optimal choice of the parameter λ is still an open question to research. However, our experience has shown that small variations of μ do not significantly change the result FISTA converges to and hence they have been set manually based on a priori knowledge on the measurement scenario and assuming that sparser scenarios require $\mu \to 1$, since μ allows to balance the trade-off between sparsity and least squares fit. To give a concrete example: The results of the comparison based on measurement data in Figure 4.11 do not change substantially by e.g. choosing $\mu = 0.3$ or $\mu = 0.5$ instead of the given $\mu = 0.4$. The main goal of this section is to compare the different proposed sampling strategies in different scenarios. It can be generally noted that in the uncompressed case FISTA converges to the correct solution in every presented scenario. Further, we note that when using subsampling, the expected imaging performance becomes visible after less than 10 iterations and adding further steps *does not* add significant changes to that trend. Hence, all comparisons are made using a constant amount of iterations for each scenario. This comes with the additional benefit that all comparisons are made for equal computational effort. The largest singular value is computed using ARPACK.

In addition to the different sampling strategies, different versions of the model matrix $\boldsymbol{H}$ can be used for reconstruction: (i) $\boldsymbol{H}$ represents the complex analytic model as defined in Section 4.1.2, (ii) $\text{Re}\{\boldsymbol{H}\}$ is equivalent to modeling $p(t)$ as a Gaussian windowed cosine, cf. [115].

Further, with respect to (ii) we can choose $\boldsymbol{S}_{n_x,n_y}$ either symmetrized or one-sided as outlined in Section 4.1.3.2.

The 3-D results are plotted as 2-D images by taking the maximum absolute value of an axis along the two remaining axes. For the $x-y$ plane this is called a C-scan image and for the $x-z$ plane it is called B-mode image. In the complex-valued case, this calculates the envelope.

4.1.7.1 Experiments using Synthetic Data

We first consider reconstructions of synthetic data sets to showcase the performance of the different sampling strategies, where simulate a shoebox steel specimen. The parameters used for the simulation are listed in Table 4.3 and the model for the pulse shape and transducer directivity are the same as in Section 4.1.5. The imaging region starts at a depth of $z_d = 29.60\,\text{mm}$. All data sets presented in this section have been simulated noise free by simply evaluating $\boldsymbol{H}\boldsymbol{a}$ for a previously defined $\boldsymbol{a}$ as the goal is to compare only the performance of the different sampling strategies. The ground truth $\boldsymbol{a}$ is designed by setting $[\boldsymbol{a}]_d = \mathrm{e}^{\jmath\pi/4}, d \in [D]$ and zero otherwise, where $\mathcal{D}$ is the set of indices forming the defect as a sum of point sources. The amplitude is chosen since it represents the largest possible phase offset that can arise from a scatterer actually lying "between" two grid points. It is therefore also the most challenging scenario for the popular choice of $Re\{\boldsymbol{H}\}$ as a forward model. As scatterers we consider two types of defects. In the first scenario, the specimen contains a straight line simulating a defect at depth $25z/\Delta z$. In the second scenario, the specimen contains a rectangle of size $40\sqrt{(z/\Delta z)^2 + (x/\Delta x)^2} \times 40y/\Delta y$ diagonally placed along the z-x-plane as a defect. The resulting datasets can be compressed by applying different incarnations of $\boldsymbol{\Phi}$ to $\boldsymbol{H}\boldsymbol{a}$. The compressed datasets are then reconstructed using FISTA and the largest singular value is computed using ARPACK if not explicitly stated otherwise. The results are plotted using a so-called C-scan image, i.e. we plot the maximum amplitude of an axis along the two remaining axes.

As a first scenario, we compare the reconstructions using different subsampling strategies sampling only $n_f = 1$ coefficient per A-scan. In addition to the different sampling strategies, different versions of the model matrix $\boldsymbol{H}$ can be used for reconstruction: (i) $\boldsymbol{H}$ represents the complex analytic model as defined in Section 4.1.2, (ii) $\mathrm{Re}\{\boldsymbol{H}\}$ is equivalent to modeling $h(t)$ as a Gaussian windowed cosine, cf. [O8, 115]. The latter is only able to reconstruct a real-valued version of $\boldsymbol{a}$. Further, with respect to (ii), there exist two approaches on how to define $\boldsymbol{S}_{n_x,n_y}$ for the

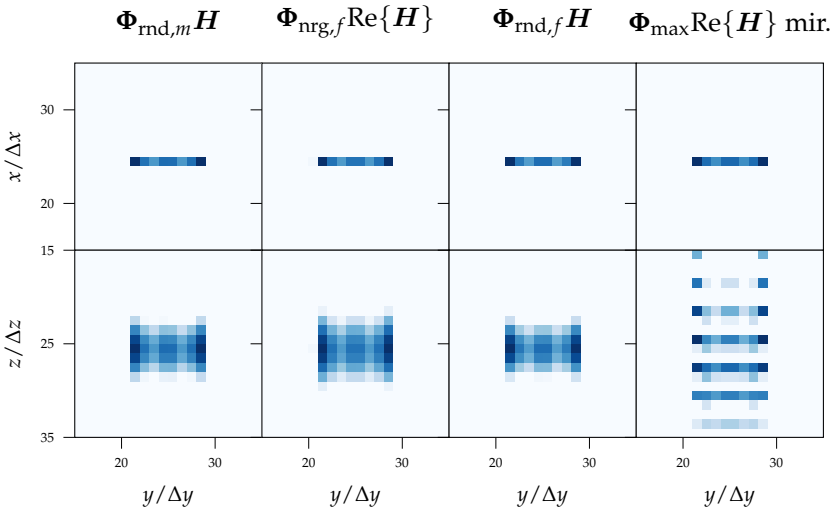


Figure 4.7 *The proposed Fourier subsampling mostly influences depth resolution.* – Reconstruction results from simulated data for $n_f = 1$ using FISTA with $\mu = 0.6$ stopped after 80 iterations. ■

reconstruction: $\mathbf{\Phi}_{\mathrm{max}}\mathrm{Re}\{\boldsymbol{H}\}$ inputs only the n_f coefficients at one half of the symmetric spectrum into the reconstruction algorithm, "$\mathbf{\Phi}_{\mathrm{max}}\mathrm{Re}\{\boldsymbol{H}\}$ mirrored" mirrors these coefficients to the other half. By this, essentially $2n_f$ coefficients are input but only n_f coefficients need to be measured at each scan. The latter reproduces the setup as presented in [O8].

The results are depicted in Figure 4.7. All strategies correctly reconstruct the shape of the target in the x-y plane since the lateral focusing is not deteriorated (which is also in accordance with the asymptotic results in Section 4.1.5). Note that the difference between the second column of Figure 4.7 (a) and the second column of Figure 4.7 (b) is that in (a) only

N_x	N_y	f_s	Δx	Δy	c_0
50	50	20 MHz	0.50 mm	0.50 mm	5920 m/s
θ	N_z	f_c	t_0	z_d	ξ
30°	50	3.20 MHz	10 µs	29.60 mm	$(0.65 f_c)^2$

Table 4.3 Simulation parameters for Section 4.1.7.1 ■

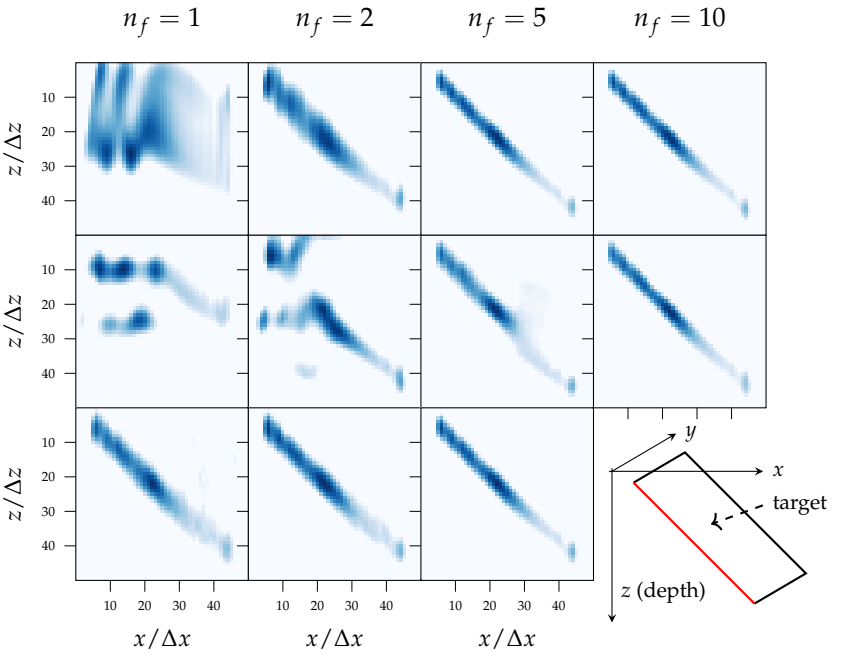


Figure 4.8 *The strategy $\mathbf{\Phi}_{\mathrm{nrg},f}$ allows to exploit the spatial correlation across A-scans by varying the sampled frequencies randomly.* – Left to right: $n_f = 1, 2, 5, 10$, except for the figure entitled Ground Truth. Top to bottom: $\mathbf{\Phi}_{\mathrm{max}}$, $\mathbf{\Phi}_{\mathrm{rnd}}$, $\mathbf{\Phi}_{\mathrm{nrg},f}$. Note that the missing result for $\mathbf{\Phi}_{\mathrm{nrg},f}$ with $n_f = 10$ is visually equivalent to the reconstruction for $\mathbf{\Phi}_{\mathrm{nrg},f}$ with $n_f = 5$. Reconstructed using FISTA with $\mu = 0.1$, stopped after 80 iterations. ■

one half of the spectrum is sampled although a real-valued matrix $\boldsymbol{H}$ is used. Using a real-valued model as well as a constant $\mathbf{\Phi}$ for all scan positions results in the worst localization in depth. In addition, the peak along z in the reconstruction is at a wrong position due to the defect being off-grid (phase offset) (cf. Figure 4.7 (b), second column). In contrast, as expected from the CRB, using distinct subsampling patterns restores the localization accuracy along the z-axis. Further, it can be seen that in the case of random uniform subsampling, it is equivalent to vary the mixing pattern or the set of Fourier coefficients per A-scan. Note that $\mathbf{\Phi}_{\mathrm{rnd},f}\boldsymbol{H}$ and $\mathbf{\Phi}_{\mathrm{nrg},f}\boldsymbol{H}$ perform equivalently in this scenario, which is why the latter is not depicted. Finally, it can be noted that using $\boldsymbol{H}$ instead of $\mathrm{Re}\{\boldsymbol{H}\}$ leads to a slightly sparser solution.

As another scenario, we compare the subsampling schemes for varying n_f in a more complex scenario. The results are depicted in Figure 4.8 (the bottom right figure shows the side view of the square as ground truth). The chosen scenario results in a measurement, where every A-scan sees echoes from almost all point sources. The ramifications of this are as follows: Choosing a constant subsampling pattern for all A-scans leads to a failed reconstruction for n_f too small (see the first and second row of Figure 4.8). This is basically also what our findings about the CRB already infer analytically. On the other hand, varying the pattern (third row) strongly improves the reconstructed image. This again emphasizes that the large number of (necessary) spatial scanning positions reduces the number of required temporal measurements. The spatial grid spacing needs to be small enough to ensure that defects with a given minimum size are still detected. However, due to this small spatial grid spacing, the scan at position (x, y) also yields a certain amount of information about $(x \pm \Delta x, y \pm \Delta y)$.

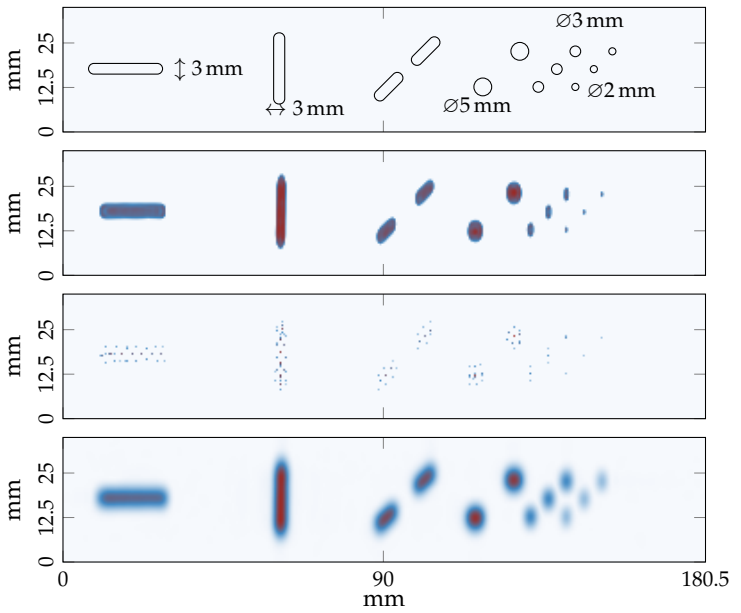
4.1.7.2 Synthetic Aperture Measurements

Next, we conduct numerical experiments based on measurement data of a steel specimen that was constructed for testing different ultrasonic imaging approaches. To this end it contains predefined and precisely known defects of various sizes, shapes, depths and distances.

Reconstructions from Compressed Data using OMP

In order to compare the imaging quality of OMP given in Algorithm 3.1 and FISTA as given in Algorithm 3.2 we execute them for data that is uncompressed, hence $\boldsymbol{\Phi} = \boldsymbol{I}_{N_t N_x N_y}$, hence we attain results that are not affected by the compression scheme. The data was acquired for a steel specimen containing two oblong holes that are 3 mm wide and 17 mm long, two smaller oblong holes with 4 mm length, and two flat bottom holes with 5 mm, 3 mm and 2 mm diameter, respectively. The measurement was performed using a transducer with $f_c = 4\,\text{MHz}$. However, due to the attenuation of high frequency components within the propagation medium, the parameters listed in the table in Figure 4.9 where chosen to model $\boldsymbol{H}$. The effective center frequency f_c was manually extracted from the measurement data.

Figure 4.9 shows the reconstruction results of the specimen sketched in the top figure using SAFT, FISTA after 50 steps with $\mu = 0.2$ and OMP after 150 steps. It can be seen how FISTA reconstructs the defects at their



c_0	f_s	Δx	Δy	θ
5920 m/s	80 MHz	0.50 mm	0.50 mm	15°
Δz	N_x	N_y	N_z	f_c
37 µm	361	70	520	3.40 MHz

Figure 4.9 *OMP is not well suited for estimating the size and shape of defects realiably* – Top to bottom: Sketch of the specimen, FISTA reconstrution after 50 steps with $\mu = 0.2$, OMP reconstruction after 150 steps, 3D SAFT image. ■

exact sizes with relatively sharp edges. In contrast while the additional blur in the SAFT image impedes exact sizing. OMP, due to its greedy and thus local strategy, creates sparser solutions than FISTA, which makes the reconstruction result harder to interpret, since the contours of the defects are hard to determine. Especially, when using the same amount of calls to $f_{\boldsymbol{H}}$ and $b_{\boldsymbol{H}}$ as for the execution of FISTA, i.e. spending the same computational effort in both algorithms.

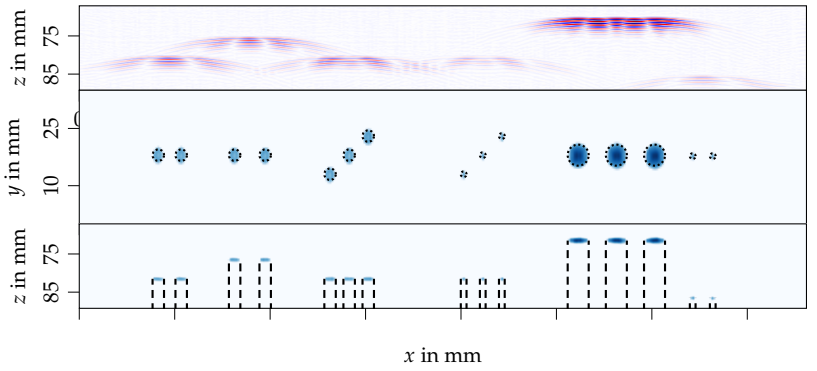


Figure 4.10 *A few steps of FISTA suffice to accurately locate defect positions and size, even for strongly overlapping echoes.* – Top and center plot: Top and side view of a FISTA reconstruction using $n_f = 1$ and $\boldsymbol{\Phi}_{\text{rnd},m,f}\boldsymbol{H}$. Here, FISTA was run for 20 steps with $\mu = 0.4$. Bottom plot: B-scan slice through the actual measurement data showing the strongly overlapping echoes. ■

Reconstructions from Compressed Data using FISTA

In the following, we compare the different approaches defined in Section 4.1.3 based on synthetic aperture measurements of a steel specimen. The specimen contains several flat bottom holes with diameters ∅2 mm, ∅3 mm and ∅5 mm, representing artificial flaws. The measurements were taken using time domain sampling at a sampling rate of $f_s = 20\,\text{MHz}$. Fourier subsampling was simulated by calculating an FFT and keeping only n_f Fourier coefficients per A-scan based on the chosen strategy. To calculate $\boldsymbol{H}$ the same Gaussian pulse model and transducer directivity as in Section 4.1.7.1 is used. The center frequency of the transducer is at $\omega_c = 2\pi f_c$ and $\xi = (0.8 f_c)^2$. The specimen was scanned with $\Delta x = \Delta y = 0.50\,\text{mm}$. The opening angle of the transducer is set to $\theta = 30°$. The speed of sound in this steel is assumed to be $c_0 = 5920\,\text{m/s}$. For the reconstruction, FISTA is used with $\mu = 0.4$ and stopped after 20 iterations.

Figure 4.10 shows the result using $\boldsymbol{\Phi}_{\text{rnd},m,f}$ using $n_f = 1$ per A-scan. The top figure shows a C-scan view from the top. The bottom image shows the projection onto the y-axis. In both cases a ground truth sketch is superimposed onto the image. It clearly shows that the reconstruction reproduces the positions of the defects accurately in all three dimensions as well as their extent in x-y-direction.

Figure 4.11 shows a zoomed in comparison of all investigated sub-

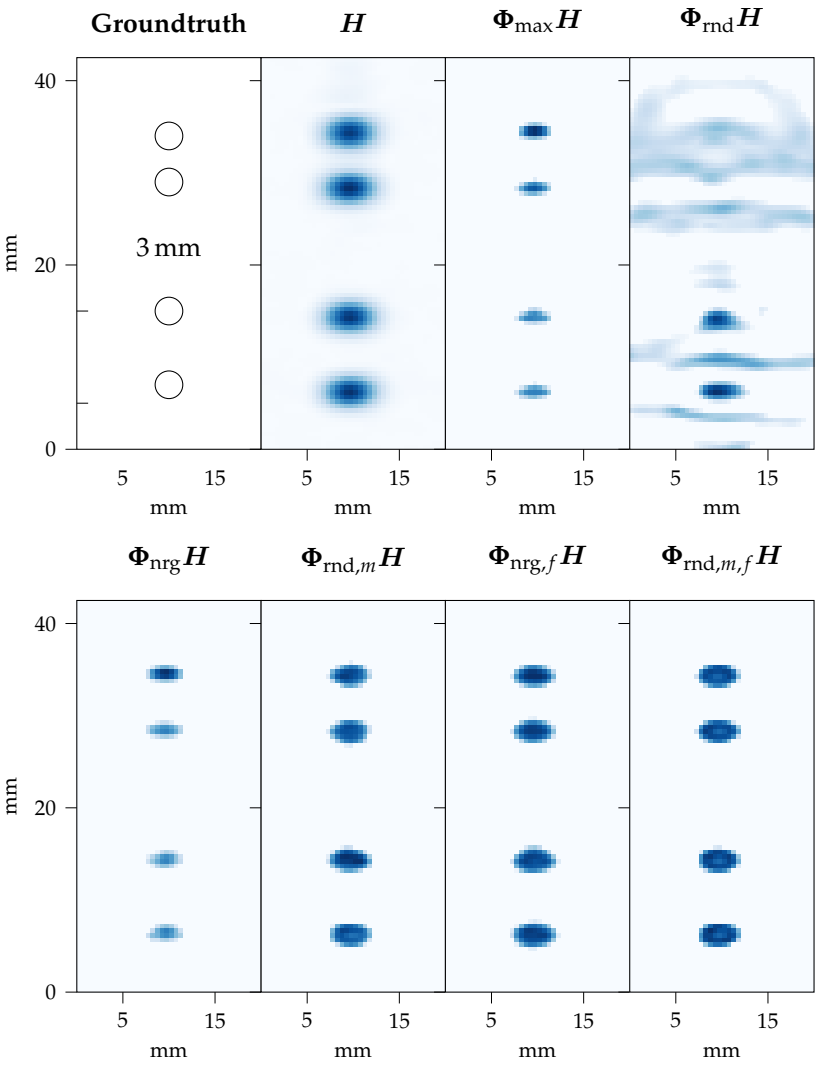


Figure 4.11 *If compression is done random enough, one Fourier coefficient is sufficient.* – The second figure from the top left shows the standard uncompressed SAFT reconstruction. The six remaining figures show the reconstructions based on the different sampling schemes indicted by the different $\mathbf{\Phi}$. The reconstructions are computed using FISTA with $\mu = 0.4$, stopped after 20 iterations. ∎

sampling strategies as well as the conventional uncompressed SAFT reconstruction. Comparing the top row with the bottom row reveals the advantage of varying the mixing pattern and/or the measured Fourier coefficient, since the edges of the defect are much more pronounced.

Remark 4.6. Varying the sample patterns for each A-scan provides a significant gain compared to only using a single constant pattern. With this improvement, $n_f = 1$ can already be enough to obtain high resolution reconstructions and $\mathrm{Re}\{\boldsymbol{H}\}$ performs worse since it is incapable of dealing with off-grid contributions.

The inclusion of the symmetric counterparts of the Fourier coefficients of the (real-valued) measurements into the reconstruction leads to a deterioration of the reconstruction.

The CRB is a valid proxy to evaluate the performance of the different subsampling strategies, since at least qualitatively the CRB is able to predict the differences in the various approaches. So, the improvement when varying the sampling or mixing patterns is well predicted.

Both the CRB and the reconstruction show how the proposed subsampling mainly affects the performance along the depth axis: the depth resolution can be deteriorated substantially while still maintaining a high quality C-scan image along the x-y plane. ∎

4.1.8 Conclusion

In this chapter, we developed and analyzed a novel CS-based ultrasound acquisition framework for synthetic aperture NDT. The proposed new strategies $\boldsymbol{\Phi}_{\ldots,f}$ and $\boldsymbol{\Phi}_{\ldots,m}$ are superior to existing state of the art strategies. However, this is only true, if we consider a forward model that exploits the high correlation between adjacent scans such as the employed 3-D model for the reconstruction. By doing so, the number of necessary Fourier coefficients per scan can even be reduced to the absolute minimum of a single coefficient even in realistic scenarios. Hence, we directly see the relation of AOI-C and AOI-M, since in the spirit of Remark 2.3 an improved data model allows more aggressive compression. In total, $\boldsymbol{\Phi}_{\mathrm{nrg},f}$ provides the superior choice, since it provides the best imaging performance (together with $\boldsymbol{\Phi}_{\mathrm{rnd},m}$ and $\boldsymbol{\Phi}_{\mathrm{rnd},f}$) but allows for a simpler hardware architecture.

The employed matrix-free implementations by means of `fastmat` as outlined in Section 3.4 represent a practical approach for this even if the chosen reconstruction algorithm requires additional quantities of the underlying matrix, as illustrated for the approximation of the largest

singular value necessary in FISTA. Additionally these computational advantages can even be improved by moving the calculations to a GPU.

In terms of modeling, it is beneficial to use the analytic signal instead of the widely used real-valued model of the RF signal originating from [104]. The increasing demands in computation complexity are compensated by the improved imaging quality by actually approximating a solution to the inverse problem compared to only calculating an image based on a simple heuristic.

The numerical simulations are in agreement with the theoretical observations. This demonstrates that the question of which and how many Fourier coefficients are needed can be directly answered by evaluating the CRB, which greatly simplifies the parameter specification for a given target scenario. To conclude, the presented results indicate that the use of CS is beneficial in the context of synthetic aperture ultrasound NDT.

4.1.9 Outlook

The fact that for the reconstruction the model for the sampling kernel and the propagation are separable allows to easily tailor the imaging pipeline to the requirements of different scenarios and independently optimize the respective implementations.

Finally, the proposed sampling strategies can also be straightforwardly included into a multi-channel setup and combined with additional spatial sub-sampling as presented in [O16].

4.2 Compressed Sensing for an Ultra Wide-Band Radar

In this chapter, we propose a compact hardware architecture for measuring sparse channel IRs by combining the M-Sequence UWB measurement principle with the concept of CS. A channel is excited with a periodic M-sequence and its response signal is observed using a RD, which captures pseudo-random linear functionals of the response signal at a rate significantly lower than the measurement bandwidth. By doing so we directly implement Equation (2.1) directly in the analog domain, hence we simultaneously address AOI-C and AOI-H. The excitation signal and the RD mixing signal are generated from compactly implementable Linear Feedback Shift Register (LFSR) and operated from a common clock. A linear observation model is derived that allows reconstructing the sparse Impulse Response (IR) from a set of observations using SSR. For the SSR stage we employ a matrix-free model as in Chapter 3 where we exploit the choice of synchronous LFSRs as signal generators, resulting in low computational complexity by means of OMP, which contributes to AOI-R. For validation, real measurement data of a time-variant channel containing multipath components is processed by synthetic models of our proposed architecture and the classic M-Sequence method. We show successful IR recovery using our architecture based on SSR, outperforming the classic method significantly in terms of IR measurement rate. Compared to the state of the art, the proposed architecture allows faster measurements of sparse time-varying channels, resulting in higher Doppler tolerance without increasing hardware or data stream complexity. Note that the ideas in this section lead to the submission of a patent application [P1].

4.2.1 Introduction

Estimating the IR of a linear system is a core task in many engineering applications, including system identification, channel sounding, radar, localization and others [133, 134, 135, 50, O10]. The knowledge of the respective IR often allows to infer further information about the system of interest. More often than not, these IRs are not entirely static but (slowly) changing in time, e.g. due to motion of scattering objects in wireless propagation conditions, which in turn give rise to Doppler shifts. In such scenarios, the IR needs to be measured repeatedly and the repetition rate we can sustain determines the Doppler range we can support [136].

A wide variety of principles exists to measure IRs, including impulse methods, Frequency-Modulated Continuous-Wave (FMCW) or methods based on Pseudo-Noise (PN) sequences [137]. Due to their advantages in implementation complexity and the low achievable crest-factor, here we focus on the latter category. For PN methods, it is common to excite the linear system with a periodic PN signal of high bandwidth. The trade-off between hardware complexity and achievable Doppler range is then controlled by subsampling the received signal and varying the subsampling factor, capitalizing on the fact that the periodic signal can be recovered from samples taken in subsequent periods after proper rearrangement [138]. Here by subsampling we mean that we are actually dropping, hence not using samples. So naturally, a drawback of subsampling is that most of the receive signal remains unused and measurement time is increased considerably, drastically reducing IR measurement speed and tolerable Doppler range.

Measuring sparse IRs of linear systems or channels based on CS theory has been demonstrated using different concepts. The work on sub-Nyquist radar [139] and the Modulated Wideband Converter (MWC) [140] perform multiple observations in parallel and sample in the Fourier domain, which becomes increasingly infeasible at higher operating frequencies. In [141] the RD concept was applied to pulse-based UWB IR measurements observing from a single channel over multiple excitations. Although the concept can be implemented for very high operating frequencies, generating the RD mixing signal efficiently is not addressed in [141] and the signal basis is highly susceptible to interference.

In this work we propose an extension to the M-Sequence Method (MSM) of [138] that uses the CS principles outlined in Chapter 2 and Chapter 3 to significantly reduce the measurement time, yet maintaining the low hardware implementation complexity known from the MSM. Applying the RD concept [142, 143], we obtain sufficient information about the IR from only a few observations of pseudo random linear projections. Assuming the IR is sparse, in Section 2.2 we have argued that it can be recovered via ℓ_0- or ℓ_1-minimization techniques efficiently. The linear system model of this architecture is composed of structured matrices, which when exploited during implementation, yield substantial benefits in computation efficiency by exploiting the methods introduced in Chapter 3. The proposed architecture is targeted for very high operating frequencies well exceeding 10 GHz and specifically considers aspects of hardware implementation feasibility.

4.2.2 Measuring Impulse Responses

We start by describing a well established method in order to acquire suitable samples of a band-limited IR. The IR $h : \mathbb{R} \to \mathbb{C}$ with $h \mapsto h(t)$ of a linear system can be measured by exciting its input port with an impulse $x : \mathbb{R} \to \mathbb{C}$ $x \mapsto x(t)$ resembling the (Dirac) δ-function as closely as possible. Then, the IR can be directly observed at its output signal y which satisfies $y = x * h$, yielding $y \approx h$. Here, $*$ denotes the convolution operation. In practice, h can be assumed to be band-limited, exhibiting a maximum frequency component $f_{\max}$ as introduced in Section 2.1.1. It is then sufficient to use an equally band-limited approximation of the Dirac impulse δ as excitation signal. Additionally, we assume that y is *sparse* in the sense that h only exhibits a few spikes, such that the convolution of x with h is well approximated by

$$y = \sum_{i=1}^{S} \mathcal{S}_{\tau_i} x, \tag{4.16}$$

where $\mathcal{S}_{\tau_i} : \{x : \mathbb{R} \to \mathbb{R}\} \to \{x : \mathbb{R} \to \mathbb{R}\}$ is the periodic shift-operator of functions. So we have that $x(t) = \{\mathcal{S}_\tau x\}(t - \tau)$ for all x, t and τ.

Additionally, in many applications, h is not static but actually slowly varying over time. Furthermore, h is usually also only non-zero up to $\tau_{\max}$ in time-domain, so he have $|h(t)| \approx 0$ for all $t > \tau_{\max}$. Measuring the IR can then be repeated periodically at a rate of up to $f_{\mathrm{IRF}} = \tau_{\max}^{-1}$, allowing to also measure time-variant systems as long as they can be assumed to be stationary within the observation time frame $\tau_{\max} = f_{\mathrm{IRF}}^{-1}$.

Obtaining the IR as described using impulse excitation, the generation of sharply peaked, steep impulses x is required in order to achieve a large measurement bandwidth. For good dynamic range in the presence of noise, pulses of large amplitude must be generated. The Crest factor (CF) of the input signal x defined as

$$x \mapsto CF(x) = \frac{\|x\|_\infty}{\|x\|_2} = \frac{\sup_t |x(t)|}{\sqrt{\int_{-\infty}^{+\infty} |x(t)|^2 \, \mathrm{d}t}}$$

is commonly used as a metric for characterizing a signal's peak-to-root mean squared (RMS) dynamic. This method imposes severe demands on the circuit capturing the output y, since frequency components of up to $f_{\max}$ must be preserved with both, high linearity and high dynamic range. Furthermore, direct coupling of the excitation pulse into the capturing

circuitry is usually quite strong and must be tolerated without damage or impeding performance.

Due to the severity of these constraints for demanding measurement applications, more advanced methods have been presented to measure the IR, especially for UWB systems or channels. Some of them focus on optimizing the excitation signal x, some target the linearity or the CF and again others leverage on the dynamic range by applying additional signal processing on the collected data stream. But also the implementation effort can motivate to go for alternative approaches. For example, the FMCW method employs a narrowband continuous-wave signal as excitation signal x, which is swept through the frequency band over time. This greatly reduces hardware complexity, and also improves linearity and reduces coupling due to the low instantaneous bandwidth of x. However, in these cases, the attainable measurement rate is rather low, since the sweeping is rather time-consuming. Hence, in time-varying scenarios the resulting clutter must be handled.

Selecting the excitation signal x as a sum of carefully chosen, periodic narrowband signal components ("*multi-tone*"), achieves high instantaneous bandwidth and reduces measurement time. The IR is then retrieved by decorrelating y with x. Optimizing x such that $x \circledast x \approx \delta$ can be assumed, the computationally expensive decorrelation operation may be replaced by $y \circledast x$, where $\circledast$ represents cyclic convolution carried out on the periodic signals with period $\tau_{\max}^{-1}$. As a side effect, this also suppresses noise and interfering signal components, increasing dynamic range. A thorough review of the mentioned methods in the context of UWB systems can be found in [137].

4.2.3 Signal Model of the M-Sequence Method

In this section we first present the classical signal model of the MSM and then readily extend it to the proposed signal model that employs the RD-based compression step.

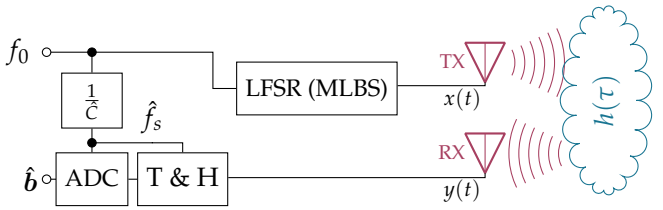


Figure 4.12 *The Track and Hold (T&H) allows the ADC to operate at a lower rate at the cost of more measurement cycles* – MSM as proposed in [138]. ■

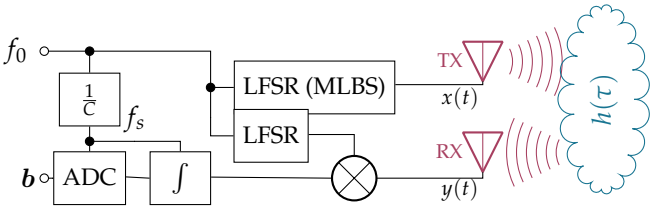


Figure 4.13 *We remove the need for a T & H by using an RD* – The architecture proposed in this work. ■

4.2.3.1 State-of-the-Art M-Sequence Method (MSM) Model

In this work, we further promote the choice for x as a periodic Maximum Length Binary Sequence (MLBS) of order k, exhibiting a period of $N = 2^k - 1$. The advantages are that these signals have a low CF close to 1 and the desirable auto-correlation properties, as introduced in Section 4.2.2. Such a signal can be generated efficiently from the system clock f_0 using a LFSR, as presented in [138], requiring only a few standard digital gates to generate $x[n]$, which denotes a sampled version of x as already motivated by means of Theorem 2.2. Figure 4.12 shows the block diagram of a measurement device employing the MSM.

The periodic excitation sequence $x[n] = x[n + k \cdot N]$ for $k \in \mathbb{N}$ is also referred to as the vector $\boldsymbol{x} \in \{+1, -1\}^N$. Here, $N \in \mathbb{N}$ denotes the period length, i.e. the minimal N such that the periodicity requirement is fulfilled. Similarly, we describe the periodic system response signal y via $\boldsymbol{y} = \boldsymbol{x} \circledast \boldsymbol{h} \in \mathbb{R}^N$. As we have already noticed in Section 3.3, the cyclic convolution operator can be defined as a circulant matrix $\boldsymbol{D} \in \{+1, -1\}^{N \times N} = \boldsymbol{\Gamma}(\boldsymbol{x})$, such that

$$\boldsymbol{y} = \boldsymbol{D} \cdot \boldsymbol{h}, \tag{4.17}$$

which is simply the discretized version of (4.16) and a specific version of

(2.2). The circulant operator $\mathbf{\Gamma}$ is introduced in Section 3.3.2.1. The cyclic discretized auto-correlation function of a periodic MLBS is almost equal to a scaled version of $\boldsymbol{e}_n$ except for a small constant offset [144], so that we have

$$\boldsymbol{x} \circledast \boldsymbol{x} = N \cdot \boldsymbol{e}_1 - \mathbf{1}_N, \tag{4.18}$$

where $\boldsymbol{e}_1$ is the first canonical unit-vector and $\mathbf{1}_N$ is a vector with all N entries equal to 1. It is now possible to estimate the IR from the system response $\boldsymbol{y}$ via

$$\tilde{\boldsymbol{h}} = \boldsymbol{D}^{\mathrm{T}} \cdot \boldsymbol{y} = \boldsymbol{D}^{\mathrm{T}} \cdot (\boldsymbol{x} \circledast \boldsymbol{h}) = \boldsymbol{x} \circledast \boldsymbol{x} \circledast \boldsymbol{h} \approx N \cdot \boldsymbol{h}. \tag{4.19}$$

Note that Equation (4.19) claims a correlation gain of factor $\approx N$, which allows the use of significantly smaller excitation signal amplitudes to achieve a certain level of SNR.

To reduce hardware complexity in the receive path induced by the high operating bandwidth, [138] proposes to employ subsampling as depicted in Figure 4.12. Using a fast T&H circuit, any time sample of y can be stored sufficiently long to convert the sample using slow, low-cost ADC circuits. A sampling clock $\hat{f}_s = f_0/C$ controls the conversion rate, which is derived from the system clock f_0 by means of an integer divide-by-C clock divider circuit. Due to subsampling, $\boldsymbol{y}$ can be collected over the course of C excitation signal periods. This effectively allows trading IR measurement rate $\hat{f}_{\mathrm{IRF}} = f_0/(C \cdot N)$ against ADC conversion rate, greatly reducing hardware component requirements.

Especially when measuring radio channels, which is a popular application of the MSM, moving objects cause the system response to be clinched/stretched in time domain. This effect, also known as the Doppler effect is tolerable, as long as the total absolute time distortion does not exceed the amount of one half sample duration $T_{\max} = 1/2f_0^{-1}$. This upper limit can also be derived from a frequency domain perspective, where an object moving through the channel at speed v relative to transmitter and receiver causes a Doppler shift in the signal spectrum. Since we need to sample the Impulse Response Function (IRF) with at least twice the rate of the maximum Doppler shift to avoid degradation of the estimate $\tilde{\boldsymbol{h}}$ in moving scenarios, the respective limit can be derived as

$$\nu = \frac{2f_0 \cdot v}{c_{\mathrm{prop}}} \quad \text{and} \quad f_{\mathrm{IRFmin}} = \frac{v_{\max} \cdot f_0}{c_{\mathrm{prop}}}, \tag{4.20}$$

where c_{prop} is the relative propagation speed in the medium (usually the speed of light, when we are concerned with common radio channel) and

f_{IRFmin} is the minimum IR measurement rate, for which moving objects of relative speeds up to $v_{\max}$ can be tolerated. If the motion limit of (4.20) is exceeded, coherent sampling is lost and the correlation gain of (4.18) diminishes by leaking to other taps of $\tilde{\boldsymbol{h}}$. As we see later, Figure 4.16 (b) in Section 4.2.5 exhibits this degradation strongly as soon as the motion limit is exceeded.

If we want to derive the sampling as in (2.1) for the T&H based IR acquisition, we need to derive the structure of the measurement matrix $\boldsymbol{\Phi}^0$, which in our case can be described as a permutation operator $\boldsymbol{\Phi}^0 \in \{0,1\}^{N\times N}$, defined as

$$\boldsymbol{\Phi}^0_{ij} = \begin{cases} 1 & \text{for} \quad j = (i \cdot C) \operatorname{Mod} N, \\ 0 & \text{else} \quad . \end{cases} \tag{4.21}$$

Since the track and hold only allows us to sample every C-th sample we have to measure so many repetitions of y such that we have collected each sample of $\boldsymbol{y}$ exactly once. However these samples occur in a different ordering such that we end up with $\boldsymbol{\Phi}^0$. Then, in the presence of additive noise $\hat{\boldsymbol{n}} \in \mathbb{R}^N$, the linear model for our observation completes to

$$\boldsymbol{b}^0 = \boldsymbol{\Phi}^0 \cdot \boldsymbol{y} = \boldsymbol{\Phi}^0 \cdot \boldsymbol{D} \cdot \boldsymbol{h} + \hat{\boldsymbol{n}} \tag{4.22}$$

and to ensure full rank of $\boldsymbol{\Phi}^0$, C must not be a factor of N. Note that again we have derived a special version of (2.3) presented in Section 2.1.3.

Since as a permutation matrix $\boldsymbol{P} = \boldsymbol{\Phi}^0$ is trivial to invert by means of $\boldsymbol{P}^{-1} = \boldsymbol{P}^{\mathrm{T}}$, (4.19) applies for estimating $\tilde{\boldsymbol{h}}$ from $\boldsymbol{b}^0 \in \mathbb{R}^N$. If this kind of subsampling is employed, it is self-evident that the MSM is not efficient in terms of signal energy utilization, since only one in C samples is actually used and we wish to improve upon this with our architecture.

Hence, to further improve robustness against noise, multiple observation vectors $\boldsymbol{b}^0$ can and often should be averaged, which in turn further trades dynamic range against measurement speed.

4.2.3.2 Proposed Compressive Architecture

The proposed hardware architecture, as depicted in Figure 4.13 (b), addresses the problem of low system signal energy usage by applying a more efficient sampling scheme. This is a similar idea to the single-pixel camera presented in Example 2.1, where we presented a sensor that incorporates spatially distributed information into a single sample. In the following we propose a similar idea, by accumulating whole time spans of y into single samples.

Under the assumption of sparsity in the IR and hence also of $\boldsymbol{h}$ only a few degrees of freedom must be determined in order to retrieve the IR [145], for which according to the theory developed in Chapter 2 already a small number of carefully designed observations is sufficient in to recover $\boldsymbol{h}$ and hence $\boldsymbol{y}$.

To accomplish this, we replace the T&H circuit by a RD structure as in [142, 141], comprised of a second LFSR sequence generator, a multiplication circuit and a short-time integrator. Note that the RD is strictly synchronous to the excitation signal generator. This way, instead of observing one entry of $\boldsymbol{y}$ once every C system clocks, a pseudo-random linear combination of C successive elements in $\boldsymbol{y}$ is measured. These pseudo-random linear samples form the observation vector $\boldsymbol{b} \in \mathbb{R}^M$. The random projection kernels are defined by the RD mixing signal $\boldsymbol{m} \in \{-1, +1\}^{C \cdot M}$, generated by the second LFSR generator as depicted in Figure 4.13 (b). Similar to (4.21) for the MSM case, the Sampling operation of the proposed architecture can be defined as

$$\boldsymbol{\Phi}_{i \bmod N, \lfloor \frac{i}{C} \rfloor} = m[i] \quad \text{for} \quad i = 0 \ldots (C \cdot M). \tag{4.23}$$

Since we now utilize large portions of the signal $\boldsymbol{y}$ in the projections to acquire $\boldsymbol{b}$, after just a few observations ($M \ll N$), $\boldsymbol{y}$ is already captured multiple times in $\boldsymbol{b}$ and reconstructing the IR is possible from these observations.

Controlling the number of observations per $\boldsymbol{b}$, gives the flexibility to adaptively set f_{IRF} independent of f_s or f_0 and thus trade measurement rate which influences the Doppler sensitivity against the treatable sparsity of the channel's IR. This dependency is a direct consequence of statements in the flavor of Remark 2.4, since there sparsity directly influences the number of necessary measurements.

4.2.4 Sparse Recovery

Under the assumption that the linear system's IR $\boldsymbol{h}$ is sparse when using $\boldsymbol{D}$ to represent $\boldsymbol{y}$, we can now for instance use the following ℓ_0-minimization problem for reconstructing $\boldsymbol{h}$ as

$$\underset{\boldsymbol{h}}{\operatorname{argmin}} \|\boldsymbol{h}\|_0 \quad \text{s.t.} \quad \|\boldsymbol{\Phi} \cdot \boldsymbol{D} \cdot \boldsymbol{h} - \boldsymbol{b}\|_2^2 \leqslant \eta, \tag{4.24}$$

for some suitably chosen $\eta > 0$, which is a specific version of (2.12) as introduced in Section 2.2.4. Now we aim at solving (4.24) approximately by the means outlined in Section 2.3.1 for which in Section 3.2.1 we presented

an efficient version of Algorithm 3.1 utilizing the matrix-free representation $\phi_{\boldsymbol{\Phi}\cdot\boldsymbol{D}}$. However we note that also FISTA could be employed here.

Similarly to the non-linear reconstruction proposed in Section 4.1, retrieving the IR from $\boldsymbol{b}$ requires significantly more computational resources than the simple correlation post-processing employed by the MSM [138]. Since the model (4.24) consists of highly structured linear mappings, significant improvements in both memory footprint and run time performance may be achieved by exploiting structure realized by the architecture from Section 3.4. Additionally, recently an implementation of OMP has been demonstrated in silicon [146], giving way to realizing the proposed architecture as high-performance system-on-chip including SSR processing.

4.2.5 Measurement Setup

In the following sections we evaluate the performance of our proposed method based on real measurements of a moving UWB radio channel. We measure the channel IRs with a radar device employing the MSM, the *IS-HAD12HS* of *Ilmsens GmbH, Germany* and two wideband horn antennas, featuring a $-10\,\mathrm{dB}$ bandwidth of 3.11 GHz. The device features an MLBS generator of order $k = 12$ and a fixed sampling clock divider of factor $S = 128$. Providing a low-jitter stable clock source of $f_0 = 9.22\,\mathrm{GHz}$ we obtain a sampling clock of $f_s = 72.03\,\mathrm{MHz}$, thereby capable of measuring $\approx 17\,590\,\mathrm{IRF/s}$ and able to resolve an IR spread of up to 444 ns.

In addition to subsampling, we add another subsampling factor of $C_{\mathrm{add}} = 32$. This results in a total subsampling factor of $C_{\mathrm{total}} = C \cdot C_{\mathrm{add}} = 4096$. Since the excitation signal period is $N = 2^{12} - 1 = 4095$, we now have the case that according to (4.21) the sampling matrix becomes the identity matrix: $\hat{\boldsymbol{\Phi}} = \boldsymbol{I}_{\mathbf{4095}}$. Then the data stream $z[n]$ is sampled at a virtual rate of $f_s = f_0/4096 \approx 2.25\,\mathrm{MHz}$. However the sample stream is now equivalent to sampling at f_0, since the effective subsampling factor is $C_{\mathrm{eff}} = C_{\mathrm{total}} \bmod N = 1$, and reordering the samples can be omitted. The so produced data stream $z[n]$ now serves as the virtual system response signal $y[n]$ for both methods, as shown in Figures 4.12 and 4.13 (see Section 4.2.3). This allows to evaluate the performance of both methods synthetically in software on a common realistic data stream $z[n]$. Choosing C for the proposed method such that it is a factor of N is highly advisable, since it then allows to use the same $\boldsymbol{\Phi}$ for every reconstruction of (4.24). This is crucial in maintaining comparable reconstruction performance independent of currently measurement period. This also

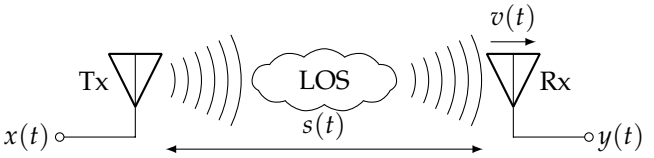


Figure 4.14 *We consider a non-static Single Input Single Output (SISO) measurement setup.* – Measurement scenario with a strong LOS and time variant receiver and sender distance s and the respective rate of change v ■

gives us the ability of sweeping Compression Ratio (CR), since for every additional repetition of the excitation signal we then gather $\frac{N}{C}$ additional observations. Overall, this leaves us with the problem that we cannot use the same C both for MSM and the proposed compressive scheme, since the MSM requires that C must not be a factor of N in order to maintain proper sampling.

To be comparable to the original IR measurement device, for the proposed compression method we select C in the vicinity of 128, which yields the choice of $C = 117$, giving an extra $\Delta M = 4095/117 = 35$ observations in $\boldsymbol{b}$ for every additional excitation signal repetition. This allows to sweep the compression factor beginning from $\approx 2.56\,\%$. Due to the constraint of the MSM we decide to give the MSM a slight advantage by choosing $C = 116$, hence collection marginally more samples.

4.2.5.1 Time-variant Line of Sight (LOS) Scenario with Multipath

Given these parameters, (4.20) yields a movement limit $\hat{v}_{\max} = 0.16\,\mathrm{m/s}$ for the MSM. Setting the highest compression rate for the proposed architecture, yields $v_{\max} = 5.93\,\mathrm{m/s}$. Figure 4.14 shows the measurement scenario, where the RX antenna is moved back and forth from the TX antenna, with $|v_{max}| \approx 1\,\mathrm{m/s}$. The antenna distance $s(t)$ corresponds directly to the peak delay $\tau_{\mathrm{LOS}}(t)$ in the channel IR $h(\tau) = \omega_0 \cdot \delta(t - \tau_{\mathrm{LOS}}(t))$, where $\omega_0(t)$ denotes the LOS intensity.

The measurement was conducted in a laboratory room, filled with large amounts of structures and objects, to produce uncontrolled multipath components, which can be seen in Figure 4.16(a). By also collecting energy from the WiFi bands during the acquisition of the measurements a slight model mismatch due to RF interference was introduced. However, we expect both methods to measure the channel IR properly during

immobility and to see the MSM fail during phases of movement due to a Doppler shift that exceeds the theoretical limit.

4.2.5.2 Matrix-free Representation

Essentially, one has two possibilities to implement $\boldsymbol{D}$ in a matrix-free fashion. Both of these we compare to the conventional dense presentation in Figure 4.15.

The first option is to calculate the sequence $\boldsymbol{d} \in \mathbb{C}^N$ that is generated by the LFSR and then use (3.9) for the matrix $\boldsymbol{D} = \boldsymbol{\Gamma}(\boldsymbol{d})$ to derive the algorithm for $\phi_{\boldsymbol{D}}$ and $\beta_{\boldsymbol{D}}$.

Alternatively, since we are employing an LFSR to generate $x[n]$, this allows to compute the convolution operator $\boldsymbol{D}$ even more efficiently using the Fast Hadamard Transform [76] as outlined in [147], which is exploited in `fastmat` by the `fastmat.LFSRCirculant` class. The key idea is to factor $\boldsymbol{D}$ as

$$\boldsymbol{D} = \boldsymbol{\Gamma}(\boldsymbol{d}) = \boldsymbol{P}_1 \cdot \boldsymbol{H} \cdot \boldsymbol{P}_2 \tag{4.25}$$

for two permutation matrices $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ depending on $\boldsymbol{d}$. Further $\boldsymbol{H}$ is the so called Hadamard matrix introduced in Section 3.3.3, which has a fast implementation for $\phi_{\boldsymbol{H}}$ by means of the FHT. Naturally, $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ have efficient matrix-free implementations, since the input vectors only have to be reordered in memory.

We compare both implementations based on (4.25) and (3.9) in Figure 4.15. Here, we can observe the usual behavior. The dense representation is very efficient for small system sizes, but already for moderates sizes of $N \approx 2^8$ the matrix-free implementations become more efficient. Among these two, the FHT-based algorithm performs slightly better, but both expose a similar asymptotic behavior.

4.2.5.3 Simulation and Measurement Results

We compare three systems defined according to Section 4.2.5, which are parameterized such that their hardware implementation effort is comparable:

systemA: Ground truth (according to MSM of Section 4.2.3.1) with $C = 1$ ($\hat{f}_s = 2.25\,\text{MHz}$), yielding $\approx 550\,\text{IRF/s}$.

systemB: The conventional MSM with $C = 116$ ($\hat{f}_s = 19.40\,\text{kHz}$), yielding approximately $4.74\,\text{IRF/s}$.

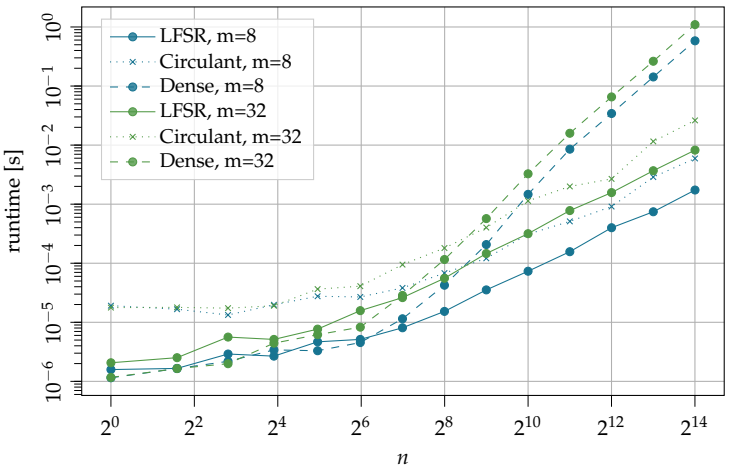


Figure 4.15 *Circulant Matrices originating from linear feedback shift registers have a fast matrix-free representation* – Comparison of the runtime between the dense, the `fastmat.Circulant`-based matrix-free implementation and the `fastmat.LFSRCirculant`-based matrix-free implementation for $\boldsymbol{x} \in \mathbb{C}^{n \times m}$. ■

systemC: The proposed compressive architecture with $C = 117$ ($f_s = 19.20\,\text{kHz}$), yielding $\approx 36.60\,\text{IRF/s}$ for $m = 525$ such that we have a compression rate of 12.82 %.

In the sparse recovery case, $K = 41$ components, which assumes a sparsity $\approx 1\,\%$ as estimated from $\tilde{\boldsymbol{h}}$ of systemA, were recovered using OMP. The columns of $\boldsymbol{D}$ are approximately linearly independent ($\mu(\boldsymbol{D}) \approx \frac{1}{N} \approx 2.44 \times 10^{-4}$) and although the pseudo-random sampling matrix $\boldsymbol{\Phi}$ exhibits high sparsity, its mutual coherence was determined as $\mu(\Phi) \approx 1.24 \times 10^{-2}$. However, motivated by Theorem 2.4, crucial to successful recovery is the coherence of the linear system model matrix $\boldsymbol{\Psi} = \boldsymbol{\Phi} \cdot \boldsymbol{D}$, which in our case was found to be $\mu(\boldsymbol{A}) \approx 2.10 \times 10^{-6}$.

Figure 4.16 shows results of this evaluation over measurement time t, plotted along the x-axis of the graphs. In plot (a) the section up to 27.10 ns of the ground-truth radargram of systemA is shown. Plots (b) and (c) show the radargrams of systemB and systemC, which in our case are setup such that both exhibit almost identical f_s and hardware complexity. Plot (e) depicts the total energy of the IR for any given time t for the three systems. Once the relative motion $v(t)$ exceeds $v_{\max}$ (i.e., during the transitions), it can be seen that the IR collected by systemB possesses much lower energy compared to systemA and systemC. Furthermore, the

inspection of the radargram of systemB also shows the leakage of energy into wrong delay times due to aliasing in time-domain. An expected outcome of the loss of correlation coherence due to too high Doppler-shift. Since systemC does not exceed $v_{\max}$ due to its compressive sampling approach, plots (c) and (e) reveal that coherence is maintained.

Plot (d) shows the trajectory of the sender and receiver distance $s(t)$ of the measured LOS scenario and their relative movement speed $v(t) = \dot{s}(t)$, which determines the Doppler properties according to (4.20) for the different systems. To derive $s(t)$ directly from the measurement data, we use the outputs of systemA where we determine the distance of sender and receiver by means of a peak search at each observation time t. In order to be less sensitive to local oscillations of these peak searches, we low-pass filter these peak locations along t to yield a smooth estimate of $s(t)$ from the measured data of systemA. In a similar fashion to $s(t)$ the signal peak amplitude $a(t)$ can be derived directly from the output of systemA. Combining both it is possible to define a Reconstruction Error Metric (REM) ϵ considering both delay- and amplitude errors:

$$\epsilon_t = (s_X(t) - s(t)) + \left| a_X(t) \cdot a(t)^{-1} - 1 \right| \tag{4.26}$$

where X is to be replaced by the system indicator.

Plot (f) concludes with a phase diagram showing ϵ_t of (4.26) for systemC at different choices of the compression factor. The best system was identified by a total error metric $\operatorname{argmin}_{cr} \int \epsilon_{cr}(t)dt$, and defined as systemC as well as indicated in the plot by the green dashed line. The solid green line indicates the Doppler limit of (4.20) in relation to the chosen compression factor. Once the Doppler limit is exceeded, the proposed method also fails to correctly reconstruct the scenario. Also, the reconstruction is more robust for higher signal strength as is indicated by plot (f) around $t = 4\,\mathrm{s}$, where reliably a lower CR can be chosen than for the borders of the plot.

4.2.6 Conclusion

Compared to previous architectures based on PN sequences, the proposed architecture makes better use of the received signal's energy, hence allows reconstructing the channel IR from fewer observations. While maintaining the low complexity of the MSM hardware frontends [138], adaptive control of key measurement parameters is added, which allows to trade IR measurement rate against supported Doppler range. Considering advances in integrated technology [146, 148], our proposed method presents

the prospect of highly-integrated and -agile IR measurement systems of manageable hardware complexity. This way we successfully addressed AOI-H.

On the algorithmic side we have demonstrated that the employed measurement setup allows for efficient signal recovery based on matrix-free implementations, as it readily leverages the methods developed in Chapter 3, hence we contributed to AOI-R.

Our simulative evaluation, which is based on real channel measurement data, has shown that for improving the recovery performance according to CS theory, good conditioning of the system matrix $\boldsymbol{\Psi} = \boldsymbol{\Phi} \cdot \boldsymbol{D}$ must be further investigated in the spirit of the results in Sections 2.2 and 2.4. Aside from integrated circuit demonstration of the architecture, investigating methods for selecting the RD mixing signal and architectural parameters m, S in correspondence to the excitation signal were identified as important matters for further research.

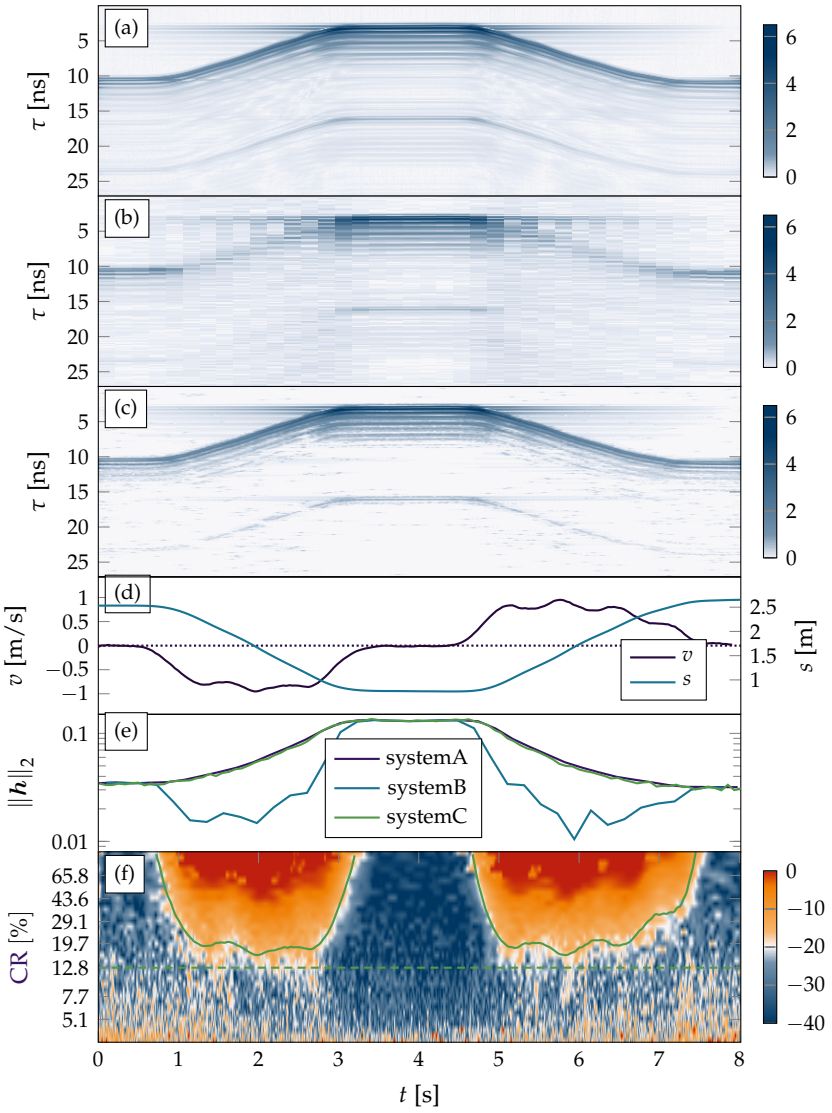


Figure 4.16 *The proposed RD based UWB system can can capture more energy of the impulse response* $\boldsymbol{h}$ *and is more robust against Doppler shifts.* – The plots contain:
(a) systemA - Ground truth acc. to MSM (Section 4.2.3.1) with $C = 1$
(b) systemB - Results for the MSM (Section 4.2.3.1) with $C = 116$
(c) systemC - The proposed method (Section 4.2.3.2) with $C = 117$
(d) Trajectory of moving LOS scenario according to Section 4.2.5.1
(e) Signal energy in $\boldsymbol{h}$ for case (a), estimated $\tilde{\boldsymbol{h}}$ for cases (b) and (c)
(f) REM ϵ for systemC at different CR choices in [dB].
Evalation results (a) to (e) show 27.10 ns of a total of 444 ns ■

4.3 Summary

Both of the previous sections show that CS can either help to reduce the amount of collected data when applied carefully as we demonstrate in the US-NDT application or the same amount of acquired data can lead to substantial improvement in terms of stability and robustness as outlined in the CS-UWB scenario. Hence, in both cases a suitable application of the principles of CS allows a more flexible design of the sensing stage and hence allows us to collect more information about the signals of interest.

In both cases the matrix-free representation allows significant improvements in terms of computational efficiency. When confronted with the high dimensionality of the data and the parameter space as in the US-NDT scenario one cannot circumvent the application of these methods without substantially degrading the amount of data that can be processed or the attainable resolution for the parameter estimation task.

Finally, we have seen that classical analysis tools like the CRB, as studied in Section 4.1, still allow to infer the performance of a CS system. In this sense, CS pipelines still behave similarly to other system employed for parameter estimation. The same holds true for the robustness to Doppler-shifts in the previous Section 4.2, where the well known physically motivated bounds on the non-stationarity can also be applied to a CS system.

These two applications of the grid-based CS theory conclude our study of this subject. In the next chapter we introduce an extension of CS, which allows to apply compressive measurement in even more general and hence more realistic estimation tasks.

Chapter 5

Grid-Free Compressed Sensing

> "All we have to do is decide what to do with the time that is given to us."
>
> *Gandalf*

Up until this point, we have treated the setting which originally was introduced as CS in the form of (2.3). If framed correctly, this model for compressive measurements can be used to rephrase parameter estimation problems where we can expect sparsity in the parameter domain. In Example 2.2 and Chapter 4 we have encountered several examples for this sparsity-based parameter estimation. However, this compressive modeling step forces us to assume a discretization in the parameter domain, which imposes an unnatural and as we will see sometimes detrimental restriction on the parameter space that can be used for estimation. The following sections first outline the ramifications of this imposed grid on the task of parameter estimation. Afterwards, we show how to circumvent this while still allowing the notion of sparsity to play a role during the reconstruction step.

Contributions

During the development of this chapter about grid-free CS we propose a very general method for line-spectral estimation in Section 5.2.6, which is able to cope with an arbitrary-dimensional parameter space, compressive measurements and multiple snapshots. This way, we are contributing to AOI-R. Additionally, in Section 5.2.3 we also show how to apply this general sparse recovery scheme to the task of DoA estimation with arbitrary and hence realistic antenna geometries, which addresses AOI-M and AOI-R as well. Next, in Section 5.3 we propose a flexible optimization strategy that is motivated by machine learning techniques that allows to retrieve measurement matrices in the grid-free CS setting for very general parametric models. By doing so, we address AOI-C. As such, these contributions are based on the publications [O18, O5, O4]. Finally, we show that both the measurement optimization procedure and the reconstruction approach can be combined to result in a well performing grid-free CS pipeline.

We start by presenting a simple motivating scenario in Example 5.1 and we outline some existing approaches to tackle the estimation process, where not all of them treat sparsity as the driving factor.

5.1 Introduction and Motivation

Very soon after the advent of CS, research actively explored the possibility to use this more general sampling theory to extract parameters of interest from all sorts of different data [149, 150, 49]. If we reconsider Equation (2.3), which reads as

$$\boldsymbol{z} = \boldsymbol{\Psi}\boldsymbol{x} = \boldsymbol{\Phi}\boldsymbol{y} \in \mathbb{C}^m$$

we not only want to accurately reconstruct the signal $\boldsymbol{y} \in \mathbb{C}^N$, but also the correct sparse representation $\boldsymbol{x} \in \mathbb{C}^M$, since in the parameter estimation setting, the location of the non-zero entries encodes the information of interest, see for instance Example 2.2. There, the non-zeros entries' positions encode the estimated delay. However, due to the nature of finite-dimensional vectors $\boldsymbol{x}$, we can also only recover a finite number of unknown parameter configurations. However, in reality the unknown parameters usually occupy a continuous domain and we are merely sampling this domain with a hopefully dense enough grid in order to capture the behavior of our system of interest. But, however dense this sampling

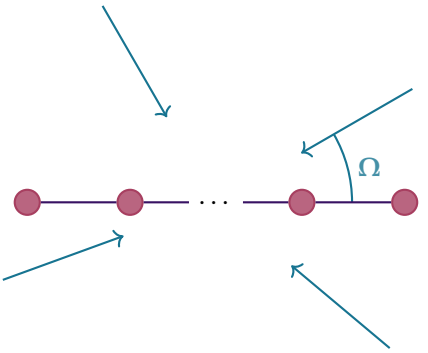


Figure 5.1 *Schematic for the DoA estimation problem.* – An array of antenna elements where plane waves with unknown spherical angle Ω impinge from the antenna's far-field. ■

might be, there will be an inherent estimation bias and even cases where reconstruction fails completely.

To have a clearer image of the problem at hand, we introduce a typical DoA estimation scenario in Example 5.1.

Example 5.1 (DoA Estimation)**.** Much like sampling in the frequency domain allows to infer the time shifts of a known waveform by means of estimating linear-phase behavior in Example 2.2, one can also use spatial sampling by means of an array of antenna elements to extract propagation directions of electromagnetic plane waves. In Figure 5.1 we show a highly simplified scenario where a Uniform Linear Array (ULA) is collecting information about $S = 4$ impinging paths.

In order to derive a data model, we assume we have access to a function $\boldsymbol{a} : \mathbb{R}^2 \to \mathbb{C}^P$, which is called the array manifold, since $\boldsymbol{a}(\vartheta, \varphi) \in \mathbb{C}^N$ models the response of the P-element antenna array to a plane wave impinging from elevation angle ϑ and azimuth angle φ with (for instance) vertical polarization. Due to the superposition principle of electromagnetic waves, we can assume that our observation that is output by the antenna ports can be modeled as

$$\boldsymbol{y} = \sum_{s=1}^{S} \gamma_s \cdot \boldsymbol{a}(\boldsymbol{\Omega}_s),$$

where $\boldsymbol{\Omega}_s = (\vartheta_s, \varphi_s)$ are the unknown spherical angles and S is assumed to be small enough such that the array with P elements is able to resolve these sources. Hence, we model the number of impinging waves as the

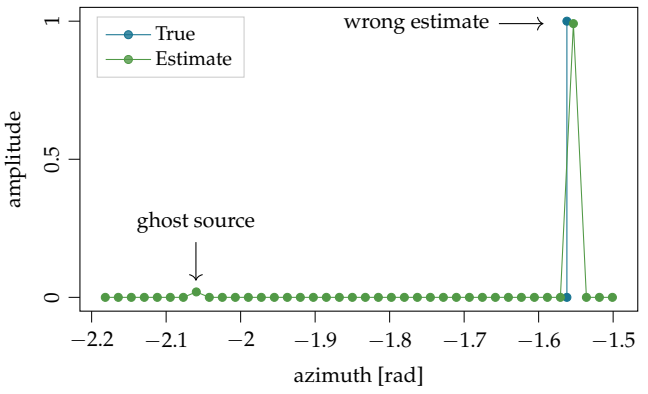


Figure 5.2 *Grid mismatch introduces ghost sources.* – OMP reconstruction (green) to extract the true DoA (blue), which does not reside on the imposed parameter grid. ■

sparsity S in this scenario. Additionally, by following [151, 8, 7, 152, 150] and [149] we employ an analog precoding network, which is represented by the matrix $\boldsymbol{\Xi} \in \mathbb{C}^{m \times P}$ and we finally have

$$\boldsymbol{z} = \boldsymbol{\Xi}\boldsymbol{y} + \boldsymbol{n} = \sum_{s=1}^{S} \gamma_s \cdot \boldsymbol{\Xi}\boldsymbol{a}(\boldsymbol{\Omega}_s) + \boldsymbol{n} \in \mathbb{C}^m$$

where $\boldsymbol{n} \in \mathbb{C}^m$ represents additive measurement noise. This precoding is applied to reduce the necessary effort in terms of hardware, since fewer (now m compared to N) amplification and digitization chains are necessary to process the incoming signals, while the promise of CS is that the aperture and hence the hereby achievable resolution are not substantially inflicted negatively by the reduction of data represented by $\boldsymbol{\Xi}$.

The goal now is to estimate the unknown tuples (θ_s, φ_s) and the so-called path-weights $\gamma_s \in \mathbb{C}$ from the observation $\boldsymbol{z}$ under the assumption that the model we posed is valid and we have perfect knowledge of $\boldsymbol{a}$. ■

To solve this with the previously developed methods, in spirit of Example 2.2, one again imposes a $2D$ grid for the unknown spherical angles (θ, φ) and forms the matrix

$$\boldsymbol{A} = \left[\boldsymbol{a}(\theta_1, \varphi_1), \ldots, \boldsymbol{a}(\theta_i, \varphi_j), \ldots, \boldsymbol{a}(\theta_{G_1}, \varphi_{G_2})\right] \in \mathbb{C}^{N \times G_1 \cdot G_2}$$

as the sparsifying dictionary and arrives again at a sparse model that reads as

$$\boldsymbol{z} = \boldsymbol{\Xi}\boldsymbol{A}\boldsymbol{\gamma} \in \mathbb{C}^m,$$

where $\boldsymbol{\gamma} \in \mathbb{C}^{G_1 \cdot G_2}$ is assumed to be S-sparse. However, since in reality the angles do not reside on this finite grid, and hence cannot be correctly represented by $\boldsymbol{A}$, one introduces an unavoidable model mismatch. To demonstrate this effect, in Figure 5.2 we setup a DoA estimation scenario, similar to [153], where we run OMP to recover a single source's unknown azimuth angle from which an antenna collects its signal. Here, the incidence angle is not on the assumed grid and we notice the introduction of an additional ghost source to compensate for the inherent model mismatch.

This observation is still valid, even if one introduces a denser grid for the azimuth angles. Additionally, one then has to cope with the higher computational effort and more importantly, the effect can only be reduced, but not mitigated completely.

This sparked the development of techniques, like those in presented in [153], where first clustering of sources and then correction is performed to estimate those sources not resided on the grid more reliably, where the correction might estimate a source angle not on the original grid. However, this approach is still bound to a grid-based reconstruction scheme, like ℓ_1-minimization or OMP and it assumes that the ghost sources only appear close to the true source location. However, we see in Figure 5.2 that this does not need to be the case necessarily.

Hence, investigations started [154, 155] to develop approaches that reconstruct unknown parameters in a continuous manner directly without resorting to a grid in the first place, while still maintaining the notions of sparsity and allowing for compressive observations.

5.1.1 Other Methods

The task to extract parameters from a given measurement has been addressed and considered from various perspectives [26]. Conventionally, one coins these methods methods as High Resolution Parameter Estimation (HRPE) methods, since usually one strives to beat some sort of Rayleigh-limit as outlined in [156] for the DoA, i.e. array processing, case. Often one invokes some kind of parametric model as we also did in Chapter 4 in order to derive the nature of the dictionary $\boldsymbol{A}$ in (2.3). Then this parametric model allows to infer on the underlying parameters of interest by (approximately) solving a suitably regularized inverse problem or exploiting other geometric properties of the scenario at hand.

Other Grid-free Sparse Recovery Approaches

The research field that explores the possibility to use grid-free sparse recovery methods for DoA estimation has recently gained a lot of interest. In [157] the authors use a coprime array interpolation method to which they adapt the framework of Atomic Norm Minimization (ANM), which we introduce in Section 5.2.1. However, the case of non-uniform, hence realistic antenna arrays is not considered there. Also no compression step is employed to reduce the amount of data captured.

In [158] the authors propose a spatial sub-selection from data captured by a ULA. However, the proposed method still heavily depends on the assumed array geometry and it cannot deal with more realistic setups. Also, the authors only propose simple sub-selection of the antennas' output instead of more involved compression schemes.

The authors of [159] consider the estimation of angles and receive delays from spatial and frequency data. However, they separate the parameter domain into distinct domains, hence the parameters are not estimated jointly, which ultimately results in a pairing problem when trying to correctly match the estimated quantities to the impinging waves.

In [158] a channel estimation technique based on ANM is proposed, where the authors only treat the two dimensional DoA estimation case, which is valid for two ULAs at the transmit and receive side.

The publication coming closest to our proposed methods is [160], where the authors present a similar approach as we do in Section 5.2.2, where the antennas beampattern is approximated by a finite Fourier series for the one dimensional DoA estimation task from a single snaptshot. However, in our developed theory, we present a more general setting, where an arbitrary dimensional estimation task from multiple compressed snapshots is treatable. ■

Richter Maximization Approach

In [161] the authors propose the very flexible and highly optimized estimation framework Richter Maximization Approach (RIMAX) that allows to cope with measurements following a double directional radio-channel model as in [162]. The approach taken by this framework is implementing an approximate maximum-likelihood estimator that can incorporate arbitrary antenna geometries and also is able to estimate signal components that are not represented by the signal model itself. Additionally, a heuristic for model order selection is applied, in order to estimate the model complexity. However, due to the nature of the algorithmic approach one

cannot formulate any theoretical guarantees for the convergence and correctness of the algorithm. Also, no estimates about the statistical efficiency of the algorithm can be derived. However, the practical performance of this framework can be verified for a large number of empirical studies. Finally, one can easily extend the assumed model to the case where one uses analog precoding as a compression step, as introduced in Example 5.1. ■

Subspace Methods

Another set of well understood methods are so-called subspace methods [156] where the two most prominent members are Estimation of Signal Parameters via Rotational Invariance Techniques (ESPRIT) as initially proposed in [163] and Multiple Signal Classification (MUSIC), originally formulated in [164]. Both methods construct suitable subspaces which implicitly classify the received signal into measurement noise and the actual signal. Then, one can use geometric arguments to extract the parameters of interest from the signal subspaces. In these cases one can even infer the estimation performance [165], hence give some theoretical insight in the behavior of the estimation process. Unfortunately, one has to assume that some invariances are fulfilled, which often is the case only for very regular and idealized array geometries and patterns, again leading to a model-mismatch and hence an estimation bias. Also, these methods are not directly applicable to compressed data as in Example 5.1. ■

Sparse Bayesian Learning

As an extension to the well known Support Vector Machine (SVM) as developed by [166] the authors in [167] presented the Relevance Vector Machine (RVM), which naturally results in sparse representations based on a given parametric model. This approach can also be implemented more efficiently using the so-called Fast Relevance Vector Machine (FRVM) as proposed by [168]. In [169] the authors use a first order Taylor approximation in order to estimate the grid-offset directly in a sparse parametric model which is used in DoA estimation. These approaches allow to incorporate very general antenna models and can also cope with the analog precoding in Example 5.1 while also allowing statements about the convergence of some of the usually employed algorithms. ■

5.1.2 Generalized Line Spectral Estimation via ADMM

Despite all these already existing efforts, we are going to take a different route and present the body work initially jump started by [154] and [155] that present statements in the spirit of the previously developed grid-based CS, but without the assumption of a fixed and discrete parameter grid. Again, we ask the same questions and propose heavily intertwined (partial) answers as before:

- ↠ How should we conduct the compression step?
- ↠ How should we recover the parameters of interest?
- ↠ Under which conditions can we guarantee unique solutions of the estimation problem?

These problems are addressed in the following sections of this thesis and a summarized answer is given Theorem 5.2. We first introduce the concept of ANM in a very general setting in Section 5.2.1 as an abstract framework for sparsity based estimation from compressed data. Afterwards, we show how in the case of line spectral estimation the dual problem to ANM can be used to get an optimization problem that can theoretically be solved efficiently. For this highly structured optimization problem, we present a specialized and iterative Alternating Directions of Multipliers Method (ADMM) approach, which can deal with very general compression and reconstruction setups.

5.2 Reconstruction Methods

In the following, we present the theory of grid-free CS based on ANM as an extension and generalization of grid-based CS based on ℓ_1-minimization.

5.2.1 Atomic Norm Minimization

For the introduction of ANM we mostly follow the story line of [170]. Let $\mathcal{A} \subset \mathbb{C}^N$ be a subset of a finite dimensional complex valued vector space. The model for the signal we are going to consider is of the form

$$\boldsymbol{y} = \sum_{s=1}^{S} c_s \cdot \boldsymbol{a}_s \in \mathbb{C}^N, \tag{5.1}$$

where the so-called amplitudes satisfy $c_s \in \mathbb{R}^+$ and the atoms $\boldsymbol{a}_s \in \mathcal{A} \subset \mathbb{C}^N$. As such, we assume *sparsity* of $\boldsymbol{y}$ with respect to the set $\mathcal{A}$,

by assuming S to be relatively small compared to the dimension of the space that $\mathcal{A}$ is embedded in. To see how this generalizes our previous notion of sparsity, assume that $\mathcal{E} = \{-\boldsymbol{e}_1, +\boldsymbol{e}_1, \dots, -\boldsymbol{e}_M, +\boldsymbol{e}_M\}$ and we have a matrix $\boldsymbol{A} \in \mathbb{C}^{N \times M}$. If we now set $\mathcal{A}$ as the image of $\mathcal{E}$ under $\boldsymbol{A}$ ($\mathcal{A} = \boldsymbol{A} \cdot \mathcal{E}$), we directly recover the sparse representation as in (2.2). Many variants of the set $\mathcal{A}$ have been studied so far, i.e. sparse or low-rank matrices [171], permutation matrices [172] and many more, see again [170] for an extensive list.

In order to proceed we need the notion of a convex combination. Let $\boldsymbol{a}_1, \dots, \boldsymbol{a}_n$ be a subset of $\mathbb{C}^N$. Then, a convex combination is defined as

$$\boldsymbol{y} = c_1 \cdot \boldsymbol{a}_1 + \cdots + c_n \cdot \boldsymbol{a}_n,$$

where each $c_i \geqslant 0$ and $\sum_i c_i = 1$ and $n < \infty$, which renders convex combinations to be finite sums. Then, we define the convex hull $\mathrm{conv}(\mathcal{A})$ as the set of all convex combinations that can be formed among elements in the set $\mathcal{A}$. Another definition of the convex hull is to set it as the intersections of all convex sets that contain $\mathcal{A}$. This allows us to define the atomic norm $\|\cdot\|_{\mathcal{A}} : \mathbb{C}^N \to \mathbb{R}_0^+$ such that

$$\boldsymbol{y} \mapsto \|\boldsymbol{y}\|_{\mathcal{A}} = \inf_{t \geqslant 0} \{\boldsymbol{y} \in t \cdot \mathrm{conv}(\mathcal{A})\}.$$

The intuition is that the atomic norm quantifies by how much we have to inflate the convex set $\mathrm{conv}(\mathcal{A})$ such that $\boldsymbol{y}$ is contained in it. Interestingly, we can alternatively calculate

$$\|\boldsymbol{y}\|_{\mathcal{A}} = \inf \left\{ \sum_{\boldsymbol{a} \in \mathcal{A}} c_{\boldsymbol{a}} : \boldsymbol{y} = \sum_{\boldsymbol{a} \in \mathcal{A}} c_{\boldsymbol{a}} \cdot \boldsymbol{a},\ c_{\boldsymbol{a}} \geqslant 0,\ \forall \boldsymbol{a} \in \mathcal{A} \right\}. \tag{5.2}$$

Note that in this general case, the evaluation of $\|\cdot\|_{\mathcal{A}}$ in itself already requires the solution of an optimization problem. As in Section 2.1.2 in (2.1), we assume we have access to $\boldsymbol{y}$ not directly but in the form of noisy linear measurements

$$\boldsymbol{z} = \boldsymbol{\Phi} \cdot \boldsymbol{y} + \boldsymbol{n} \in \mathbb{C}^m \tag{5.3}$$

for the measurement matrix $\boldsymbol{\Phi} \in \mathbb{C}^{m \times N}$, where $m < N$ such that we are *compressing* the signal to fewer observations and vector $\boldsymbol{n} \in \mathbb{C}^m$ representing additive noise.

As we see from (5.2), the less complex $\boldsymbol{y}$ is in terms of $\mathcal{A}$, the smaller $\|\boldsymbol{y}\|_{\mathcal{A}}$ is. Hence, in order to recover $\boldsymbol{y}$ from $\boldsymbol{z}$ we use the following optimization problem [170]

$$\min_{\boldsymbol{y} \in \mathbb{C}^N} \|\boldsymbol{y}\|_{\mathcal{A}} \quad \text{s.t.} \quad \|\boldsymbol{\Phi}\boldsymbol{y} - \boldsymbol{z}\|_2^2 \leqslant \eta, \tag{5.4}$$

in order to find a simple representation of $\boldsymbol{y}$ in terms of $\boldsymbol{A}$. This can be regarded as a generalized version of (2.13). Since if we have a solution $\boldsymbol{x}$ to (2.13) we can directly infer that $\|\boldsymbol{A} \cdot \boldsymbol{x}\|_{\mathcal{A}} = \|\boldsymbol{x}\|_1$. Finally, the process of solving the problem in (5.4) is what is called Atomic Norm Minimization.

Although we now already have theoretically derived (5.4) as a sparsity promoting reconstruction method, it is not as useful as (2.13), since without specifying $\mathcal{A}$, the specific methods for the optimization cannot be constructed. For different sets of $\mathcal{A}$ the involved methods might turn out to be very different. For example, if $\mathcal{A}$ is the set of unit norm one-sparse vectors, we can make use of the FISTA algorithm presented in Section 2.3.2. In the case when $\mathcal{A}$ is the set of rank-1 matrices, it corresponds to nuclear norm minimization [173] for which efficient algorithms exist as well. Hence, it is necessary to precisely specify the set $\mathcal{A}$ before we are in the position to derive means of solving the problem in (5.4).

5.2.2 ANM for DoA Estimation

For DoA estimation, the array manifold $\boldsymbol{a} : \mathbb{R}^2 \to \mathbb{C}^P$ is the key component to derive a realistic data model for a P-element antenna array. Here, we consider the narrow-band case, which means the waveforms impinging on the array have a small relative bandwidth in terms of the employed carrier frequency. This allows us to treat the array-manifold as if it were frequency independent. Additionally, we only consider one polarization for the sake of simplicity. Although note that the methods can easily be extended to two polarizations.

The function $\boldsymbol{a}$ models the far-field antenna arrays response depending on the angle of incidence of the impinging wave. Interestingly, due to the natural periodicity of antenna beampatterns, one can use the so-called Effective Aperture Distribution Function (EADF) [174, O7] to express it as a two-dimensional Fourier series, via

$$\begin{aligned}\boldsymbol{a}(\vartheta, \varphi)_p \approx \sum_{\ell_1=1}^{L_1} \sum_{\ell_2=1}^{L_1} g_{\ell_1,\ell_2,p} \cdot \exp\left[\jmath 2\pi \frac{\ell_1 - (L_1+1)/2}{(L_1+1)/2}\vartheta\right] \\ \cdot \exp\left[\jmath 2\pi \frac{\ell_2 - (L_2+1)/2}{(L_2+1)/2}\varphi\right],\end{aligned}$$

for odd L_1 and L_2 to express the arrays response at the p-th port. This holds true, since the antenna beampatterns are naturally band-limited. The $g_{\ell_1,\ell_2,p} \in \mathbb{C}$ are the so-called EADF (or aperture) coefficients and can be computed efficiently from samples of the function $\boldsymbol{a}$. A visualization for a measured antenna is given in Figure 5.3.

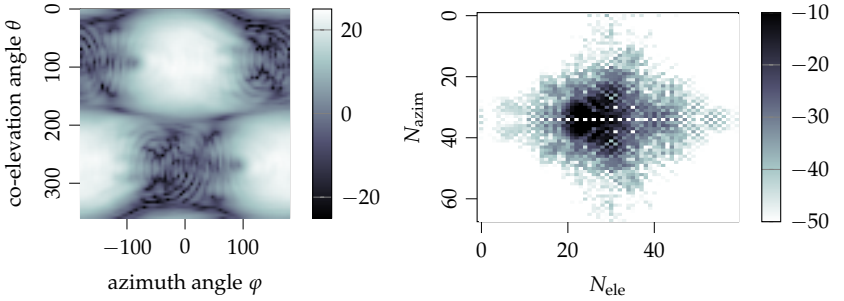


Figure 5.3 *Fourier series of beampatterns allow efficient representations.* – Measured and periodified beampattern slice $\boldsymbol{a}(\vartheta, \varphi)_p$ of a patch antenna (left), the corresponding 2D (61×69) Fourier coefficients $g_{\ell_1,\ell_2,p}$ (right). ■

In order to arrive at a model as in (5.1), we define the function $\boldsymbol{f}_L : \mathbb{R} \to \mathbb{C}^L$ for odd L as

$$\theta \to \boldsymbol{f}_L(\theta) = \left[\exp\left(\jmath 2\pi \frac{\ell - (L+1)/2}{(L+1)/2}\theta\right)\right]_{\ell=1}^{L}, \tag{5.5}$$

and the matrix

$$\boldsymbol{\Gamma} = [\gamma_{\ell_1 \cdot L_2 + \ell_2, p}] = g_{\ell_1,\ell_2,p},$$

such that we have $\boldsymbol{\Gamma} \in \mathbb{C}^{P \times L_1 \cdot L_2}$. Then, we can write

$$\boldsymbol{a}(\boldsymbol{\Omega}) = \boldsymbol{a}(\vartheta, \varphi) = \boldsymbol{\Gamma} \cdot (\boldsymbol{f}_{L_1}(\vartheta) \diamond \boldsymbol{f}_{L_2}(\varphi)) \in \mathbb{C}^P,$$

where $\diamond$ denotes the Khatri-Rao (or columnwise Kronecker-product) product. Hence, we can describe the antenna beampattern by a linear combination of sampled complex harmonic functions, whose frequency depend on the incidence angles. Now, if we reconsider Example 5.1, we can finally present the single-snapshot model for DoA estimation as

$$\boldsymbol{z} = \boldsymbol{\Xi} \cdot \boldsymbol{\Gamma} \cdot \sum_{s=1}^{S} c_s \cdot (\boldsymbol{f}_{L_1}(\vartheta_s) \diamond \boldsymbol{f}_{L_2}(\varphi_s)), \tag{5.6}$$

where $\boldsymbol{\Xi} \in \mathbb{C}^{m \times P}$ constitutes the part of $\boldsymbol{\Phi} = \boldsymbol{\Xi} \cdot \boldsymbol{\Gamma} \in C^{m \times N}$ (with $N = L_1 \cdot L_2$) that represents the analog precoding network and can be chosen freely. The key observation here is that the extraction of the unknown angles ϑ_s and φ_s constitutes a so-called generalized $2D$ line spectral estimation problem as presented in [175].

To reformulate it a little further, we reformulate (5.6) by means of the definition

$$\boldsymbol{f}_{L_1,\dots,L_R}(\theta_1,\dots,\theta_R) = \boldsymbol{f}_{L_1}(\theta_1) \diamond \cdots \diamond \boldsymbol{f}_{L_R}(\theta_R),$$

where we can reformulate this Khatri-Rao product to

$$[\boldsymbol{f}_{L_1}(\theta_1) \diamond \cdots \diamond \boldsymbol{f}_{L_R}(\theta_R)]_{\ell_1,\dots,\ell_R} = \exp\left(\jmath 2\pi \sum_{r=1}^{R} \frac{\ell_r}{L_r}\theta_r\right).$$

Then, (5.6) can be considered a special version of

$$\boldsymbol{z} = \boldsymbol{\Phi} \cdot \boldsymbol{y} = \boldsymbol{\Phi} \cdot \sum_{s=1}^{S} c_s \cdot \boldsymbol{f}_{L_1,\dots,L_R}(\theta_{1,s},\dots,\theta_{R,s}), \tag{5.7}$$

if we set $R = 2$ and $\theta_1 = \vartheta$ and $\theta_2 = \varphi$. Estimating $(\theta_1,\dots,\theta_R)$ from $\boldsymbol{z}$ with the model as in (5.7) constitutes an R-dimensional generalized line spectral estimation problem. Here, generalized denotes the fact that we do not observe the complex harmonic directly, but linear functionals of these encoded by $\boldsymbol{\Phi}$. Finally, we can also rephrase (5.7) by means of an atomic set, if we let

$$\mathcal{A} = \left\{\boldsymbol{f}_{L_1,\dots,L_R}(\theta_1,\dots,\theta_R) \quad \text{for} \quad \theta_1,\dots,\theta_R \in \mathbb{R}\right\}, \tag{5.8}$$

which means we now have that (5.7) is of the form (5.3) based on (5.1).

The geometric idea behind this specific $\mathcal{A}$ is that we collect the image of the atomic function $\boldsymbol{f}$ in $\mathcal{A}$, since this corresponds to the set of all 1-sparse uncompressed signals we know from the grid-based approach (up to scalar rescaling). Additionally, in the specific case of DoA estimation we have that the compression matrix $\boldsymbol{\Phi} = \boldsymbol{\Xi}\boldsymbol{\Gamma}$ is now structured, since it depends on the matrix $\boldsymbol{\Gamma}$ which describes the angular dependency of the antenna array.

However, we still are in lack of efficient means for recovery. Hence, the goal of the next section is to derive a suitable optimization problem to reformulate the general ANM approach in (5.4) to our special case, such that we end up with an optimization problem that can be solved by a specific algorithm.

5.2.3 Semi-definite Programming for Line Spectral Estimation

In order to proceed, we first introduce a multiple-snapshots model based on Equation (5.3) and we define the appropriate extension of the atomic norm. To this end, let the atomic function for line spectral estimation $\boldsymbol{f} : \mathbb{R}^R \to \mathbb{C}^{L_1 \cdots L_R}$ be defined as in (5.7) then the respective atomic set for K snapshots denoted as $\mathcal{A}_K$ is defined as

$$\mathcal{A}_K = \left\{ \boldsymbol{f}_{L_1,\ldots,L_R}(\theta_1,\ldots,\theta_R)\boldsymbol{b}^{\mathrm{H}} \;\middle|\; \boldsymbol{b} \in \mathbb{C}^K, \|\boldsymbol{b}\|_2 = 1, \theta_1,\ldots,\theta_R \in \mathbb{R} \right\}, \quad (5.9)$$

such that $\mathcal{A}_K \subset \mathbb{C}^{L_1 \ldots L_R \times K}$. Then we can also describe K snapshots of the same parameter constellation as a sparse representation via

$$\boldsymbol{Z} = \boldsymbol{\Phi} \cdot \sum_{a_s \in \mathcal{A}_K} c_s \cdot \boldsymbol{a}_s = \boldsymbol{\Phi} \cdot \boldsymbol{Y} + \boldsymbol{N},$$

which is the analogue to the single-snapshot case in Equation (5.3). Next, one also has to extend the definition of the atomic norm itself to the MMV version $\|\cdot\|_{\mathcal{A}_k} : \mathbb{C}^{L \times K} \to \mathbb{R}$ defined as

$$\|\boldsymbol{Y}\|_{\mathcal{A}_K} = \inf \left\{ \sum_{s=1}^{S} |c_s| \;\middle|\; \boldsymbol{Y} = \sum_{s=1}^{S} c_s \boldsymbol{a}_s \quad \text{where} \quad \boldsymbol{a}_s \in \mathcal{A}_k \right\}, \quad (5.10)$$

to account for the case when we have K observations of the signal. Now, the set $\mathcal{A}_K$ consists of outer products of the original atoms $\boldsymbol{f}$ with unit norm vectors $\boldsymbol{b} \in \mathbb{C}^K$, which forms differently scaled copies of $\boldsymbol{f}$, where the scalings satisfy an energy constraint by means of $\|\boldsymbol{b}\|_2 = 1$.

Next, we need the notion of Hermitian multi-level Toeplitz matrices. Based on the definition in Section 3.3.2.4 via (3.12), we define

$$\boldsymbol{T}^{\mathbf{H}}_{(\boldsymbol{n},R)}(\boldsymbol{u}) = \boldsymbol{T}_{(\boldsymbol{n},R)}(\boldsymbol{t}(\boldsymbol{u})),$$

where the R-dimensional array $\boldsymbol{t}(\boldsymbol{u})$ satisfies

$$t_{n_1,n_2,\ldots,n_r} = \begin{cases} u_{n_1,n_2,\ldots,n_r} & \text{for} \quad n_1 \leqslant L_1 \\ \overline{u}_{2L_1+1-n_1,n_2,\ldots,n_r} & \text{for} \quad n_1 > L_1 \end{cases},$$

in order to render $\boldsymbol{T}^{\mathbf{H}}_{(\boldsymbol{n},R)}(\boldsymbol{u})$ Hermitian by definition. First, note that $\boldsymbol{T}^{\mathbf{H}}_{(\boldsymbol{n},R)}(\boldsymbol{u})$ is not the same as $\boldsymbol{T}_{(\boldsymbol{n},R)}(\boldsymbol{u})^{\mathrm{H}}$ and we have

$$\boldsymbol{T}^{\mathbf{H}}_{(\boldsymbol{n},R)}(\boldsymbol{u}) = (\boldsymbol{T}^{\mathbf{H}}_{(\boldsymbol{n},R)}(\boldsymbol{u}))^{\mathrm{H}}$$

instead.

Finally, we are in the position to present the statement that allows us to calculate the atomic norm in the multiple snapshot case, if we adapted $\mathcal{A}_K$ specifically to the setting where we want to do line spectral estimation.

Theorem 5.1 ([154], [176]). *With $\mathcal{A}_K$ given as in Equation (5.9) we have that $\|\boldsymbol{Y}\|_{\mathcal{A}_K} = w$, where w is the objective value of the solution to*

$$\min_{\boldsymbol{W},\boldsymbol{u}} \operatorname{tr} \boldsymbol{T}^{\mathbf{H}}_{(\boldsymbol{n},R)}(\boldsymbol{u}) + \operatorname{tr} \boldsymbol{W} \quad s.t. \quad \begin{bmatrix} \boldsymbol{T}^{\mathbf{H}}_{(\boldsymbol{n},R)}(\boldsymbol{u}) & \boldsymbol{Y} \\ \boldsymbol{Y}^{\mathrm{H}} & \boldsymbol{W} \end{bmatrix} \succeq \mathbf{0},$$

which constitutes a semi-definite program, since the constraints involve requirements on the spectrum of a matrix that depends on the vector of interest. ∎

Once one has access to $\boldsymbol{u}$ and hence also $\boldsymbol{T}^{\mathbf{H}}_{(\boldsymbol{n},R)}(\boldsymbol{u})$, once can use R-dimensional versions of ESPRIT or MUSIC to extract the unknown c_s and $(\theta_{1,s}, \ldots, \theta_{R,s})$ in Equation (5.7). This is based on the R-dimensional version of the Vandermonde-decomposition [176, Lem. 2.2] and [177], which in the single-dimensional case states that a positive definite Hermitian Toeplitz matrix $\boldsymbol{T}$ with rank S can be factorized as

$$\boldsymbol{T} = \boldsymbol{V} \cdot \boldsymbol{D} \cdot \boldsymbol{V}^{\mathrm{H}},$$

where $\boldsymbol{V} \in \mathbb{C}^{L \times S}$ is a Vandermonde matrix and $\boldsymbol{D} \in \mathbb{R}^{S \times S}$ is a diagonal matrix. The R-dimensional versions of the subspace algorithms ESPRIT and MUSIC allow to estimate $\boldsymbol{V}$ as

$$\boldsymbol{V} = [\boldsymbol{f}(\theta_{1,1}, \ldots, \theta_{R,1}), \ldots, \boldsymbol{f}(\theta_{1,S}, \ldots, \theta_{R,S}),]$$

and hence they allow to extract the parameters of interest from knowledge of $\boldsymbol{T}^{\mathbf{H}}_{(\boldsymbol{n},R)}(\boldsymbol{u})$.

As a last building block, we need to combine Theorem 5.1 with the approach of Atomic Norm Minimization in order to retrieve the methods for reconstruction of $\boldsymbol{Y}$ from $\boldsymbol{Z}$. The equivalent optimization problem to (5.4) is given in the case of generalized line spectral estimation as

$$\begin{aligned} &\min_{\boldsymbol{W},\boldsymbol{u}} \operatorname{tr} \boldsymbol{T}^{\mathbf{H}}_{(\boldsymbol{n},R)}(\boldsymbol{u}) + \operatorname{tr} \boldsymbol{W} \\ &\text{s.t.} \quad \begin{bmatrix} \boldsymbol{T}^{\mathbf{H}}_{(\boldsymbol{n},R)}(\boldsymbol{u}) & \boldsymbol{Y} \\ \boldsymbol{Y}^{\mathrm{H}} & \boldsymbol{W} \end{bmatrix} \succeq \mathbf{0}, \|\boldsymbol{Z} - \boldsymbol{\Phi}\boldsymbol{Y}\|_2^2 \leqslant \eta. \end{aligned} \tag{5.11}$$

Since the number of dimensions R of the parameter space is equal to the number of levels of the R-level Toeplitz matrix, this kind of optimization

problem becomes increasingly high-dimensional and hard to solve. On the one hand, general purpose solvers for semi-definite programs, like [178, 179] are not able to account for the R-level Toeplitz structure for the case $R > 1$. On the other hand, many variants of the algorithms proposed to solve (5.11) do not estimate the parameters jointly [159] or assume only simple compression strategies, like sub-selection [180]. Hence, the problem in Equation (5.11) should be solved as is, but we might have to be satisfied with an approximate solution.

5.2.4 Recovery Guarantees for Atomic Norm Minimization

One question that remains is the existence of a unique solution of the ANM problem given in (5.4) or more specifically the problem of generalized line spectral estimation as in (5.11). In [175] the authors study the guarantees that can be obtained from employing sub-Gaussian measurements as we did for the grid-based setting in Section 2.2. In order to present their main result, we first define sub-Gaussian matrices.

Definition 5.1. A matrix $\boldsymbol{A} \in \mathbb{C}^{K \times M}$ is b-sub-Gaussian with population covariance $\boldsymbol{\Sigma}$ if its rows are stochastically independent of each other and for all $k = 0, \dots, K-1$ the k-th row $\boldsymbol{a}^k \in \mathbb{C}^{1 \times M}$ of $\boldsymbol{A}$ satisfies

$$\mathbf{E}\{\boldsymbol{a}^k\} = 0, \quad \mathbf{E}\{\boldsymbol{a}^{k\mathrm{H}} \boldsymbol{a}^k\} = \boldsymbol{\Sigma} \tag{5.12}$$

for an invertible covariance matrix $\boldsymbol{\Sigma} \in \mathbb{C}^{M \times M}$ and for any vector $\boldsymbol{x} \in \mathbb{C}^M$ it holds that

$$\mathbf{P}\left(\left|\boldsymbol{a}^k \boldsymbol{x}\right| \geqslant t \|\boldsymbol{x}\|_2\right) \leqslant \exp\left(-\frac{t^2}{b^2}\right). \tag{5.13}$$

■

Examples for sub-Gaussian distributions are of course Gaussian distributions or sums of Rademacher distributions. With this notion at hand, we can present the result in [175, Thm. 2], where the authors state that these matrices are good compression matrices in the sense stated below.

Theorem 5.2. *Let $\boldsymbol{\Phi} \in \mathbb{C}^{m \times N}$ be a b-sub-Gaussian matrix with population covariance $\boldsymbol{\Sigma}$ and measurements given by $\boldsymbol{z} = \boldsymbol{\Phi}\boldsymbol{y}$ for $\boldsymbol{y}$ as in (5.7). Assume furthermore that the unknown parameters satisfy a minimum separation condition*

$$\min_{s_1 \neq s_2 \in [S]} \min_{r=1,\dots,R} |\boldsymbol{\theta}_{r,s_1} - \boldsymbol{\theta}_{r,s_2}| \geqslant \frac{4}{L_1 \cdot \dots \cdot L_R}.$$

Then as long as

$$m \geqslant c(R) S \log(L) b^{-2} \kappa(\boldsymbol{\Sigma})$$

for a fixed constant $c(R)$ depending on the dimension R of the frequencies, $\boldsymbol{y}$ is the unique minimizer of (5.11) *with probability at least* $1 - \exp(-(K-2)/8)$. *Here* $\kappa(\boldsymbol{\Sigma})$ *is the condition number of* $\boldsymbol{\Sigma}$. ∎

If we remember Remarks 2.1 and 2.4 from the previous parts of the thesis, we can see that above statement has the same flavor. It states under which conditions a certain compressive measurement process is able to efficiently recover the unknown frequencies. Again, it merely guarantees this recovery with high probability. Interestingly, even in this seemingly much more challenging setting where the frequencies that have to be reconstructed do not reside on a grid, the scaling law is essentially the same as the one provided by Equation (2.19) given that we obey a certain minimum separation condition.

In the following, we use two prominent sub-Gaussian ensembles. First, properly rescaled Gaussian matrices and second matrices whose entries are drawn from a Rademacher distribution. It is easy to show for both cases that they are instances of the sub-Gaussian ensemble.

By means of Theorem 5.2 we have also demonstrated that the approach of ANM together with these random matrices is a viable method to extract the spectral components from compressive measurements. And finally, it motivates us in Sections 5.3 and 5.3.3 to use sub-Gaussian matrices as initial guesses to initialize the proposed iterative method for measurement matrix optimization.

The next section Section 5.2.5 puts the previously developed theoretic results to a first test by conducting numerical experiments, which tackle a one-dimensional version of the problem in Example 5.1.

5.2.5 Semi-definite Programming Simulations

To demonstrate the performance of the proposed estimator, which consists of solving Equation (5.11) for the specific model as given in (5.6), we implement it using the Convex Optimization toolboX (CVX) and the SDPT3 solver [178, 179] for the case $R = 1$ and $K = 1$, where the side-constraint is that we have a block-matrix that contains an Hermitian Toeplitz block. As such this scenario can still be handled by CVX.

For the first experiment, we use a Stacked Polarimetric Uniform Circular Patch Array (SPUCPA) which is depicted in Figure 5.4. It consists of two stacked 12-element uniform circular patch arrays and an additional cube of five patch elements on top. Each element has two ports for vertical and horizontal polarization. For the simulations, we use only the two rings of 12 elements and only the vertically polarized port so that

$M = 24$ ports are available. Moreover, the EADF for the beam pattern (which was measured in an anechoic chamber) contains $L = 25$ coefficients per antenna element. We would like to stress the fact that although we use a special array, the simulations generalize to any array, where the EADF coefficients are known and we merely incorporated the SPUCPA array as an illustrative example. To quantify the performance of our estimator, we compare it to the deterministic CRB computed according to Equation (2.25) in Section 2.4.4.

In Figure 5.5 we show the empirical phase transition for the noiseless case $\boldsymbol{w} = \mathbf{0}$. We vary the number of sources S and the number of measurements m and draw the source positions randomly. Moreover, the amplitudes $\boldsymbol{c}$ in (5.7) are drawn randomly on the complex unit circle. We depict the empirical estimation error defined as $\sum_{s=1}^{S}(\theta_s - \hat{\theta}_s)^2$ on a logarithmic scale, i.e., values below -10 correspond to an estimation error below 10^{-10} which can be considered to be rounding errors of the floating point representation. Figure 5.5 considers a ± 1 binary iid Rademacher distribution and an iid Gaussian distribution for the creation of $\boldsymbol{\Phi}$, showing the best realization among 100 trials. We observe a quite sharp phase transition that occurs between $m = S$ and $m = 2S$ illustrated by the fact that left of the cases corresponding to $m = S$ the reconstruction error is constantly high, whereas right of the cases $m = 2S$ we only observe rounding errors.

Figure 5.4 Stacked polarimetric uniform circular patch array with 58 ports. We have used only the ports corresponding to the two stacked circular arrays with 12 elements per ring. ■

To investigate the performance in the presence of noise with variance σ^2, the results depicted in Figure 5.6 show the empirical estimation error of the proposed method vs. the CRB for the compressed and the uncompressed cases, where the optimization in (5.11) was run with $\eta = \sigma/m$. We compare the effect of using iid Gaussian and Rademacher distributed compression matrices. For both we show the median and the 25/75 percentiles. Moreover, the corresponding CRBs are shown as well. In the top of Figure 5.6 we consider $S = 2$ sources and $m = 12$ (i.e., a compression rate of 50 %) whereas in bottom of Figure 5.6 we choose $S = 3$ sources

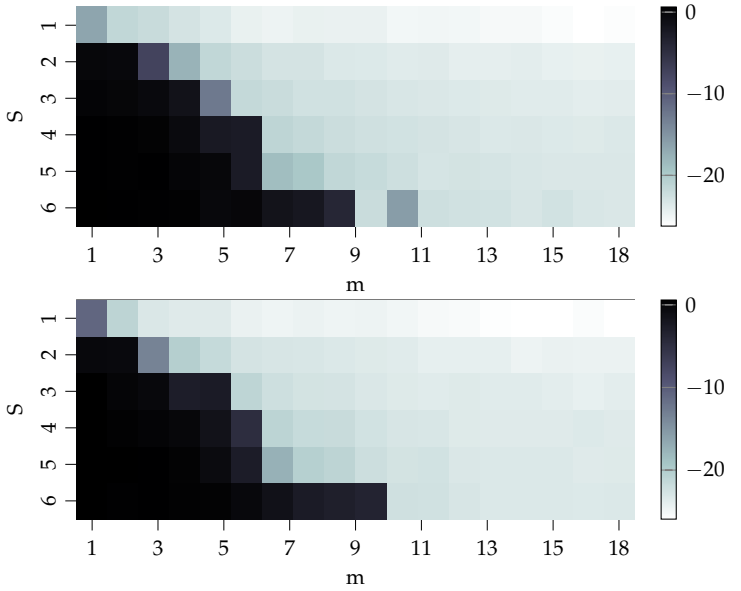


Figure 5.5 *As in the grid-based setting the phase transition in terms of measurements vs. sparsity is very sharp.* – Estimation error (logarithmic scale) vs. measurement size m, and sparsity S for the noise-free case and Gaussian (top) and Rademacher (bottom) distributed compression matrices. ■

and $m = 15$ (i.e., a compression rate of 62.5 %). As before, the source positions are drawn randomly. The results show the statistics over 2500 trials, where we also regenerate the compression matrices, and confirm that both compression matrix distributions behave almost identically and provide estimation errors that show some gap to the CRB, but behave similarly to it.

In the last experiment we consider an array of $M = 29$ isotropic antenna elements in a randomly generated array geometry, which is depicted in Figure 5.7 on the left. We generate $S = 5$ sources at random positions in a manner similar to the previous experiment. The right plot in Figure 5.7 shows the empirical estimation error vs. the CRB for the case where no compression is applied ($\boldsymbol{\Phi} = \boldsymbol{I}_M$). The result demonstrates that the proposed method enables grid-free sparsity-based DoA estimation with arbitrary array geometries and that closely follows the CRB.

Finally, we would like to note that the simulations in Figure 5.5 are conservative by design in terms of number of sources S. This is due to the

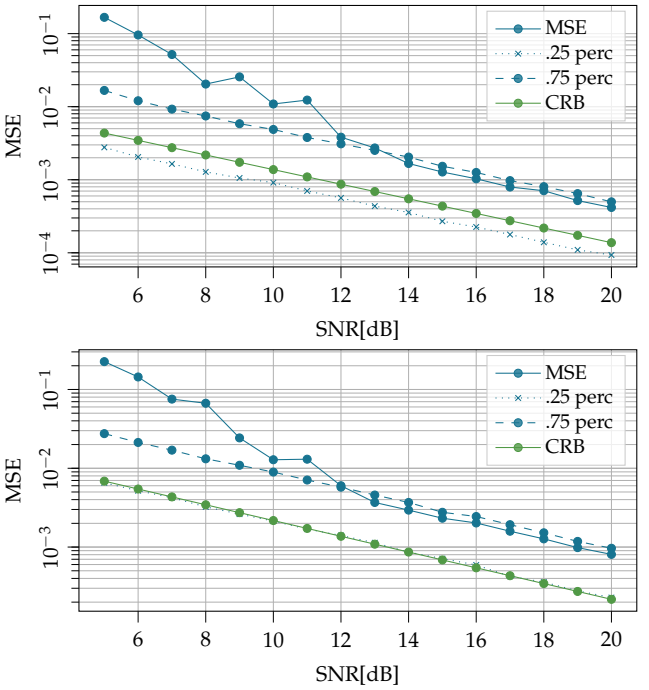


Figure 5.6 *The proposed estimator is not able to achieve the CRB.* – Estimation error vs. SNR for $S = 2$ sources and $m = 12$ (top), Estimation error vs. SNR for $S = 3$ sources and $m = 15$ (bottom) ■

fact that they have to be aligned with the minimum separation condition in Theorem 5.2, which has to be fulfilled to make ANM a theoretically viable approach. To this end, we have to keep the number of sources significantly lower than M.

These nevertheless promising results for the case $R = 1$ motivate us to extend this approach to arbitrary dimensions. Hence, in the following section we present as a main result of this thesis a derivation of an algorithm that is capable of approximating a solution to the problem in (5.11) based on the so-called ADMM [181].

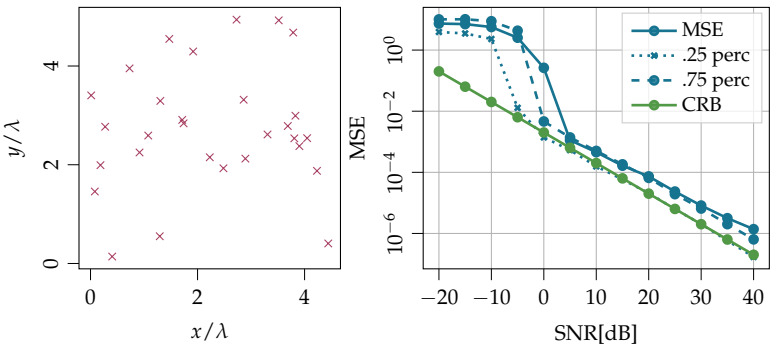


Figure 5.7 *The random geometry allows estimation closer to the CRB.* – Randomly drawn array geometry with $M = 29$ elements and the resulting estimation error vs. SNR for $S = 5$ sources (no compression). ■

5.2.6 ADMM for Semi-definite Programming

As we have seen in Equation (5.7), the data model for line spectral estimation can be formulated for arbitrary dimension R of the frequency parameters. Also, the reconstruction method in Equation (5.11) can be used for arbitrary R due to the recursive nature of the Hermitian R-level Toeplitz matrices. Further, it is of high interest to derive efficient solvers for the case $R > 1$, since the setting in Example 5.1 for DoA estimation already requires $R = 2$. Another motivation for the case $R > 1$ is given in [162], where essentially the case $R = 6$ has to be considered, if one would like to treat the data model for multiple input multiple output (MIMO) channel sounding considered there.

Given these motivations, as a main contribution of this thesis we would like to replace the direct solvers as employed in the previous Section 5.2.5 with a more flexible and and the same time specialized algorithm by more explicitly exploiting the inherent structure of the semi-definite program. For this task, a specialized version of the ADMM [181] has been proposed [182, 183].

We take these results as a starting point, to derive explicit closed form formulations for the iterative update steps of the ADMM in order to approximate a solution to (5.11). Consequently, following [181, 182, 183], we first define the so-called augmented Lagrangian of the problem in (5.11), which reads as

$$\min_{\boldsymbol{W},\boldsymbol{u},\boldsymbol{Y}} \mathcal{L}(\boldsymbol{W},\boldsymbol{u},\boldsymbol{Y},\boldsymbol{V},\boldsymbol{\Lambda}) \quad \text{s.t.} \quad \boldsymbol{V} \succeq \boldsymbol{0}, \boldsymbol{\Lambda} \succeq \boldsymbol{0} \tag{5.14}$$

where $\boldsymbol{W} \in \mathbb{C}^{K\times K}$, $\boldsymbol{u} \in \mathbb{C}^{N_1\times 2N_2-1\cdots\times 2N_d-1}$, $\boldsymbol{Y} \in \mathbb{C}^{M\times K}$, $\boldsymbol{V} \in \mathbb{C}^{M+K\times M+K}$ and $\boldsymbol{\Lambda} \in \mathbb{C}^{M+K\times M+K}$. The augmented Lagrangian function $\mathcal{L}$ is evaluated as

$$\begin{aligned}\mathcal{L}(\boldsymbol{W},\boldsymbol{u},\boldsymbol{Y},\boldsymbol{V},\boldsymbol{\Lambda}) = &\left\langle \boldsymbol{\Lambda}, \boldsymbol{V} - \underbrace{\begin{bmatrix} \boldsymbol{T}^{\mathrm{H}}_{(\boldsymbol{n},R)}(\boldsymbol{u}) & \boldsymbol{Y} \\ \boldsymbol{Y}^{\mathrm{H}} & \boldsymbol{W}\end{bmatrix}}_{=\boldsymbol{T}} \right\rangle \\ &+ \frac{1}{2}\|\boldsymbol{\Phi}\boldsymbol{Y} - \boldsymbol{Z}\|_F^2 + \frac{\tau}{2}(\operatorname{tr}\boldsymbol{W} + \operatorname{tr}\boldsymbol{T}^{\mathrm{H}}_{(\boldsymbol{n},R)}(\boldsymbol{u})) \\ &+ \frac{\rho}{2}\left\| \boldsymbol{V} - \begin{bmatrix} \boldsymbol{T}^{\mathrm{H}}_{(\boldsymbol{n},d)}(\boldsymbol{u}) & \boldsymbol{Y} \\ \boldsymbol{Y}^{\mathrm{H}} & \boldsymbol{W}\end{bmatrix} \right\|_F^2,\end{aligned}$$

where $\tau > 0$ and $\rho > 0$ are suitably chosen constants. Here τ plays the role of a regularizing parameter between data fitting and the magnitude of the atomic norm of $\boldsymbol{Y}$. And ρ tunes the influence of the convex term it is multiplied with, which renders $\mathcal{L}$ a convex function for large enough ρ. We also partition the matrices $\boldsymbol{\Lambda}$ and $\boldsymbol{V}$ such that they match the partitioning of the blocks in $\boldsymbol{T}$ as

$$\boldsymbol{\Lambda} = \begin{bmatrix} \hat{\boldsymbol{\Lambda}} & \boldsymbol{\Lambda}_1 \\ \boldsymbol{\Lambda}_1 & \boldsymbol{\Lambda}_0 \end{bmatrix} \quad \text{and} \quad \boldsymbol{V} = \begin{bmatrix} \hat{\boldsymbol{V}} & \boldsymbol{V}_1 \\ \boldsymbol{V}_1 & \boldsymbol{V}_0 \end{bmatrix}.$$

We now use the results of Appendix C.2 in order to calculate the partial derivatives of $\mathcal{L}$, which have to be handled carefully, since we have to consider the fact that the variables $\mathcal{L}$ depends on are complex valued and highly structured. To this end, we make use of the results in Appendix B.2, which introduce the so-called Wirtinger calculus.

In general and according to [183] the iteration of ADMM in step k can be expressed as

$$(\boldsymbol{W}^{k+1}, \boldsymbol{u}^{k+1}, \boldsymbol{Y}^{k+1}) \leftarrow \operatorname*{argmin}_{\boldsymbol{W},\boldsymbol{u},\boldsymbol{Y}} \mathcal{L}(\boldsymbol{W},\boldsymbol{u},\boldsymbol{Y},\boldsymbol{V}^k,\boldsymbol{\Lambda}^k)$$

$$\boldsymbol{V}^{k+1} \leftarrow \operatorname*{argmin}_{\boldsymbol{V}\succeq 0} \mathcal{L}(\boldsymbol{W}^{k+1},\boldsymbol{u}^{k+1},\boldsymbol{Y}^{k+1},\boldsymbol{V},\boldsymbol{\Lambda}^k)$$

and

$$\boldsymbol{\Lambda}^{k+1} = \boldsymbol{\Lambda}^k + \rho(\boldsymbol{V}^{k+1} - \boldsymbol{T}^{k+1})$$

By using the derivatives of Appendix C.2 in (C.11), (C.12) and (C.13) we can compute above updates in closed form by setting the respective

derivatives to 0, thus satisfying the necessary conditions for local optimality, and solving for the variable of interest. Note that here, 0 is the neutral element with respect to addition in the respective vector space the derivatives are resided in. Finally, they read as follows

$$\boldsymbol{u}^{k+1} = \frac{1}{\rho \mathfrak{D}_{\boldsymbol{n},d}(\mathbf{1})} \left(\mathfrak{D}_{\boldsymbol{n},d}(\hat{\boldsymbol{\Lambda}}^k - \rho \hat{\boldsymbol{V}}^k) - \frac{\tau}{2} \boldsymbol{i}_1 \right) \tag{5.15}$$

$$\boldsymbol{W}^{k+1} = \frac{1}{\rho} \boldsymbol{\Lambda}_0^k + \boldsymbol{V}_0^k - \boldsymbol{I} \frac{\tau}{2\rho} \tag{5.16}$$

$$\boldsymbol{Z}^{k+1} = \left(\boldsymbol{\Phi}^{\mathrm{H}} \boldsymbol{\Phi} - \rho \boldsymbol{I} \right)^{-1} \cdot \left(\boldsymbol{\Phi} \boldsymbol{Y} + \boldsymbol{\Lambda}_1^k + \rho \boldsymbol{V}_1^k \right), \tag{5.17}$$

where $\left(\boldsymbol{\Phi}^{\mathrm{H}} \boldsymbol{\Phi} - \rho \boldsymbol{I} \right)^{-1}$ can be precomputed in advance to avoid repeatedly solving a linear system. See Appendix C.2 for the definition of $\mathfrak{D}_{\boldsymbol{n},d}$. Furthermore we also update $\boldsymbol{T}$ iteratively from the current variables according to

$$\boldsymbol{T}^{k+1} = \begin{bmatrix} \boldsymbol{T}^{\mathbf{H}}_{(\boldsymbol{n},R)}(\boldsymbol{u}^{k+1}) & \boldsymbol{Y}^{k+1} \\ \boldsymbol{Y}^{\mathrm{H}k+1} & \boldsymbol{W}^{k+1} \end{bmatrix} \tag{5.18}$$

whereas we finally can also update $\boldsymbol{V}$ via

$$\boldsymbol{V}^{k+1} = \mathfrak{P}^{\succeq} \left(\boldsymbol{T}^{k+1} - \rho \boldsymbol{\Lambda}^k \right), \tag{5.19}$$

where $\mathfrak{P}^{\succeq}$ is the orthogonal projection onto the positive cone of the positive semi-definite matrices, which can be realized numerically by an eigen-decomposition and is the computationally most expensive step during the iterate updates. With this we have completed the iterations for ADMM which now only consists of initializing $(\boldsymbol{W}^0, \boldsymbol{Y}^0, \boldsymbol{u}^0, \boldsymbol{V}^0, \boldsymbol{\Lambda}^0)$ and iteratively carrying out the steps in (5.15), (5.16), (5.17), (5.18) and (5.19). With the operators $\boldsymbol{T}^{\mathbf{H}}$ and $\mathfrak{D}$ defined as above, which can be implemented recursively for arbitrary dimensionality R of the underlying parameter space, one can create a very general implementation as outlined in Algorithm 5.1.

The interpretation of Algorithm 5.1 is that it provides an estimator for the signal covariance $\operatorname{Cov} \boldsymbol{Y}$ by means of the assumption that it is a Toeplitz matrix, since the signal $\boldsymbol{Y}$ originates from a superpositions of complex sinusoidals. The proposed method allows to estimate the covariance of $\boldsymbol{Y}$ directly from the compressed observations $\boldsymbol{Z}$ by making use of the underlying sparsity of the representation of $\boldsymbol{Y}$ and the knowledge how $\boldsymbol{Z}$ originates from $\boldsymbol{Y}$. Hence, this allows the combination of ADMM

Data: Set of observed measurement vectors $\boldsymbol{Z}$; Sparsity order $S_{\max} \in \mathbb{N}$; Dimensionality of the underlying parameters R; Regularization parameters ρ, τ; Steps T to run the iteration;

[1] Initialize $(\boldsymbol{W}^0, \boldsymbol{Y}^0, \boldsymbol{u}^0, \boldsymbol{V}^0, \boldsymbol{\Lambda}^0)$ randomly;

[2] $k = 0$;

[3] **while** $k < T$ **do**

[4] $\quad \boldsymbol{u}^{k+1} = \frac{1}{\rho \mathfrak{D}_{\boldsymbol{n},d}(\mathbf{1})} \left(\mathfrak{D}_{\boldsymbol{n},d}(\hat{\boldsymbol{\Lambda}}^k - \rho \hat{\boldsymbol{V}}^k) - \frac{\tau}{2} \boldsymbol{i}_1 \right)$;

[5] $\quad \boldsymbol{W}^{k+1} = \frac{1}{\rho} \boldsymbol{\Lambda}_0^k + \boldsymbol{V}_0^k - \boldsymbol{I} \frac{\tau}{2\rho}$;

[6] $\quad \boldsymbol{Z}^{k+1} = \left(\boldsymbol{\Phi}^{\mathrm{H}} \boldsymbol{\Phi} - \rho \boldsymbol{I} \right)^{-1} \cdot \left(\boldsymbol{\Phi} \boldsymbol{Y} + \boldsymbol{\Lambda}_1^k + \rho \boldsymbol{V}_1^k \right)$;

[7] $\quad \boldsymbol{T}^{k+1} = \begin{bmatrix} \boldsymbol{T}^{\mathbf{H}}_{(\boldsymbol{n},R)}(\boldsymbol{u}^{k+1}) & \boldsymbol{Y}^{k+1} \\ \boldsymbol{Y}^{\mathrm{H}k+1} & \boldsymbol{W}^{k+1} \end{bmatrix}$;

[8] $\quad \boldsymbol{V}^{k+1} = \mathfrak{P}^{\succeq} \left(\boldsymbol{T}^{k+1} - \rho \boldsymbol{\Lambda}^k \right)$;

[9] **end**

[10] run a variant of ESPRIT or MUSIC on $\boldsymbol{T}^{\mathbf{H}}_{(\boldsymbol{n},R)}(\boldsymbol{u}^{T-1})$ to estimate $S_{\max}$ parameters of dimension R;

[11] run least-squares to estimate the complex amplitudes c_s;

Algorithm 5.1: *The resulting algorithm for R-dimensional generalized line spectral estimation from multiple measurements.* – The derived Alternating Directions of Multipliers Method based on (5.15), (5.16), (5.17), (5.18) and (5.19).

with subspace methods that work with estimates of the covariance of $\boldsymbol{Y}$. Note that theoretically ADMM can even provide estimates for the covariance of $\boldsymbol{Y}$ even in the single snapshot case $K = 1$.

5.2.7 Simulations for ADMM

To quantify the performance of our estimator in Algorithm 5.1, we compare its estimation error to the deterministic CRB. With the proposed spatial compression, the deterministic CRB for the R-dimensional case with K snapshots can be computed via (2.25) and in this case reads as

$$C(\boldsymbol{\theta}) = \frac{\sigma^2}{2 \cdot K} \operatorname{tr}\left(\left[\Re(\boldsymbol{D}^{\mathrm{H}} \boldsymbol{\Pi}_{\boldsymbol{G}}^{\perp} \boldsymbol{D} \odot (\mathbf{1}_{R \times R} \otimes \hat{\boldsymbol{R}})^{\mathrm{T}}) \right]^{-1} \right), \tag{5.20}$$

with $\boldsymbol{\Pi}_{\boldsymbol{G}}^{\perp} = \boldsymbol{I} - \boldsymbol{G}(\boldsymbol{G}^{\mathrm{H}}\boldsymbol{G})^{-1}\boldsymbol{G}^{\mathrm{H}}$ and $\hat{\boldsymbol{R}} = 1/K \cdot \boldsymbol{S}\boldsymbol{S}^{\mathrm{H}}$ being the sample covariance and we have set

$$\boldsymbol{G} = \boldsymbol{\Phi}[\boldsymbol{f}(\theta_1), \ldots, \boldsymbol{f}(\theta_S)],\ \boldsymbol{D}_i = \frac{\partial}{\partial \boldsymbol{\theta}_i}\boldsymbol{G},\ \boldsymbol{D} = [\boldsymbol{D}_1, \ldots, \boldsymbol{D}_R]$$

First, we carry out the atomic norm minimization for 3D line spectral estimation with uncompressed measurements, so $\boldsymbol{\Phi} = \boldsymbol{I}$ in (5.11) and also compressed measurements, where the entries of $\boldsymbol{\Phi}$ are drawn iid from a zero-mean Gaussian distribution and then we project the columns to the complex unit sphere in $\mathbb{C}^m$, thus normalizing the columns independently. Here, we chose $m = \lfloor \rho \cdot N \rfloor$ according to some compression rate $\rho \in (0, 1]$. In case of $\boldsymbol{\Phi} = \boldsymbol{I}$, so $\rho = 1$, we also run 3D-Standard-ESPRIT [163] directly on $\boldsymbol{Y}$ as a comparison, which is only applicable in this case, since ESPRIT is not able to deal with compressive measurements of the kind employed here. In any case, we choose $\tau = \sigma^{0.8}$ as a simple heuristic and $\rho = 0.05$ to run the ADMM and we initialized the state variables $(\boldsymbol{W}^0, \boldsymbol{u}^0, \boldsymbol{Z}^0, \boldsymbol{V}^0, \boldsymbol{\Lambda}^0)$ by sampling the real and imaginary parts from standard Gaussian distributions.

The results in Figure 5.8, where we plot the reconstruction error versus the noise variance σ^2, show that the derived ADMM approach is able to replicate the performance predicted by the CRB for the case $\rho = 1$, thus delivering the same performance as 3D-Standard-ESPRIT.

In the case $\rho = 0.75$ we see that the ADMM algorithm's performance highly depends on the number of steps T carried out to estimate the covariance $\boldsymbol{T}^{\mathbf{H}}_{(\boldsymbol{n},d)}(\boldsymbol{u}^{T-1})$, since the error floor decreases when iterating for 1000 steps instead of 100. In conclusion, this means that the ADMM approach also achieves the CRB after a suitable amount of iterations. However these many iterations are usually only necessary in the high-SNR regime.

Moreover, we use the derived algorithm's flexibility and apply it to the 2D DoA estimation problem in Example 5.1 with a 12×3, so $P =$

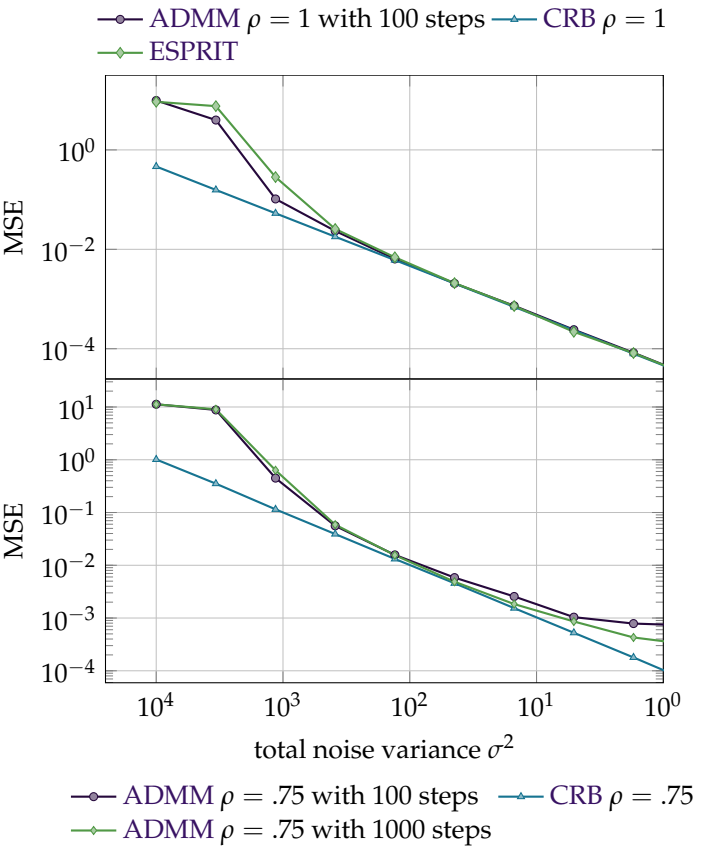


Figure 5.8 *For the line spectral estimation case, the ADMM attains the CRB.* – ADMM reconstruction performance for line spectral estimation of $S = 3$ sources and $d = 3$-dimensional frequencies with $\boldsymbol{k} \in [3, 3, 3]$ in comparison to standard ESPRIT using $K = 100$ snapshots. ■

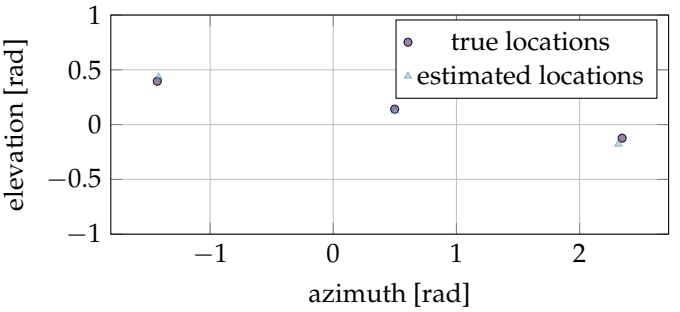


Figure 5.9 *In the case of DoA estimation, the reconstruction is also close to the ground truth.* – Simulation results from a 2D DoA estimation scenario using a stacked uniform linear array with 3 stacks and 12 elements each with noise standard deviation of $\sigma = 0.01$. ■

36, stacked circular array, where the stacks are aligned in the x-y-plane with distance $\mathrm{d}z = 0.375\lambda$ and diameter $12/16\lambda = 0.75\lambda$ and λ is the wavelength of the impinging wave. We use the Fourier coefficients of this (synthetic) array to formulate the DoA problem into a line spectral estimation problem as in (5.6) for $\boldsymbol{\Xi} = \boldsymbol{I}_P$, hence $\boldsymbol{\Phi} = \boldsymbol{\Gamma}$ in (5.7). A single scenario is depicted in Figure 5.9 where the noise variance is $\sigma^2 = 0.001$ and we recover these locations in the 2D angular domain from $K = 100$ snapshots. It is worth noting that these results can also be obtained from realistic arrays described by measured data and also if $\boldsymbol{\Phi}$ actually carries out a compression step in the case $\boldsymbol{\Xi} \neq \boldsymbol{I}_P$ as we will see later.

As a final simulation, we strive for a comparison of a vanilla ANM approach using CVX and the proposed ADMM. To this end we setup a scenario with randomly subselected identity matrices as compression matrices for a compression level of 75% with the rest of the properties as in Figure 5.6 which we averaged over 50 trials per level of SNR. As expected, we can see in Figure 5.10 that the CVX-based optimization is very closely following the CRB and the proposed ADMM due to its approximate and iterative nature is not exactly replicating the behavior of the CRB and CVX, but instead requires an increasing amount of iterations for higher SNR levels.

Summarizing, we have demonstrated that the derived ADMM iteration scheme is capable of recovering the unknown multidimensional frequencies using a very general model based on multiple snapshots and how it can be applied to 2D DoA estimation using arbitrary antenna

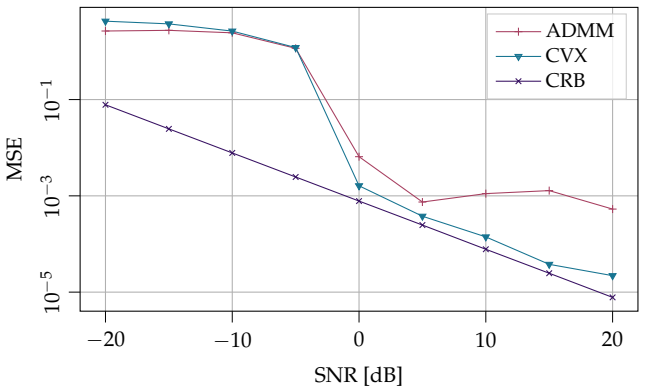


Figure 5.10 *The proposed ADMM closely follows the CRB and CVX, especially in the low-SNR regime.* – Comparison of the exact solver for ANM by solving (5.11) using CVX and the proposed approximate solver ADMM. ■

arrays.

Now that we have contributed to AOI-M by rephrasing DoA estimation as a generalized line spectral estimation problem and have shown how to solve it by means of ADMM, hence addressing AOI-R, we turn to AOI-C. To this end, we show how to construct measurement matrices $\mathbf{\Phi}$ that are better suited for parameter estimation than the random ensemble.

5.3 Sensing Matrix Design using Stochastic Gradient Descent

In this chapter, we propose an algorithm based on Stochastic Gradient Descent with momentum to find well behaved measurement schemes for arbitrary atomic sets, or alternatively: very general parametric data models which are combined with a compression step. As such it operates grid-free, so it does not impose sampling of the parameter space on a finite set as in Section 2.4.3 and it can operate on arbitrary dimensional parameter spaces, which in light of the previous Section 5.2.6 is a desired property. Despite the fact that it is originally designed to be employed in the grid-free setting, it can also be easily adapted to the grid-based setting to solve problems like the one in Equation (2.21).

Furthermore, it allows for fine-grained tuning of the computational complexity using the size of the used mini-batches and the number of iterations that are carried out. As such, we are presenting a generalized version of a variant already introduced in [O4] and by doing so we address AOI-C.

5.3.1 Introduction and Motivation

Naturally, the generalized reconstruction method based on ANM sparks the need for reconstruction guarantees similar to the ones presented in Section 2.4. One would like to derive statements in the spirit of Theorem 2.8 that specify under which conditions a certain measurement matrix $\mathbf{\Phi} \in \mathbb{C}^{m \times M}$ allows for efficient, robust and stable recovery using ANM.

Due to the success of the random Gaussian example in the grid-based setting, publications like [170] and [175] study the ramifications of implementing the sensing step with a Gaussian matrix when recovery happens via ANM. In [170] the authors treat a very general setting, without the need to specify the atomic set $\mathcal{A}$, whereas in [175] the specific case of generalized line spectral estimation as in Section 5.2.3 is treated. In both cases, the authors derive sufficient conditions for a unique solution of the problem in Equation (5.4) in terms of the necessary number of measurements m. We present one of these results in Section 5.2.4

In the case when $\mathcal{A}$ is derived from a parametric model $\boldsymbol{a}$, as for DoA estimation or line spectral estimation, this random process of selecting the sensing matrix introduces a randomness in the parameter space, since

for random $\boldsymbol{\Phi}$ the mapping $\boldsymbol{b} : \Theta \to \mathbb{C}^m$ defined as

$$\boldsymbol{b}(\boldsymbol{\theta}) = \boldsymbol{\Phi} \cdot \boldsymbol{a}(\boldsymbol{\theta})$$

is then random as well. Hence, for a fixed parameter $\boldsymbol{\theta}$ the information one can infer on $\boldsymbol{\theta}$ from observing $\boldsymbol{b}(\boldsymbol{\theta})$ can vary substantially, around the expectation $\mathbf{E}(\boldsymbol{b}(\boldsymbol{\theta}))$, see Figure 5.15. This means that some parameter configurations can be estimated better than others, which is not a good feature of a compressive sensing system, when the estimation process cannot provide uniform recovery precision over the whole parameter set Θ.

This was also already observed in the grid-based setting, where the theory dictates that random Gaussian matrices behave optimally in terms of the needed number of measurements to guarantee exact and stable recovery. Hence, interest quickly shifted to optimization methods that improve upon the random ensemble.

In publications like [184, 185, 186, 187] the approach is to parametrize a suitable optimization problem involving the sparsifying dictionary $\boldsymbol{A}$, where the problem itself optimizes $\boldsymbol{\Phi}$ when being solved numerically. Often, these problems are convex and have convex side-constraints, hence can be solved efficiently using (projected) gradient iterations or can be even solved in closed form by exploiting the SVD of an appropriate matrix. Usually, these methods optimize the coherence of the resulting matrix, which is motivated by Theorem 2.4.

However, these methods are not directly applicable to the grid-free setting, where the dimensionality of the parameter space is large ($R > 1$), since they heavily depend on the definition of the matrix $\boldsymbol{A}$ as the dictionary. Hence, to have a meaningful result for the optimization the dictionary $\boldsymbol{A}$ has to be created as

$$\boldsymbol{A} = [\boldsymbol{a}(\boldsymbol{\theta}_1), \dots, \boldsymbol{a}(\boldsymbol{\theta}_M)] \in \mathbb{C}^{N \times M},$$

where M usually has to be of the order $\mathcal{O}(\Delta^{-R})$ for some small Δ, to achieve a coverage of the parameter space Θ with spacing Δ. To problems arise. First, it is not clear how small Δ has to be, second the magnitude of R has a significant influence on the feasibility of the algorithms presented in [184, 185, 186, 187], especially when one has to compute an SVD of a matrix involving the dimensions of $\boldsymbol{A}$. Lastly, it it not directly obvious how the optimization of $\boldsymbol{\Phi}$ based on $\boldsymbol{A}$ translates to parameters θ that are not used to form $\boldsymbol{A}$, i.e. the off-grid parameters.

Instead in [38], a low-complexity design approach based on the spatial correlation function (SCF), see Equation (5.21), for 1D DoA estimation

is proposed while its extension for 2D DoA estimation is investigated in [188]. There, instead of taking the complete 4D-SCF to define the cost function for optimization, only 2D subsets are used in order to maintain a tractable computation path. A method to choose these 2D subsets, and an evaluation of the DoA performance while achieving considerable reduction in computational requirements compared to a direct extension of the approach in [38] is described in [188] as well. Despite the effectiveness of the approach in [188], its computational requirements raise substantially with increasing size, in terms of number of elements and aperture of the antenna array.

Hence, in the following, we use the core idea of [188] and extend it with respect to two important aspects. First, the extension can be applied to the grid-free setting, where the optimization does not need the imposition of a fixed grid in Θ. Second, it is not limited to the specific structure of $\boldsymbol{a}$ as in the DoA setting, but can also readily be applied to any parametric model.

5.3.2 The Algorithm

The following section aims at deriving a gradient based algorithm to generate a well performing compression matrix $\boldsymbol{\Phi} \in \mathbb{C}^{m \times N}$ for a general atomic set $\mathcal{A}$ which is generated from a parametric model $\boldsymbol{a} : \Theta \to \mathbb{C}^N$ via

$$\mathcal{A} = \{\boldsymbol{a}(\boldsymbol{\theta}) | \boldsymbol{\theta} \in \Theta\},$$

as introduced in Section 5.2.1. For the parametric model $\boldsymbol{a}$ the spatial correlation function $\rho : \Theta \times \Theta \to \mathbb{C}$ is given by

$$(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) \mapsto \rho(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) = \boldsymbol{a}^{\mathrm{H}}(\boldsymbol{\theta}_1)\boldsymbol{a}(\boldsymbol{\theta}_2), \tag{5.21}$$

which can also be viewed as the auto-correlation function of the set $\mathcal{A}$. However, since our observations in Equation (5.3) do contain linear projections of the atoms in $\mathcal{A}$ by virtue of the matrix $\boldsymbol{\Phi}$, we additionally define the effective SCF $\rho_{\boldsymbol{\Phi}} : \Theta \times \Theta \to \mathbb{C}$ depending on the compression matrix $\boldsymbol{\Phi}$ via

$$(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) \mapsto \rho_{\boldsymbol{\Phi}}(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) = \boldsymbol{a}^{\mathrm{H}}(\boldsymbol{\theta}_1)\boldsymbol{\Phi}^{\mathrm{H}}\boldsymbol{\Phi}\boldsymbol{a}(\boldsymbol{\theta}_2), \tag{5.22}$$

The key idea in [188] is to measure how much the compression by means of $\boldsymbol{\Phi}$ affects the correlation function. To this end, we evaluate how much ρ deviates from $\rho_{\boldsymbol{\Phi}}$ by means of the function $\delta : \mathbb{C}^{m \times N} \to \mathbb{R}_0^+$ which is

defined as

$$\boldsymbol{\Phi} \mapsto \delta(\boldsymbol{\Phi}) = \int_{\Theta} \int_{\Theta} |\rho(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) - \rho_{\boldsymbol{\Phi}}(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)|^2 \, \mathrm{d}\boldsymbol{\theta}_2 \, \mathrm{d}\boldsymbol{\theta}_1, \tag{5.23}$$

where the integrations are carried out with respect to a suitable measure on Θ. If we have $\delta(\boldsymbol{\Phi}_1) \leqslant \delta(\boldsymbol{\Phi}_2)$, we can expect $\boldsymbol{\Phi}_1$ to be the better compression matrix than $\boldsymbol{\Psi}_2$, since the compressed correlation behavior $\rho_{\boldsymbol{\Phi}}$ more closely resembles the uncompressed ρ.

Hence, the function δ can be considered a valid proxy to evaluate the performance of a given compression matrix, where small values of δ are generally desirable. This is why we aim at solving

$$\min_{\boldsymbol{\Phi}} \quad \delta(\boldsymbol{\Phi}) \tag{5.24}$$

in order to recover an optimal compression matrix in terms of the SCF. However, already the evaluation of δ is computationally intractable, since it involves numerical integration for non-trivial data models $\boldsymbol{a}$, where the complexity scales exponentially in the parameter space dimension R. Hence, we need to formulate a suitable proxy objective function, which we do via $\eta : \mathbb{C}^{m \times M} \times \mathcal{A}^B \to \mathbb{R}_0^+$ defined as

$$(\boldsymbol{\Phi}, \boldsymbol{A}) \mapsto \frac{1}{B^2} \sum_{i=1}^{B} \sum_{j=1}^{B} \left| \boldsymbol{a}_i^{\mathrm{H}} \boldsymbol{a}_j - \boldsymbol{a}_i^{\mathrm{H}} \boldsymbol{\Phi}^{\mathrm{H}} \boldsymbol{\Phi} \boldsymbol{a}_j \right|^2, \tag{5.25}$$

where $B \in \mathbb{N}$ is the number of elements $\boldsymbol{a}_i \in \mathcal{A}$ we consider.

If we draw the columns in $\boldsymbol{A} = [\boldsymbol{a}(\boldsymbol{\theta}_1), \ldots, \boldsymbol{a}(\boldsymbol{\theta}_B)] \in \mathbb{C}^{N \times B}$ randomly and iid according to the uniform measure on Θ we have that the then random variable $\eta(\boldsymbol{\Phi}, \cdot)$ converges in expectation and in L^2 against $\delta(\boldsymbol{\Phi})$ for $B \to \infty$, which means we have that

$$\lim_{B \to \infty} \mathbf{E}(|\eta(\boldsymbol{\Phi}, \boldsymbol{A}) - \delta(\boldsymbol{\Phi})|) = 0 \tag{5.26}$$

and

$$\lim_{B \to \infty} \mathbf{E}\left[(\eta(\boldsymbol{\Phi}, \boldsymbol{A}) - \delta(\boldsymbol{\Phi}))^2 \right] = 0. \tag{5.27}$$

As such, the function η poses a valid stochastic proxy for the calculation of δ by randomly selecting from the atomic set $\mathcal{A}$. Additionally, we notice that for fixed $\boldsymbol{A}$ the function $\eta(\cdot, \boldsymbol{A})$ is smooth in its first argument, since we can rewrite it to

$$\eta(\boldsymbol{\Phi}, \boldsymbol{A}) = \frac{1}{B^2} \left\| \boldsymbol{A}^{\mathrm{H}} \boldsymbol{A} - \boldsymbol{A}^{\mathrm{H}} \boldsymbol{\Phi}^{\mathrm{H}} \boldsymbol{\Phi} \boldsymbol{A} \right\|_F^2. \tag{5.28}$$

Elaborating on this, we now formulate a Stochastic Gradient Descent (SGD) with momentum, where the general idea is to consider $\boldsymbol{A}$ as the so-called minibatch, which we generate randomly in each step as described above.

Generally, gradient descent with momentum [189] is a popular and simple first order technique to find local minima of smooth functions. In its most simple form it minimizes a smooth function $f : \mathbb{C}^n \to \mathbb{R}$ by iterating

$$\boldsymbol{v}_{i+1} = \beta \boldsymbol{v}_i - \alpha \nabla_{\overline{\boldsymbol{x}}} f(\boldsymbol{x}_i) \tag{5.29}$$

$$\boldsymbol{x}_{i+1} = \boldsymbol{x}_i + \boldsymbol{v}_i \tag{5.30}$$

for initial velocity and state variables $\boldsymbol{v}_0, \boldsymbol{x}_0 \in \mathbb{C}^n$, a drag parameter $\beta \in [0,1)$ and a step size $\alpha > 0$. Ideally, we would simply carry out the optimization in (5.24) using the iteration above.

In order to render the iteration stochastic we use new realizations of (5.28) in each step to formulate a randomized version of (5.29) and (5.30) as

$$\boldsymbol{V}_{i+1} = \beta \boldsymbol{V}_i - \alpha \nabla_{\overline{\boldsymbol{\Phi}}} \eta(\boldsymbol{\Phi}_i, \boldsymbol{A}_i) \tag{5.31}$$

$$\boldsymbol{\Phi}_{i+1} = \boldsymbol{\Phi}_i + \boldsymbol{V}_i \tag{5.32}$$

for initial velocity and state matrices $\boldsymbol{V}_0, \boldsymbol{\Phi}_0 \in \mathbb{C}^{m \times M}$. Through the combination of randomly varying $\boldsymbol{A}$ in each step and using momentum with drag, we always use a small proportion of the previous selections $\boldsymbol{A}$ for the next search direction.

Clearly, we are still in need of an analytical expression of the gradient of $\eta(\cdot, \boldsymbol{A}_i)$, which is provided in the following result.

Lemma 5.1. *For given $\boldsymbol{A}_i \in \mathbb{C}^{M \times B}$ and $\boldsymbol{\Phi} \in \mathbb{C}^{m \times M}$ it holds that*

$$\nabla_{\overline{\boldsymbol{\Phi}}} \eta(\boldsymbol{\Phi}, \boldsymbol{A}_i) = c \cdot (\boldsymbol{\Psi} \boldsymbol{A}_i \boldsymbol{A}_i^{\mathrm{H}} \boldsymbol{\Phi}^{\mathrm{H}} \boldsymbol{\Phi} \boldsymbol{A}_i \boldsymbol{A}_i^{\mathrm{H}} - \boldsymbol{\Phi} \boldsymbol{A}_i \boldsymbol{A}_i^{\mathrm{H}} \boldsymbol{A}_i \boldsymbol{A}_i^{\mathrm{H}})$$

for $c = 4/B^2$. ∎

The proof can be found in Appendix A.3. This allows us now to carry out measurement matrix optimization for any form of atomic set as summarized in Algorithm 5.2.

It is worth noting that the proposed Algorithm 5.2 does not depend on the specific structure of $\mathcal{A}$. Hence, it can not only be applied to the setting in Example 5.1, but to far more general atomic sets as long as the measurement process is linear. Additionally, one can even define areas of interest in Θ, by varying the distribution that generates the minibatches accordingly. Further, the definitions of ρ and $\rho_{\boldsymbol{\Phi}}$ in (5.21), and

Data: input: the mapping $\boldsymbol{a}$, initial velocity $\boldsymbol{V}_0$, initial compression matrix $\boldsymbol{\Phi}_0$;
mini-batch size $B \in \mathbb{N}$, learning rate α, drag parameter β, K steps;
[1] **while** $k < K$ **do**
[2] Draw $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_B$;
[3] Calculate $\boldsymbol{A}$;
[4] $\boldsymbol{V}_{i+1} = \beta \boldsymbol{V}_i - \alpha \nabla_{\overline{\boldsymbol{\Phi}}} \eta(\boldsymbol{\Phi}_i, \boldsymbol{A}_i)$;
[5] $\boldsymbol{\Phi}_{i+1} = \boldsymbol{\Phi}_i + \boldsymbol{V}_i$;
[6] $i = i + 1$;
[7] **end**
[8] return $\boldsymbol{\Phi}_K$;

Algorithm 5.2: *The proposed sensing matrix design algorithm allows for an efficient implementation independent of the specific structure of $\mathcal{A}$.* – Iteration procedure for the proposed SGD-based algorithm to construct $\boldsymbol{\Phi}$.

(5.22) respectively, can be extended to more general inner-product spaces, if for instance $\mathcal{A}$ is the set of rank-1 matrices. Last, the computational effort can easily be tuned either by selecting the size of the mini-batches B or the number of iterations K.

As a numerical sanity check, in Figure 5.11 we show that the proposed algorithm converges reliably to an orthogonal matrix, when we set $m = M = 144$, so we employ no compression and the optimal sensing matrix is a matrix that satisfies $\boldsymbol{\Phi}^{\mathrm{H}}\boldsymbol{\Phi} = \boldsymbol{I}_M$. In the simulations we use $\boldsymbol{a} = \boldsymbol{f}_{12,12}$ from Equation (5.7). We initialize Algorithm 5.2 with a random Gaussian matrix and evaluate $\eta(\boldsymbol{\Phi}, \boldsymbol{A})$ each 200 steps when running with $\beta = 0.95$ and $\alpha = 10^{-5}$. We use a mini-batch size of $B = 25$. We plot the values of η versus the runtime up until this point for 25 trials. After a settling phase the iterations quickly show an exponential convergence speed to 0, which is the optimal value in this case.

Next, we study more closely what the algorithm can achieve for the estimation performance in a DoA estimation setting as given in Example 5.1.

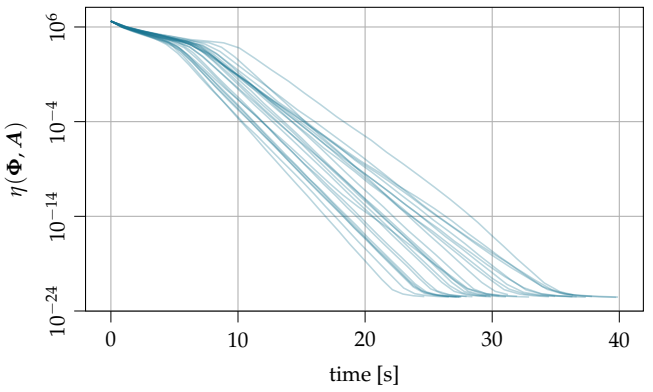


Figure 5.11 *The proposed SGD converges to an orthogonal matrix up to the numerical precision when possible.* – Value of the objective function $\eta(\boldsymbol{\Phi}_i, \boldsymbol{A})$ in iteration i given in (5.28) for fixed and a sampled and fixed dictionary $\boldsymbol{A}$ versus the runtime of the iteration. ∎

5.3.3 Application to DoA Estimation

In this section, we assess the performance of the resulting compressive antenna arrays when designed using our proposed SGD-based approach in comparison with that of the uncompressed array, the spatially compressed array obtained using the 2D SCF-based approach from [188] and also the compressive array derived from a randomly drawn combining matrix as motivated by Section 2.4.1.1. To put the approach in context, we aim at optimizing the matrix $\boldsymbol{\Xi} \in \mathbb{C}^{m \times P}$ as introduced in (5.6).

Influence on the Correlation Function

In Figure 5.12 we show how the correlation function of an antenna array changes due to the optimization process when initialized with a random Gaussian matrix. We test the algorithm with an $8x8$ Patch Uniform Rectangular Array (PURA), which is modeled based on real measurement data from an anechoic chamber and we use the EADF to evaluate the beampattern for arbitrary angles. We let the SGD run for 10^4 steps with a learning rate of $\alpha = 10^{-5}$, $\beta = 0.9$ and mini-batch size of 20. We compress the 64 ports down to 32. In Figure 5.12 we plot the auto-correlation functions $\rho(\boldsymbol{\theta}_0, \cdot)$ and $\rho_{\boldsymbol{\Phi}}(\boldsymbol{\theta}_0, \cdot)$, where $\boldsymbol{\theta}_0 = [\pi/2, 0]^{\mathrm{T}}$ corresponds to the antenna's main-beam direction. As we can see, although we compress the antenna outputs to only half of the data, the correlation

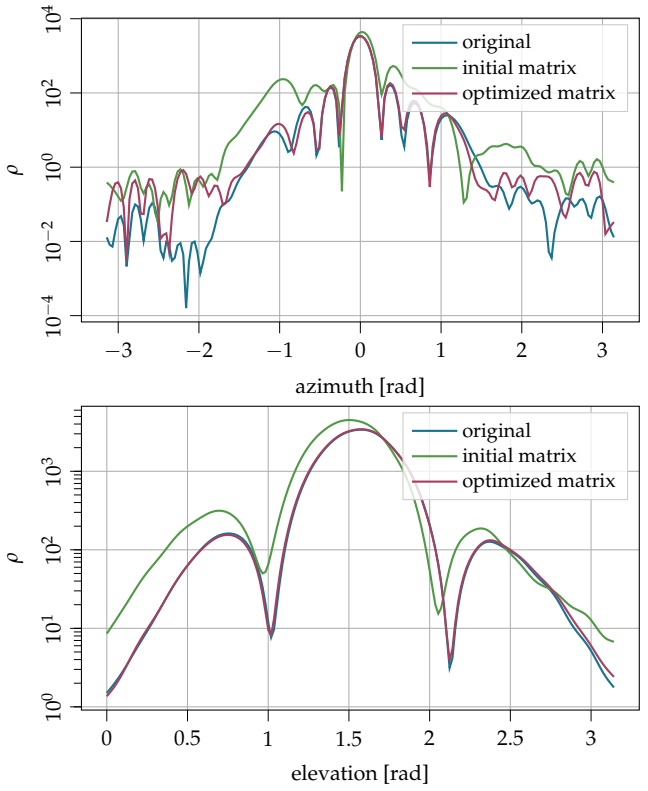


Figure 5.12 *The proposed SGD-based approach can replicate the uncompressed auto-correlation function and hence reduce sidelobes.* – Auto-correlation function along the co-elevation-$\pi/2$-cut (top) and the azimuth-0-cut (bottom). ■

function stays mainly intact after the optimization and compared to the unoptimized random compression, the sidelobes both in azimuth and elevation direction can be reduced substantially. In this case, we draw the random matrix from a so-called complex Rademacher distribution, where each entry's real and imaginary part are drawn from a Rademacher distribution.

In Figure 5.13 we evaluate the naturally 4D-SCF for the $2D$ slice where co-elevation $\vartheta = \pi/2$. Here, we use the same output matrices and settings as in Figure 5.12. As one can see, the random compression deteriorates the auto-correlation behavior of the antenna, whereas the optimized matrix resulting from the proposed SGD does follow the original correlation function very closely, especially around the first few sidelobes.

Based on both of these simulation results we can expect that the compressed beampattern can indeed deliver similar performance to the uncompressed one, which we will further verify below.

Influence on the Objective Function and Estimation Performance

After studying the effect of the SGD on the error-metrics we used to serve as proxies for parameter estimation, we now turn to the assessment of the estimation performance. This, we address by studying the resulting values of the objective function in (5.24) and the deterministic CRB by means of numerical simulations. This is a suitable proxy for any estimation algorithm, since we can assume that any viable estimator has to behave asymptotically as the CRB.

To this end, we let SGD run for $K = 5000$ steps, with $B = 250$ angles per step, step size $\alpha = 10^{-2}$ and drag parameter $\beta = 0.1$, where the distribution of the $\boldsymbol{\Theta}_k$ is the uniform distribution on $(0, 2\pi] \times [\pi/4, 3\pi/4]$. Moreover, we always use normalized sensing matrices, which means that the columns of any $\boldsymbol{\Phi}$ considered are normalized to unit length. As an antenna we consider a Stacked Uniform Circular Array (SUCA) of $(\Sigma \times P_S)$ isotropic elements, so it has $\Sigma = 3$ stacks of $P_S = 11$ elements each with the total number of elements denoted by $P = \Sigma \times P_S$. The array response of the SUCA is given by $\boldsymbol{a}_{\mathrm{SUCA}}(\theta, \vartheta)$. We choose $d = 0.5\lambda$ as the distance between two consecutive stacks, $r = 0.68\lambda$ as the radius of the stacks, where $\lambda = c/f$ is the wave-length at frequency f with the speed of light $c \approx 3 \cdot 10^8$ m/s.

For the SCF-based approach from [188], the number of grid points in azimuth and elevation used for calculation of the SCF is $N_\theta = 121$ and $N_\vartheta = 61$, respectively over $\theta \in [-\pi, \pi]$ and $\vartheta \in [-\pi/2, \pi/2]$. Also, the

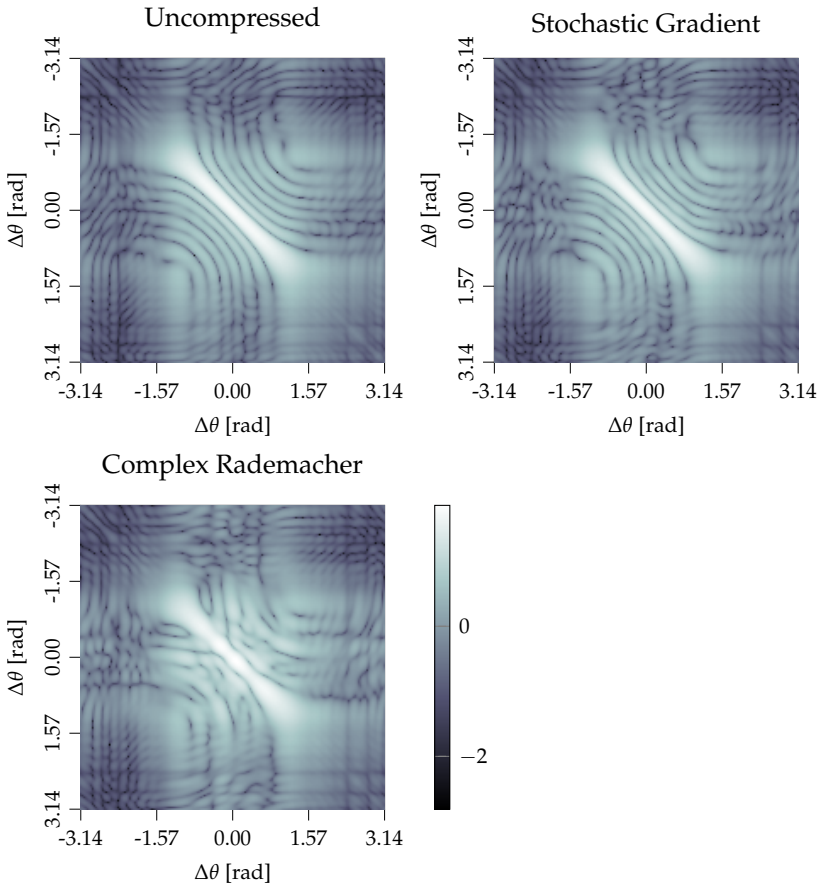


Figure 5.13 *The proposed SGD-based approach can replicate the 2D auto-correlation function.* – The absolute values of $2D$ auto-correlation function for the co-elevation-$\pi/2$-cut on a logarithmic scale for $\Delta\theta = [\pi/2, \Delta\varphi]^{\mathrm{T}}$ with $\varphi \in [] - \pi, +\pi]$. ■

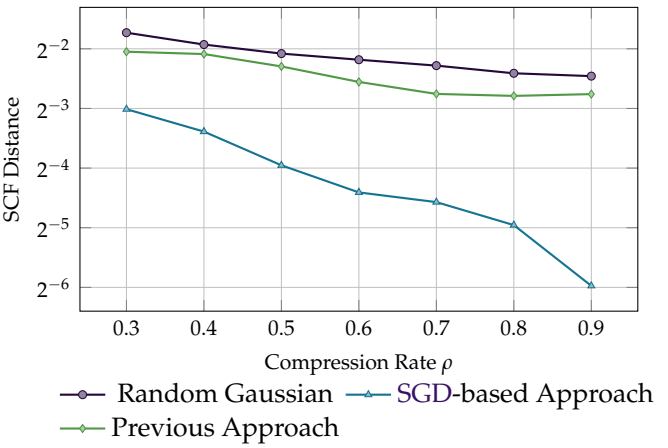


Figure 5.14 *The presented SGD-based method reaches lower values for the objective function* – Distance of the SCFs generated by the different approaches for varying levels of compression. ■

number of reference points in elevation considered for optimization using this approach is $|\mathcal{N}| = 3$, see [188] for more details.

In Figure 5.14 we evaluate how the different approaches perform in terms of the overall SCF-error. To approximate the overall error after optimization, we evaluate the SCF on a regular 2D grid in azimuth and elevation and then sum the squared absolute values of these samples. This is done for different levels of compression rate ρ. Clearly, the proposed SGD-based method is capable of approximating the original antenna response more closely than the previous approach and the conventional combining matrices resulting from a zero-mean Gaussian ensemble.

To quantify the performance of the proposed design approach for the case of 2D DoA estimation, we also evaluate the deterministic CRB in Figure 5.15, since it serves as a proxy to assess the possible performance of any unbiased estimator. For instance, the maximum likelihood estimator always reaches this lower bound asymptotically in the effective SNR, so one can expect that any well designed estimation procedure behaves similarly in this regime. With spatial compression, the deterministic CRB for the 2-dimensional case with S sources and 1 snapshot can be computed via Equation (2.25) and reads as

$$C(\boldsymbol{\theta}) = \frac{\sigma^2}{2} \operatorname{tr}\left(\left[\Re(\boldsymbol{D}^{\mathrm{H}} \boldsymbol{\Pi}_{\boldsymbol{G}}^{\perp} \boldsymbol{D} \odot (\mathbf{1}_{2\times 2} \otimes \hat{\boldsymbol{R}})^{\mathrm{T}}) \right]^{-1} \right), \tag{5.33}$$

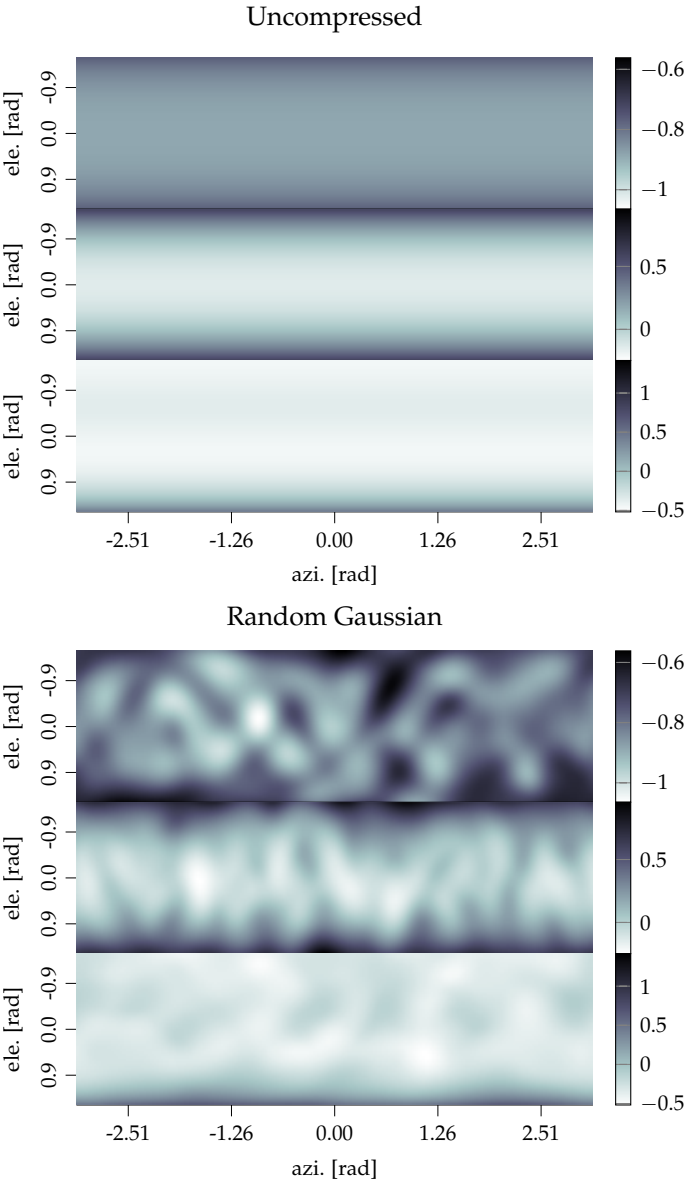
with $\boldsymbol{\Pi}_{\boldsymbol{G}}^{\perp} = \boldsymbol{I} - \boldsymbol{G}(\boldsymbol{G}^{\mathrm{H}}\boldsymbol{G})^{-1}\boldsymbol{G}^{\mathrm{H}}$, $\otimes$ denoting the Kronecker product, $\hat{\boldsymbol{R}} = \boldsymbol{x} \cdot \boldsymbol{x}^{\mathrm{H}}$ being the sample covariance and we have set

$$\boldsymbol{G} = \boldsymbol{\Phi}[\boldsymbol{a}_{\mathrm{SUCA}}(\theta_1, \vartheta_1), \ldots, \boldsymbol{a}_{\mathrm{SUCA}}(\theta_S, \vartheta_S)],$$
$$\boldsymbol{D}_i = \frac{\partial}{\partial \boldsymbol{\theta}_i}\boldsymbol{G},\ \boldsymbol{D} = [\boldsymbol{D}_1, \boldsymbol{D}_2]\,.$$

It is worth noting here that the analytical evaluation of the $\boldsymbol{D}_i$ is enabled by the EADF[174, O7].

The results in Figure 5.15 show for a fixed noise level $\sigma^2 = 1$ how the CRB changes for the four different sensing matrix designs depending on the position of a single source (top), two sources separated in azimuth (middle) and elevation (bottom). In the two sources case, the first source is located at the position denoted in the plot and the second with angular distance $2\pi/10$ in azimuth or elevation respectively. As one can see, the random combining matrix and the previous approach introduce a highly varying sensitivity of the CRB with respect to azimuth and elevation, rendering the resulting combining matrices unsuitable to be applied for DoA estimation because of this non-uniformity in the angular domain. The SGD-based approach results in a significantly smoother behavior of the CRB with a uniform behavior across the whole azimuth and elevation region. Thus, the compressed array resulting from SGD mimics the behavior of the uncompressed array more closely in terms of the CRB, which ultimately was the goal of the proposed design process, since this allows a more predictable behavior of the system over the parameter domain Θ.

In the next and final analysis we are going to combine both the design of the measurement matrices and the recovery algorithm to study the performance of the resulting compressive processing pipeline.


Uncompressed
ele. [rad]
-0.9
0.0
0.9
ele. [rad]
-0.9
0.0
0.9
ele. [rad]
-0.9
0.0
0.9
−0.6
−0.8
−1
0.5
0
1
0.5
0
−0.5
-2.51
-1.26
0.00
1.26
2.51
azi. [rad]
Random Gaussian
ele. [rad]
-0.9
0.0
0.9
ele. [rad]
-0.9
0.0
0.9
ele. [rad]
-0.9
0.0
0.9
−0.6
−0.8
−1
0.5
0
1
0.5
0
−0.5
-2.51
-1.26
0.00
1.26
2.51
azi. [rad]

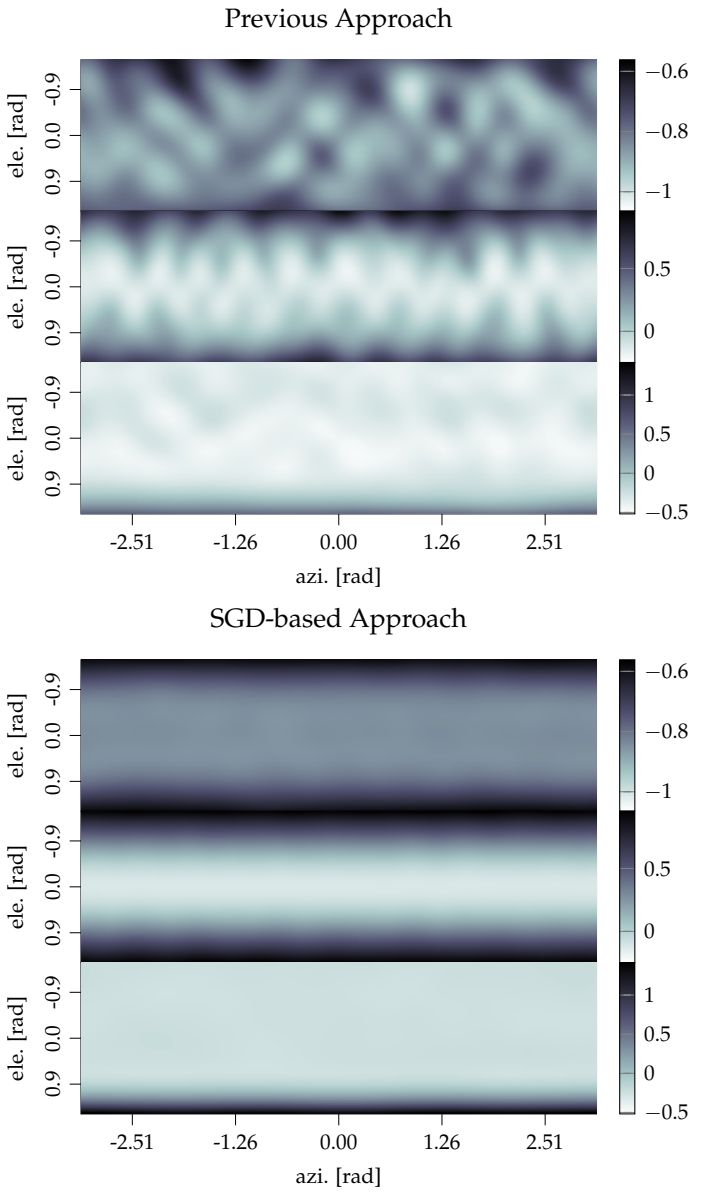


Figure 5.15 *The SGD-based approach delivers a very uniform sensitivity pattern for the compressed array* – Magnitude of the deterministic CRB on a logarithmic scale for fixed noise level $\sigma^2 = 1$ dependent on azimuth and elevation. *Top*: single source, *Middle*: two sources separated in azimuth by $2\pi/10$, *Bottom*: two sources separated in elevation by $2\pi/10$. Notice the differing color bars for top, middle and bottom. ■

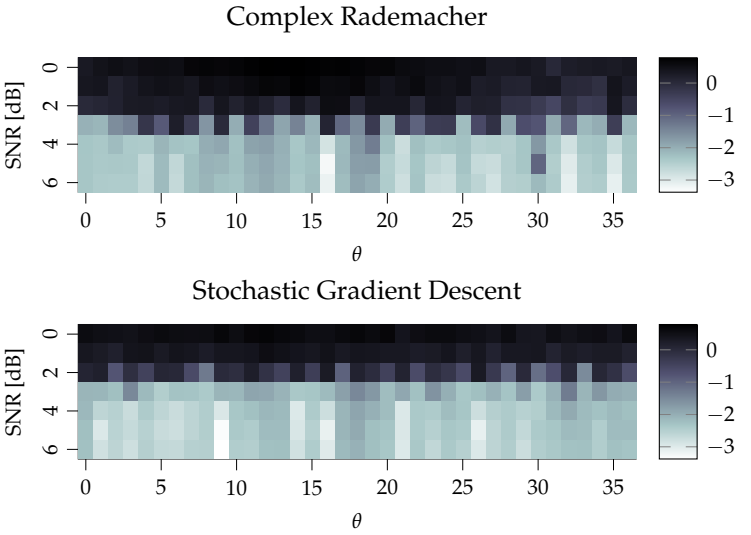


Figure 5.16 *The combination of SGD and ADMM enables a uniform and predictable recovery behavior over the parameter space.* – The mean squared error (MSE) on a logarithmic scale of a 2-sparse scenario for different levels of SNR and parameter configurations. ■

5.3.4 Combination of SGD and ANM

Up until this point, we have presented an efficient algorithm for multi-dimensional line spectral estimation in Section 5.2.6 and a versatile optimization algorithm to construct well behaved measurement matrices in Section 5.3. So finally, we would like to combine both grid-free optimization approaches for the measurement matrices and the one for the sparse signal recovery. To this end, we carry out several empirical studies to study the effect of combining these two crucial steps of a grid-free compressed sensing setup.

5.3.4.1 One-dimensional Line Spectral Estimation

As a first scenario, we consider a 2-sparse setup where we have $N = 64$ observations which we compress to $m = 48$ and we use $K = 20$ independent snapshots. We compare the results of a fixed columns normalized complex Rademacher matrix $\mathbf{\Phi}_r \in \mathbb{C}^{a \times b}$ with entries in $\{+1, -1, +\jmath, -\jmath\}$ and the output $\mathbf{\Phi}_s$ of Algorithm 5.2, where we use the given Rademacher ma-

trix $\mathbf{\Phi}_r$ as initialization. We keep the two sources at distance $d = \pi \cdot 8/5$ and move them in parallel over the range $[-\pi, +\pi]$ to cover the whole normalized frequency range. At each source configuration, we average the reconstruction error over $T = 15$ independent noise realizations and run Algorithm 5.1 for $S = 10$ steps with $\rho = 0.05$ and $\tau = \sigma^{0.8}$, where σ^2 is the noise variance of the current snapshot.

The MSE results depicted in Figure 5.16 suggest that the combination of SGD and ADMM delivers a smoother and more predictable performance over the parameter regime than the random ensemble. Additionally, we see that the phase transition in terms of the SNR happens slightly sooner for the optimized matrix. Hence, we can conclude that the combination of the proposed SGD and ADMM can serve as a robust pipeline for this parameter estimation task.

As a second line spectral estimation scenario, we consider a 2-sparse setup where we have $M = 64$ observations which we compress to $m = 56$. We compare the results of a fixed columns normalized complex binary Rademacher matrix $\mathbf{\Phi}_r \in \mathbb{C}^{a \times b}$ with entries in $\{+1, -1, +\jmath, -\jmath\}$ and the output $\mathbf{\Phi}_s$ of Algorithm 5.2 where we use the given Rademacher matrix $\mathbf{\Phi}_r$ as initialization. Instead of moving the sources in a fixed distance, we draw the first source's position randomly from the uniform distribution on $[-\pi, +\pi]$ and then put the second source at distance d to it, which is varied along the horizontal axis. At each source configuration, we average over $T = 20$ noise realizations and run Algorithm 5.1 for 10 steps with $\rho = 0.05$ and $\tau = \sigma^{0.8}$, where σ^2 is the noise variance of the current snapshot.

As the results for the estimation error in terms of the MSE in Figure 5.17 indicate, the optimization of $\mathbf{\Phi}_r$ to $\mathbf{\Phi}_s$ allows to resolve more closely spaced sources, since the MSE stays lower for small source separations and also the phase transition when the employed ADMM starts working for large separations is happening in a lower SNR regime. Finally, the errors in terms of the MSE are considerably lower for the matrices originating from the SGD-based method.

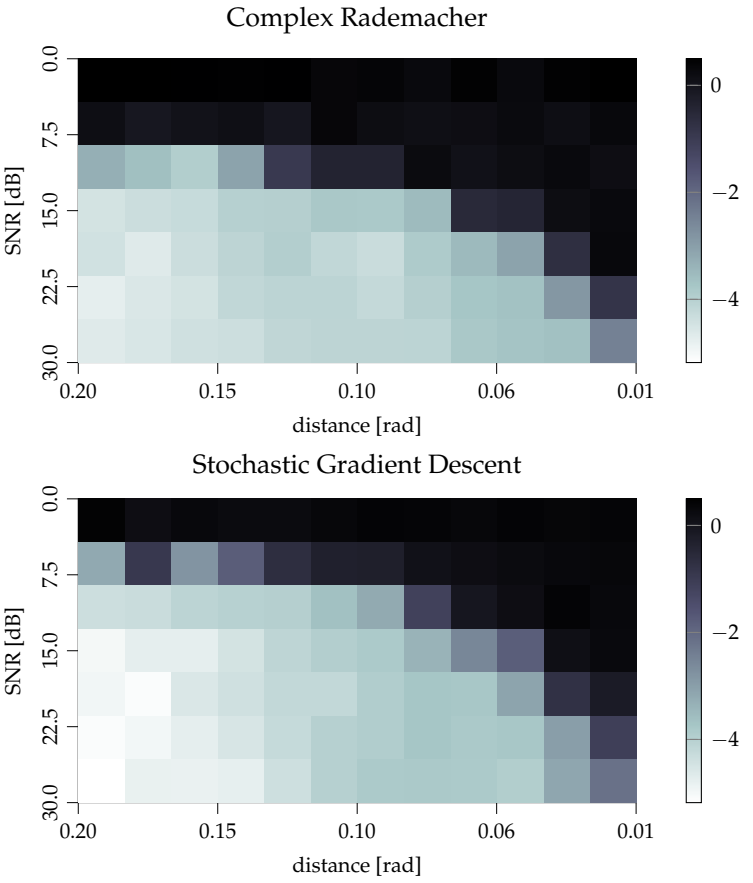


Figure 5.17 *The combination of SGD and ADMM enables to resolve more closely spaced sources than ADMM with un-optimized matrices.* – MSE on a logarithmic scale for 1-dimensional line spectral estimation for the two compression schemes when compressing $M = 64$ samples to $m = 56$ averaged of 20 realizations per SNR and distance of $S = 2$ sources. ■

5.3.4.2 Two-dimensional Line Spectral Estimation

As a straightforward consequence of the previous sections, we wish to evaluate the performance of the matrix optimization with SGD and the ADMM for the case $R > 1$, since technically, both methods can cope with this case. To this end, be carry out similar simulations as in Section 5.3.4.1.

In Figure 5.18 we show simulation results for a 2-sparse scenario, where we again vary the distance of the two sources and the SNR level. We let the SGD run starting with the complex Rademacher matrix we used as a comparison and optimize it with mini-batch sizes of $B = 15$ and let it run for $1.5 \cdot 10^4$ steps with a learning rate of $\alpha = 10^{-6}$ and momentum drag of $\beta = 0.95$. We average the results over 50 trials per combination of distance and SNR, keeping the measurement matrices fixed, and each snapshot consists of $K = 50$ observations. For these, we let the ADMM run for 10 steps, where we use $\tau = \sigma^{0.8}$ and $\rho = 0.05$. Here, σ denotes the respective noise variance.

As we can see, the optimized matrices have their SNR dependent phase transition in the more noisy regime. Also, the achieved MSE is lower as well in the region where both measurement matrices produce meaningful results. Hence, one can claim that the combination of the proposed SGD and the ADMM allows to decrease the reconstruction error during parameter estimation also for the case when $R = 2$.

5.3.4.3 One-dimensional DoA Estimation

We carry on with the case, where we can only optimize $\boldsymbol{\Xi}$ of the measurement matrix $\boldsymbol{\Phi} = \boldsymbol{\Xi} \cdot \boldsymbol{\Gamma}$. We setup a 1-dimensional DoA estimation for azimuth estimation with a synthetic 13×13 Uniform Rectangular Array (URA), where each element is a vertically polarized Hertzian dipole with length $\lambda/2$ for a frequency $f = 3.50\,\mathrm{GHz}$. Also, the elements are spaced at $\lambda/2$ distance such that the spatial sampling frequency does not violate the Nyquist criterion to avoid any angular ambiguities.

As a first numerical study for DoA estimation, we consider a 2-sparse scenario, where we again compare the two compression strategies in term of the MSE but use the uncompressed antenna output as a benchmark. To this end, we let the SGD run for $16 \cdot 10^3$ steps with minibatch size $B = 15$ and learning rate 10^{-7} and drag parameter $\beta = 0.95$. We compress the $P = 169$ ports down to $m = 81$ measurements and run Algorithm 5.1 for the three different compression strategies using just 5 steps of ADMM to keep the simulations in a reasonable time frame. One can expect that

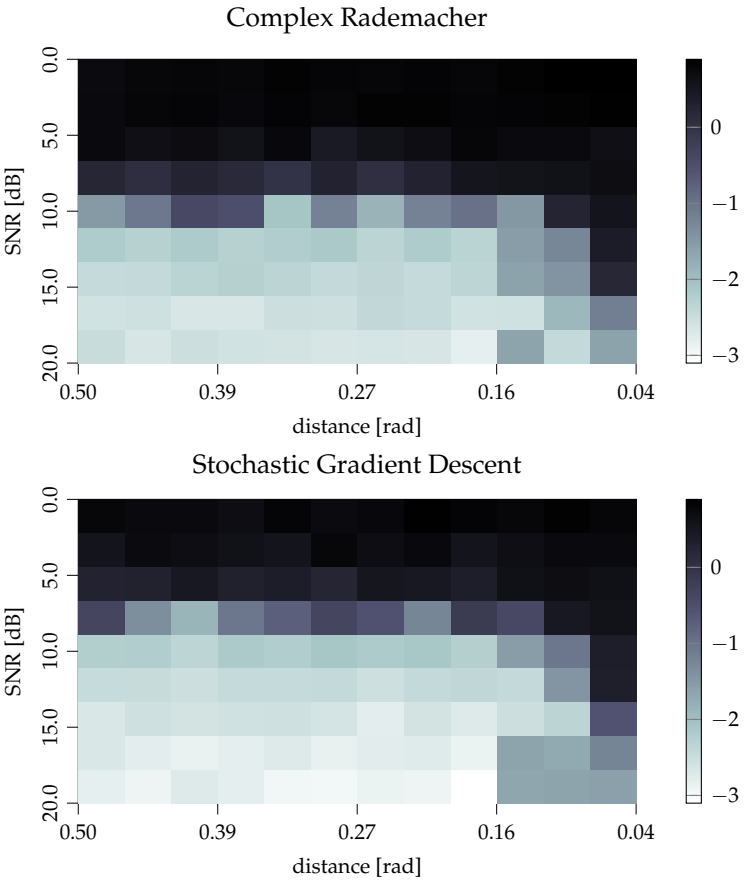


Figure 5.18 *The combination of SGD and ADMM enables to resolve more closely spaced sources than ADMM with un-optimized matrices.* – Comparison of MSE on a logarithmic scale for the 2-dimensional line spectral estimation resulting from the SGD-optimized compression matrices and the complex Rademacher ensemble. ■

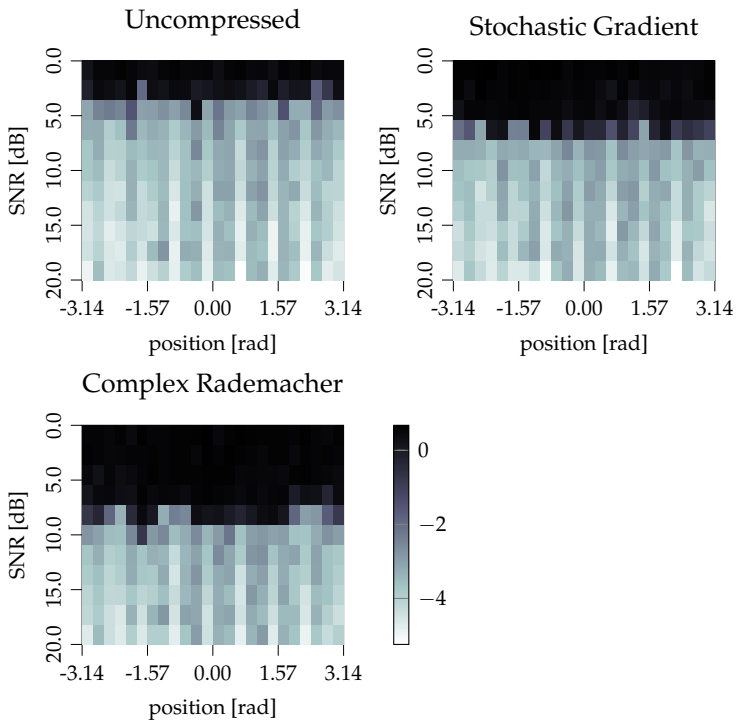


Figure 5.19 *The sensing matrix optimization using SGD combined with ADMM allows to reliably approximate the estimation performance of the uncompressed scenario.* – Comparison of the MSE during azimuth estimation using a 13×13 uniform rectangular array on a logarithmic scale for the two compression strategies compared to the uncompressed scenario. ■

similarly to before that more iterations slightly improve the resulting estimation error even further.

In this scenario, we chose the ground truths such that the first sources' positions cover the whole azimuth range and the second one is placed at fixed distance $\pi/16$. For each position and level of SNR we average the results over 25 noise realizations. As we can see in Figure 5.19 the proposed SGD measurement matrices allow the ADMM to reduce the MSE significantly, placing its performance between the results of the Rademacher measurements and those of the uncompressed observations. Hence, the results from Section 5.3.3 carry over to the estimation process when we employ the proposed ADMM algorithm.

5.3.4.4 Combined Delay- and DoA-Estimation

As a final demonstration we are going to combine the setups in Example 2.2 and Example 5.1 to a joint estimation of the delay and azimuth parameters. As such, we consider a simplified version of the estimation carried out in [161]. To this end, we assume that we have access to measurements of an antenna array for a set of frequencies covering a certain band $[f_1, f_2] \subset \mathbb{R}^+$. We assume that the bandwidth $f_2 - f_1$ is low enough compared to $(f_2 - f_1)/2$ such that we can treat the array manifold frequency independent. Hence, we have access to measurements of the form $\boldsymbol{H} \in \mathbb{C}^{N_f \times N_a}$, where N_f is the number of frequency samples we acquire and N_a is the number of individual elements the antenna is equipped with.

In order to recover the underlying parameters of the channel impulse responses one usually assumes a parametric model. In our case the model originates from the assumption that we have multiple planar waves impinging at the antenna, where each wave is parametrized by the delay within a single impulse-response and an angle with respect to the antenna coordinate system. Such a set of parameters is called a single specular path, see [190]. Hence, following the channel model presented in [162], where we make some simplifications we end up with a model for $\boldsymbol{H}$ as

$$\operatorname{vec}(\boldsymbol{H}) = \sum_{s=1}^{S} c_s \left(\boldsymbol{f}_{N_f}(2\pi\tau_s) \diamond \boldsymbol{a}(\varphi_s) \right) + \boldsymbol{N}, \tag{5.34}$$

where $\boldsymbol{f}$ is defined in (5.5). Here, c_s denote the complex path weights, $\tau_s \in [0, 1]$ the normalized delays and φ_s the azimuth angles of the impinging waves. Also, $\boldsymbol{N} \in \mathbb{C}^{N_a \cdot N_f}$ models additive measurement noise.

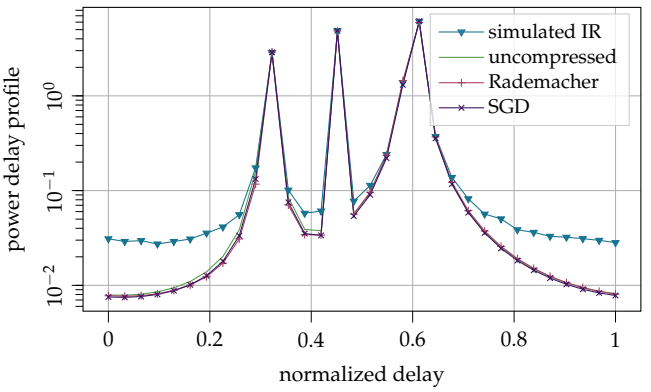


Figure 5.20 *The flexibility of the proposed ADMM allows to estimate delay shifts and incidence angles from samples in the frequency and spatial domains.* – Simulated and estimated IR at 20 dB SNR using a synthetic 13×13 URA. ■

Interestingly, we can use the derivations about the antenna beampatterns in Equation (5.6) and the definition of $\boldsymbol{f}$ in (5.8) to reformulate (5.34) to

$$\operatorname{vec}(\boldsymbol{H}) = \sum_{s=1}^{S} c_s (\boldsymbol{I}_{N_f} \otimes \boldsymbol{\Gamma}) \boldsymbol{f}_{N_f,N_e}(2\pi\tau_s, \varphi_s) + \boldsymbol{N}, \tag{5.35}$$

where $\boldsymbol{\Gamma} \in \mathbb{C}^{N_a \times N_e}$ is derived via the EADF of the receiving antenna. Additionally, if we employ the same compression as in (5.6), we observe a spatially compressed version of the channel impulse response as

$$\operatorname{vec}(\boldsymbol{H}_c) = \sum_{s=1}^{S} c_s (\boldsymbol{I}_{N_f} \otimes (\boldsymbol{\Xi} \cdot \boldsymbol{\Gamma})) \boldsymbol{f}_{N_f,N_e}(2\pi\tau_s, \varphi_s) + \boldsymbol{N}_c,$$

This allows us to conclude that the observation model for $\boldsymbol{H}_c$ essentially corresponds to a special version of (5.7), where $\boldsymbol{\Phi} = \boldsymbol{I}_{N_f} \otimes (\boldsymbol{\Xi} \cdot \boldsymbol{\Gamma})$. Hence in principle, we can use a 2-dimensional version of ADMM to extract the parameters of $\boldsymbol{H}_c$.

Summarizing, the procedure during the numerical simulations is as follows. We optimize $\boldsymbol{\Xi}$ using the presented SGD algorithm for the employed antenna array manifold $\boldsymbol{a}$ to get $\boldsymbol{\Xi}$ and then use $\boldsymbol{\Phi}$ as above to run the ADMM algorithm.

To produce the results in Figure 5.20 we use a synthetic $13x13$ URA with Hertzian dipole elements designed for an operating frequency $f = 3.5$ GHz and $N_f = 32$ frequency bins in complex baseband. For the

aperture of this antenna we use $N_e = 83$ Fourier coefficients to represent the beampattern in azimuth for co-elevation $\vartheta = \pi/2$. We simulate measurements in frequency domain for an SNR of 20 dB and used $K = 5$ snapshots to estimate the normalized delay and angle parameters for $S = 3$ paths. As compression matrix we choose $\boldsymbol{\Xi} \in \mathbb{C}^{118\times 169}$, which corresponds to a data reduction to about 70%.

We can see that by using the datamodel in (5.34) together with Algorithm 5.1 it is indeed possible to recover sparse channel impulse responses from spatially compressive and frequency measurements with non-trivial array elements.

5.3.5 Summary

Motivated by the previous results, we can conclude that the proposed SGD is capable of improving the measurement process for a wide variety of parametric models while being a computationally attractive method. The correlation behavior of compressed antenna arrays can be improved by following the uncompressed correlation function very closely, hence reducing sidelobes compared to the unguided compression using random matrices. Also, for the simpler line spectral estimation case, one can infer that the resulting compression matrices perform uniformly well over the parameter space and hence yield more predictable reconstruction behavior. We can conclude that we have successfully contributed to AOI-C.

Both of these observations also hold true when we combine the proposed SGD with the previously derived ADMM-based estimation routine both for line spectral estimation as well as DoA estimation and joint delay and DoA estimation. In these cases, the optimized matrices improve upon the overall estimation performance as well as the predictability of the estimation error, since the fluctuation of the estimation error follows the error of the uncompressed case more closely.

As further investigations, one should study the effect of the choice for the random distribution that generates the minibatches $\boldsymbol{A}$ on the convergence of the SGD and the resulting performance of the ADMM. Additionally, we should be able to account for polarimetric antenna models during the design of $\boldsymbol{\Xi}$, which due to the flexibility of Algorithm 5.2 should be a straightforward extension.

5.4 Summary

As we show, the presented generalized line spectral estimation problem can be solved by means of a general ADMM implementation that can deal with arbitrary parameter space dimensions, measurement matrices and number of snapshots. This can directly be used for DoA estimation. In which case the compression matrix is structured, where the fixed part is determined by the antenna's array manifold. Hence, in this setting we also demonstrate how to incorporate accurate antenna models into the calculations and hence address AOI-M.

We see in the numerical simulations in Section 5.2.6 that the proposed ADMM can achieve the CRB in certain scenarios and it is able to exploit the properties of the improved measurement matrices we provide by means of the SGD iteration. This way, we present a rather general framework for measurement design and parameter estimation in this sparse and grid-free setting by means of atomic norm minimization and directly contribute to AOI-C and AOI-R.

The observant reader might have noticed that during the numerical simulations we do not carry out comparisons to other sparsity-exploiting algorithms except CVX in a simple 1D DoA setting from a single snapshot in Figure 5.10. This is due to the fact that none of the existing sparsity based approaches outlined in Section 5.1 does directly allow the incorporation of multiple snapshots and arbitrary compression matrices into the reconstruction process. Moreover, it is sufficient to carry out comparisons to the CRB, since it serves as a universal estimation algorithm benchmark. Finally, the viability of the proposed SGD-based optimization schemes for the measurement process can be inferred from the improved reconstruction performance using ADMM, since it is reasonable to expect that other sparsity based algorithms are influenced the same way by the optimized sensing matrices as the herein used ADMM is.

However, executing the whole processing pipeline is still a cumbersome task, since the number of unknowns during the iteration scales exponentially in the parameter space dimension R and hence, the projection step in Line 8 in the iteration of Algorithm 5.1 poses a bottleneck for $R \gg 1$. For instance, if we were to apply this framework in the settings discussed in Chapter 4 we would not be able to carry out the necessary calculations on readily available computing hardware. Additionally, we also notice in the numerical simulations that the first few iterations of the ADMM in Algorithm 5.1 make substantial progress towards a suitable covariance estimate in the sense that ESPRIT produces meaningful re-

sults. Whereas the iterations > 10 only accomplish minor improvements to the algorithms output. It would be interesting to derive more refined results about the speed of convergence of ADMM in the spirit of (2.17). Consequently, the search for efficient sparsity-based algorithms that also perform well in the very challenging grid-free scenarios is not over yet.

Chapter 6

Conclusion

"Was this what we hoped for?
It's not what we hoped for!
It's not!"

George Pettit
Dallas Green

To conclude this thesis, we first revisit the four AOI we introduced in Section 1.2 and summarize the previously presented contributions with respect to these. Second, after having worked through the previous material, we are able to phrase the interrelation of these four AOI, which we tackle in Section 6.2. Finally, we give an outlook on future research that could use the presented material as a motivation or starting point.

6.1 Addressing the AOI

In order to differentiate between the efforts made in the presented material we introduced four distinct but highly interrelated AOI, namely AOI-C, AOI-H, AOI-R and AOI-M. After churning through the topics covered in the previous chapters we want to reiterate on these individually and evaluate the obtained results in this context.

Compression (AOI-C)

In Section 2.5 we show that by employing appropriate sensing matrices one can infer the sparsity order of a signal from a single snapshot prior to reconstruction by means of rank estimation. We demonstrate that this approach can also be improved further by optimizing the sensing matrices that have to adhere a Vandermonde structure, for which we also derive explicit bounds for the coherence.

Utilizing the results from Section 2.4.2.2 we can show in Section 4.1 that for the task of ultrasound based non-destructive testing randomly subselected Fourier matrices allow stark compression when utilized together with an appropriate volumetric model and a matrix-free reconstruction scheme. Additionally, this compression also adheres constraints posed by hardware implementations.

As developed in Section 5.3, compression matrices employed in a parameter estimation setting can be tuned to deliver a uniform reconstruction performance over the whole parameter space by means of a stochastic first order momentum method. As we show, this method can be applied readily to very general parametric data models that go beyond the models discussed in this work.

Hardware (AOI-H)

The Fourier measurements employed in Section 4.1 are directly motivated by hardware considerations that allow to acquire samples immediately in analogue domain before digitization. Since these proposed matrices deliver good performance during reconstruction in terms of defect localization and computational complexity, we can conclude that the proposed Fourier measurements adhere to hardware constraints in terms of architecture complexity while still maintaining desirable properties for reconstruction. This is backed up by the derived scaling laws for the number of measurements and deterministic error estimates for the localization of defects.

The RD-based measurement design proposed in Section 4.2 that allows for a more efficient signal acquisition than the state of the art T&H-based methods is also motivated by properties of the already existing hardware platforms. This follows from the possibility to generate the mixing sequences used by the RD from a linear feedback shift register and an integrator completely in the analogue domain. Hence, this yields a compressive sensor that can be realized entirely in the analog domain, since the compression has to take place before digitization.

Generally, one can see by the significantly different approaches for the two applications that indeed individual care has to be taken in the development of the specific sampling architecture. In other words, different applications and different goals therein will ultimately lead to completely different hardware realizations. However, both of which are motivated and driven by the same compressive sampling paradigm.

Reconstruction (AOI-R)

We have shown in Chapter 3 that it is indeed beneficial to derive and then exploit structure in the involved linear mappings to render the reconstruction stage as efficient as possible. To stress this even more, in Section 4.1 we show that this matrix-free view on linear mappings is crucial to apply CS in this specific application. However, the sections in Section 4.1 also show that this matrix-free representation needs additional attention when employed in practice, since some methods for sparse reconstruction require additional knowledge about the linear mappings.

Additionally, Section 4.1.7 shows that the choice of the employed SSR algorithm makes a difference when it comes to imaging quality, since the algorithms produce different sparse vectors as their solutions and depending on the application's intend either can be favorable.

In the grid-free CS setting, we show in Section 5.2.6 how to derive and utilize a general purpose algorithm for R-dimensional line spectral estimation from compressive measurement comprising of multiple snapshots. Moreover, this algorithm can readily be combined with the proposed matrix optimization routine presented in Section 5.3 resulting in a grid-free sensing and reconstruction pipeline.

Models (AOI-M)

In Section 2.5, we show that the estimation of the unknown sparsity order prior to reconstruction is not only possible but does also substantially improve reconstruction accuracy when used appropriately together with an algorithm that uses the sparsity of the signal as a hyper-parameter.

The well resolved defect positions in the results in Section 4.1.7 could only be attained by utilizing a volumetric data model. As such, this 3D description is crucial for the acquired measurements and their spatial dependency accurately and this has a direct impact on the reconstruction quality and the necessary number of compressive measurements.

Last, when the beampattern of an antenna is described as a Fourier series one can rephrase the atomic norm minimization problem as a generalized line spectral estimation problem that can be solved efficiently.

This in turn allows direction finding with realistic antenna arrays. Hence, one can greatly extend the scenarios where grid-free CS can be applied.

6.2 Interrelation of the AOI

As in Section 1.2 as the main goal of this thesis that we wish to determine the relation between AOI-R, AOI-C, AOI-H and AOI-M more accurately for the applications that we study, we summarize the insights we make throughout the presented material below.

A purely theoretic conclusion on the relation of AOI-C, AOI-M and AOI-R is given by Remark 2.4. It specifically tells us how many and what type of measurements (AOI-C) allow us to recover which sparsity levels (AOI-M) with a certain type of algorithm (AOI-R).

In Section 4.1 we study the relation between AOI-C and AOI-H by deriving scaling laws for randomly subsampled Fourier matrices and these matrices structure is motivated by the comparably simple hardware implementation. Moreover, we see that the 3D volumetric model (AOI-M) has a direct influence on two things. First, the reconstruction algorithm (AOI-R) can employ the imposed structure to derive matrix-free and hence efficient means of estimating the sparse vector. Second, the volumetric model allows for more aggressive compression (AOI-C) in frequency domain.

In Section 4.2 we show how existing hardware concepts, namely creating pseudo random sequences from linear feedback shift registers, (AOI-H) can be repurposed to generate pseudo random measurements (AOI-C). Moreover, the creation of these pseudo random sequences as proposed allows significant speedups during the reconstruction (AOI-R), since the matrix-free representation can be adapted to account for the special type of pseudo random sequence.

In Section 5.2.1 we see that correctly modeling AOI-M the beampattern of an antenna array allows to reformulate the problem of DoA estimation (AOI-R) into a line spectral estimation problem with a certain linear structure defined by the aperture of the antenna. Second, in Section 5.2.3 we see that the data model (AOI-M), which consists of a superposition of complex harmonics, allows to reformulate the problem of atomic norm minimization into a semi-definite program, which then can be solved efficiently (AOI-R).

6.3 Outlook

In order to provide some points of attachment for further research we take the results of this thesis and outline some aspects that connect to the material presented here.

The methodology for estimating the sparsity order prior to reconstruction is only valid in the grid-based setting. With the rising interest in the grid-free CS setting, one should also consider this scenario. The method proposed in [58] uses the concept of entropy to infer the sparsity order, which also has an equivalent in the grid-free setting. hence, it might be possible to extend this both geometric and stochastic concept to the grid-free setting.

We have presented a flexible and rather efficient ADMM algorithm for line spectral estimation in Section 5.2.6, but the limiting factor in the form of the projection onto the positive cone of matrices greatly limits the complexity of the models that we can work with – both in terms of dimensionality and resolution. Hence, the need for even more efficient methods remains, since still the proposed algorithm cannot yet be used directly for the settings as presented in Section 4.1 and Section 4.2.

In Section 5.3 we present a routine for the sensing matrix optimization that does improve the matrices behavior, if the matrices' elements are resided in a continuous space. However, well suited compression matrices from a hardware standpoint are usually from a discrete or even finite set. Hence, the search for a good compression matrix gets turned into a combinatorial optimization problem. For these type of problems other solving methods need to be developed, since the gradient iteration is not applicable anymore. One possible route would be to employ methods which are able to train (read: optimize) neural networks that contain discrete values only, as in [191].

For the two proposed applications in Section 4.1 and Section 4.2, it is still unclear how much various hardware imperfections during compression and measurement contaminate the acquired samples. Hence, one should evaluate theoretically, if this issue needs special attention by studying the robustness to these possibly existing imperfections. Then, those should be accounted for either in form of calibration or in form of appropriate modeling for instance of the then possibly colored noise statistics.

As a last aspect, with the rise of deep learning the use of GPUs has become increasingly widespread and accessible in scientific computing and thus also in signal processing. Hence, the proposed software archi-

tecture from Chapter 3 should be extended to treat this heterogeneous computing task more gracefully than we outlined in Section 4.1.

Appendix A

Theorems and Proofs

A.1 Proof of Theorem 2.2

First, we see by linearity of the Fourier transform that the sum $\lambda_1 f_1 + \lambda_2 f_2$ for $\lambda_{1,2} \in \mathbb{C}$ and $f_{1,2} \in S$ also is band-limited with band B. Additionally, we have with the triangle inequality on $\mathbb{C}$ and the linearity of integration that

$$\int_{\mathbb{R}} |\lambda_1 f_1 + \lambda_2 f_2| \, \mathrm{d}t \leqslant |\lambda_1| \int_{\mathbb{R}} |f_1| \, \mathrm{d}t + |\lambda_2| \int_{\mathbb{R}} |f_2| \, \mathrm{d}t < \infty,$$

rendering S a subspace of $L^2(R)$. Further, using Theorem 2.1 together with the periodicity of the signal we can choose a $T_0 < T$ such that there is an integer N satisfying $N \cdot T_0 = P$. Then we see for an $f \in S$ that

$$f(n \cdot T_0 + k \cdot N \cdot T_0) = f(n \cdot T_0)$$

for any $k \in \mathbb{Z}$ and have that S is finite dimensional, since it is isomorphic to the set $\mathbb{C}^N$ by means of the mapping $R : \mathbb{C}^N \to L^2$ defined as

$$\boldsymbol{x} \mapsto (R\boldsymbol{x})(t) = \sum_{n=0}^{N-1} x_{n+1} \operatorname{sinc}\left(\frac{t - nT_0}{T_0}\right),$$

whose inverse R^{-1} is simply defined by the sampling operation with sample distance T_0.

A.2 Proofs for Results in Section 2.5

A.2.1 Proof of Theorem 2.11

Assume 1 holds and consider an arbitrary but fixed 1-sparse $\boldsymbol{x}$ with $\operatorname{supp}(\boldsymbol{x}) = \{q\}$ for some $q \in [N]$. Now 1 implies that $\operatorname{rk} \boldsymbol{B} = 1$. On the other hand, denoting the q-th column of $\boldsymbol{A}_i$ by $\boldsymbol{a}_{i,q}$ yields

$$\boldsymbol{b}_i = \boldsymbol{A}_i \cdot \boldsymbol{x} = \boldsymbol{a}_{i,q} \cdot x_q \quad \text{for} \quad i \in [k].$$

Because all $\boldsymbol{b}_i$ must be linearly dependent, it follows that

$$\boldsymbol{a}_{i,q} = \varphi_{i,q} \cdot \boldsymbol{\psi}_q \quad \text{for} \quad i \in [k],$$

for some non-zero $\boldsymbol{\psi}_q \in \mathbb{C}^\ell$ and some $\varphi_{i,q} \in \mathbb{C}$. In terms of the columns of $\boldsymbol{A}$ this means $\boldsymbol{a}_q = \operatorname{vec}(\boldsymbol{\psi}_q \boldsymbol{\varphi}_q^{\mathrm{T}}) = \boldsymbol{\varphi}_q \otimes \boldsymbol{\psi}_q$ for $\boldsymbol{\varphi}_q = (\varphi_{1,q}, \ldots, \varphi_{k,q})^{\mathrm{T}} \in \mathbb{C}^k$. Since q was arbitrary, conclude that $\boldsymbol{A} = \boldsymbol{\Phi} \diamond \boldsymbol{\Psi}$, where $\boldsymbol{\Phi} = (\boldsymbol{\varphi}_i)_{i \in [N]} \in \mathbb{C}^{k \times N}$ and $\boldsymbol{\Psi} = (\boldsymbol{\psi}_i)_{i \in [N]} \in \mathbb{C}^{\ell \times N}$.

Now consider any s-sparse $\boldsymbol{x}$ for $s \leq r$. Then the structure of $\boldsymbol{A}$ and $\boldsymbol{B}$ yields

$$\boldsymbol{B} = \boldsymbol{\Psi} \operatorname{diag}(\boldsymbol{x}) \boldsymbol{\Phi}^{\mathrm{T}}, \tag{A.1}$$

since $p = \ell$. This can also be restricted to $S = \operatorname{supp}(\boldsymbol{x})$ and then reads as

$$\boldsymbol{B} = \boldsymbol{\Psi}_S \operatorname{diag}(\boldsymbol{x}_S) \boldsymbol{\Phi}_S^{\mathrm{T}}.$$

Now seeking a contradiction to $\operatorname{rk}^* \boldsymbol{\Phi} \geqslant r$, assume that $\operatorname{rk}^* \boldsymbol{\Phi} < r$. This means that there is a set $T \subset [N]$ of size r such that $\boldsymbol{\Phi}_T$ is rank deficient, since otherwise the kruskale rank would be at least r. Now set $\boldsymbol{x} = \sum_{i \in T} \boldsymbol{e}_i$ to construct a ground truth $\boldsymbol{x}$ which is less than s-sparse, which yields $\boldsymbol{B} = \boldsymbol{\Psi}_T \boldsymbol{\Phi}_T^{\mathrm{T}}$. The fact that $\boldsymbol{B}$ has rank strictly smaller than r, because $\boldsymbol{\Phi}_T$ has rank strictly smaller than r, implies the desired contradiction. An analogue argument holds in case of $\operatorname{rk}^* \boldsymbol{\Psi} < r$.

For the opposite direction, assume 2 and again consider (A.1) for some s-sparse $\boldsymbol{x}$ with support S. Because $\boldsymbol{\Phi}$ and $\boldsymbol{\Psi}$ have Kruskal rank at least s, it holds that $\operatorname{rk} [\boldsymbol{\Psi}_S \operatorname{diag}(\boldsymbol{x}_S)] = s$ and $\operatorname{rk} \boldsymbol{\Phi}_S = s$. This implies that $\operatorname{rk} \boldsymbol{B} = s$, because the two matrices are a rank factorization of $\boldsymbol{B}$.

A.2.2 Proof of Theorem 2.12

Given 1, again consider some $\boldsymbol{x} \in \mathbb{C}^N$ with $\operatorname{supp}(\boldsymbol{x}) = \{q\}$ such that

$$\boldsymbol{B} = x_q[\boldsymbol{a}_{1,q}, \dots, \boldsymbol{a}_{k,q}].$$

Since $\boldsymbol{B}$ has rank 1, its columns, which are also the columns of the blocks $\boldsymbol{A}_i$, fulfill

$$\boldsymbol{a}_{i,q} = z_{i,q}\boldsymbol{a}_{i-1,q} \quad \text{for} \quad i \in \{2, \dots, k\}, \tag{A.2}$$

$$\boldsymbol{a}_{i+1,q} = z_{i+1,q}\boldsymbol{a}_{i,q} \quad \text{for} \quad i \in \{1, \dots, k-1\}. \tag{A.3}$$

for appropriate $z_{i,q}$ and still denoting the q-th column of $\boldsymbol{A}_i$ by $\boldsymbol{a}_{i,q}$. Moreover, the overlap yields

$$(\boldsymbol{a}_{i,q})_j = (\boldsymbol{a}_{i-1,q})_{p+j} \quad \text{for} \quad j \in [\ell - p] + 1 \tag{A.4}$$

$$(\boldsymbol{a}_{i+1,q})_j = (\boldsymbol{a}_{i,q})_{p+j} \quad \text{for} \quad j \in [\ell - p] \tag{A.5}$$

and $i \in [k-1]$. Combining (A.2),(A.3) and (A.5) from above yields

$$z_{i+1,q}\left(\boldsymbol{a}_{i,q}\right)_j = z_{i,q}\left(\boldsymbol{a}_{i-1,q}\right)_{p+j} \quad \text{for} \quad j \in [\ell - p] \tag{A.6}$$

and $i \in [k-1]$. From (A.4) it follows that his has to be true for all $j \in [\ell - p]$, so

$$z_{i,q} = z_{i+1,q} = z_q \quad \text{for} \quad i \in [k-1] \tag{A.7}$$

for some z_q independent of i. Summarizing, it holds that

$$\boldsymbol{a}_{i,q} = z_q\boldsymbol{a}_{i-1,q} = z_q^2\boldsymbol{a}_{i-2,q} = \cdots = z_q^{i-1}\boldsymbol{a}_{1,q}.$$

This means that one can only choose z_q and because of the overlap the first p elements of $\boldsymbol{a}_{1,q}$ freely. Now setting $\left(\boldsymbol{a}_{1,q}\right)_{[p]} = \boldsymbol{\psi}_q$ with $\boldsymbol{\psi}_q \in \mathbb{C}^p$ then implies the Khatri-Rao structure $\hat{\boldsymbol{A}} = \boldsymbol{V} \diamond \boldsymbol{\Psi}$, where the matrix $\boldsymbol{A}$ is the restriction of $\hat{\boldsymbol{A}}$ to its first m rows and $\boldsymbol{\Psi} = [\boldsymbol{\psi}_1, \dots, \boldsymbol{\psi}_N]$. The Vandermonde structure of the first factor reads as

$$\boldsymbol{V} = \begin{bmatrix} 1 & \dots & 1 \\ z_1 & \dots & z_N \\ \vdots & \ddots & \vdots \\ z_1^{\lceil m/p \rceil - 1} & \dots & z_N^{\lceil m/p \rceil - 1} \end{bmatrix},$$

if one repeats the reasoning above for any $q \in [N]$.

For the Kruskal ranks of the involved matrices deduce from the Khatri-Rao structure and the overlap that

$$\boldsymbol{B} = \hat{\boldsymbol{\Psi}} \cdot \operatorname{diag}(\boldsymbol{x}) \cdot \boldsymbol{V}_k^{\mathrm{T}},$$

where $\hat{\boldsymbol{\Psi}}$ consists of the first ℓ rows of $\boldsymbol{V}_{\lceil \ell/p \rceil} \diamond \boldsymbol{\Psi}$ and $\boldsymbol{V}_{\lceil \ell/p \rceil}$ and $\boldsymbol{V}_k$ are the restrictions of $\boldsymbol{V}$ to its first $\lceil \ell/p \rceil$ and k rows respectively. Now consider the above equation for any s-sparse $\boldsymbol{x}$ with support set S. As in the last proof, one can argue that the factors involved in the Khatri-Rao product must fulfill the Kruskal rank conditions imposed on them in the statement of 2 for $\boldsymbol{B}$ to have rank s, since again the product of the restricted matrices $\hat{\boldsymbol{\Psi}}_S$ and $\left(V_k^{\mathrm{T}}\right)_S$ is a rank factorization of $\boldsymbol{B}$.

Now given 2, proceed similarly to the proof of Theorem 2.11, but this time, this yields for an s-sparse $\boldsymbol{x}$ with $s \leqslant r$ that

$$\boldsymbol{B} = \hat{\boldsymbol{\Psi}} \cdot \operatorname{diag}(\boldsymbol{x}) \cdot \boldsymbol{V}_k^{\mathrm{T}},$$

where $\boldsymbol{V}_{\lceil \ell/p \rceil}$ and $\boldsymbol{V}_k$ are the restrictions of $\boldsymbol{V}$ to its first $\lceil \ell/p \rceil$ and k rows respectively. Because of the conditions on $\hat{\boldsymbol{\Psi}}$ and $\boldsymbol{V}$ deduce $\operatorname{rk} \boldsymbol{B} = s$.

A.2.3 Vandermonde Matrices

Further on the following lemma will be used a couple of times, which calculates the value of the discontinued geometric series.

Lemma A.1. *For any $q \in \mathbb{C} \setminus \{1\}$ it holds that*

$$\sum_{k=1}^{n} q^k = \frac{q - q^{n+1}}{1 - q} = S_n(q) \tag{A.8}$$

In the case $q = 1$ the above formula yields

$$\lim_{q \to 1} S_n(q) = \lim_{q \to 1} \frac{q - q^{n+1}}{1 - q} = \lim_{q \to 1} \frac{1 - (n+1)q^n}{-1} = n. \tag{A.9}$$

■

Proof. This follows from a straightforward application of L'Hôpital's rule. ■

The following theorem contains all information needed to derive an algorithm for Vandermonde matrices with low coherence and bounds thereof. Here the general case for two arbitrary Vandermonde columns but also the special case where the first row of the matrix is a subset of

the complex unit circle are considered. Furthermore it defines upper and lower envelope functions (see Figure A.1) for the inner product of two Vandermonde columns depending only on the phase shift between two generating elements for fixed amplitudes.

Theorem A.1. *Consider two vectors*

$$z_1 = (c_1 e^{i\phi_1}, \ldots, c_1^n e^{in\phi_1})^{\mathrm{T}}$$

and

$$z_2 = (c_2 e^{i\phi_2}, \ldots, c_2^n e^{in\phi_2})^{\mathrm{T}},$$

where c_1, c_2 are positive real numbers and $\phi_1 \leqslant \phi_2$ with $\phi_1, \phi_2 \in [0, 2\pi]$ and define the function

$$\lambda(c_1, c_2, \phi) = \frac{|\langle z_1, z_2 \rangle|^2}{\|z_1\|^2 \|z_2\|^2} \tag{A.10}$$

with $\phi = \phi_2 - \phi_1$, then the following statements hold.

1. $$\lambda(c_1, c_2, \phi) = \frac{\dfrac{1 + c_1^{2n} c_2^{2n} - 2c_1^n c_2^n \cos[n\phi]}{(1 - c_1^{2n})(1 - c_2^{2n})}}{\dfrac{1 + c_1^2 c_2^2 - 2c_1 c_2 \cos\phi}{(1 - c_1^2)(1 - c_2^2)}}.$$

2. $$\lambda(1, 1, \phi) = \frac{\sin^2\left(\frac{1}{2} n\phi\right)}{n^2 \sin^2\left(\frac{1}{2}\phi\right)}.$$

3. *Observe symmetry along the unit circle in $\mathbb{C}$. In other words*

$$\lambda(c_1, c_2, \phi) = \lambda(c_1^{-1}, c_2^{-1}, \phi) = \lambda(c_1, c_2, -\phi).$$

4. *For $c_1 \neq 1$ and $c_2 \neq 1$ there is a bound for λ via*

$$\kappa(c_1, c_2, \phi) \leqslant \lambda(c_1, c_2, \phi) \leqslant \eta(c_1, c_2, \phi),$$

where κ and η are defined as

$$\kappa(c_1, c_2, \phi) = \frac{\dfrac{(1 - c_1^n c_2^n)^2}{(1 - c_1^{2n})(1 - c_2^{2n})}}{\dfrac{1 + c_1^2 c_2^2 - 2c_1 c_2 \cos\phi}{(1 - c_1^2)(1 - c_2^2)}},$$

and

$$\eta(c_1, c_2, \phi) = \begin{cases} \dfrac{1}{n^2 \sin^2\left(\frac{1}{2}\phi\right)} & \text{for} \quad c_1 = c_2 = 1 \\ \dfrac{(1 - c_1^n c_2^n)^2}{(1 - c_1^{2n})(1 - c_2^{2n})} \\ \dfrac{1 + c_1^2 c_2^2 - 2c_1 c_2 \cos\phi}{(1 - c_1^2)(1 - c_2^2)} & \text{otherwise.} \end{cases}$$

Moreover, the bounds satisfy for $c_1, c_2 \in (0, 1]$

$$\begin{aligned} \lambda(c_1, c_2, \phi) &= \kappa(c_1, c_2, \phi) \\ &\quad \text{if and only if} \quad \phi = k \cdot (2\pi)/n \\ &\quad \text{for some} \quad k \in \{0, \ldots, n\}, \\ \lambda(c_1, c_2, \phi) &= \eta(c_1, c_2, \phi) \\ &\quad \text{if and only if} \quad \phi = \pi/n + k \cdot (2\pi)/n \\ &\quad \text{for some} \quad k \in \{1, \ldots, n-1\}, \\ \kappa(c_1, c_2, \phi) &= \kappa(1/c_1, 1/c_2, \phi) \\ &\quad \text{for} \quad \phi \in [0, 2\pi], \\ \eta(c_1, c_2, \phi) &= \eta(1/c_1, 1/c_2, \phi) \\ &\quad \text{for} \quad \phi \in (0, 2\pi). \end{aligned}$$

5. *For any* $c > 0$ *and* $c_1 > 0$ *it holds that*

$$\lim_{c_1 \to \infty} \lambda(c/c_1, c_1, \phi) = 0 \quad \text{for} \quad \phi \in [0, 2\pi].$$

6. *Roots of* $\lambda(c_1, c_2, \cdot)$ *exist if and only if* $c_1 c_2 = 1$ *and these roots are* $\phi = k \cdot (2\pi/n)$ *for* $k \in \{1, \ldots, n-1\}$*, i.e. there are exactly* $n - 1$ *roots.*

■

Proof.

1. Calculate the inner product of z_1 and z_2 with the law of cosines and the formula in (A.8), which yields

$$|\langle z_1, z_2 \rangle|^2 = \left| \sum_{k=1}^{n} c_1^k e^{ik\phi_1} \cdot c_2^k e^{-ik\phi_2} \right|^2$$

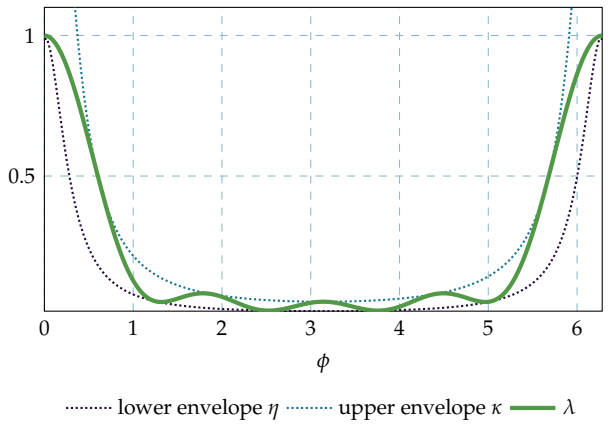


Figure A.1 *Envelopes for the correlation function of columns of Vandermonde matrices can be calculated in closed form.* – The envelope curves for $\lambda(c_1, c_2, \cdot)$ for the case $c_1 = c_2^{-1} = 1.15$ and $n = 5$. ■

$$
\begin{aligned}
&= \left| \sum_{k=1}^{n} (c_1 c_2)^k e^{ik\phi} \right|^2 \\
&= \left| \frac{c_1 c_2 e^{i\phi} - (c_1 c_2)^{n+1} e^{i\phi(n+1)}}{1 - c_1 c_2 e^{i\phi}} \right|^2 \\
&= c_1^2 c_2^2 \frac{1 + c_1^{2n} c_2^{2n} - 2 c_1^n c_2^n \cos(n\phi)}{1 + c_1^2 c_2^2 - 2 c_1 c_2 \cos\phi}
\end{aligned}
$$

In this case consider L'Hôpital's rule for $c_1 c_2 = 1$ and $\phi = 0$, which yields $c_1 c_2 e^{i\phi} = 1$ and thus $|\langle z_1, z_2 \rangle|^2 = n^2$. To derive an expression for the coherence of a matrix, divide by the norm of the vectors z_1 and z_2. Now (A.8) yields

$$
\begin{aligned}
\lambda(c_1, c_2, \phi) &= \frac{|\langle z_1, z_2 \rangle|^2}{\|z_1\|^2 \|z_2\|^2} \\
&= \frac{\dfrac{1 + c_1^{2n} c_2^{2n} - 2 c_1^n c_2^n \cos[n\phi]}{(1 - c_1^{2n})(1 - c_2^{2n})}}{\dfrac{1 + c_1^2 c_2^2 - 2 c_1 c_2 \cos\phi}{(1 - c_1^2)(1 - c_2^2)}},
\end{aligned}
$$

where one has to consider L'Hôpital's rule for $c_1 = 1$ or $c_2 = 1$ as well. Then $\|z_1\|^2 = n$ or $\|z_2\|^2 = n$ respectively.

2. Using 1 and taking the appropriate limits yields

$$\lambda(1,1,\phi) = \frac{1-\cos(n\phi)}{n^2(1-\cos\phi)} = \frac{\sin^2\left(\frac{1}{2}n\phi\right)}{n^2\sin^2\left(\frac{1}{2}\phi\right)}, \tag{A.11}$$

3. Simply manipulate $\lambda(c_1^{-1}, c_2^{-1}, \phi)$ by factoring out the proper powers of c_1 and c_2 and derive $\lambda(c_1, c_2, \phi)$.

4. The statement follows from the fact that $-1 \leqslant \cos x \leqslant 1$ for all $x \in \mathbb{R}$ and the locations where $|\cos x| = 1$.

5. Assume that $c_1 \cdot c_2 = c < 1$ and without loss of generality suppose that $c_1 \leqslant c_2$. Then hold c fixed and rewrite λ as

$$\begin{aligned}
&\lambda(c_1, c_2, \phi) = \lambda(c/c_2, c_2, \phi) = \\
&= \frac{(\cos(n\phi) - c^n)^2 + \sin^2(n\phi)}{(\cos\phi - c)^2 + \sin^2\phi} \frac{(1-(c/c_2)^2)(1-c_2^2)}{(1-(c/c_2)^{2n})(1-c_2^{2n})} = \\
&= \frac{(\cos(n\phi) - c^n)^2 + \sin^2(n\phi)}{(\cos\phi - c)^2 + \sin^2\phi} \frac{1 + c^2\left(1 - \frac{1}{c_2^2}\right) - c_2^2}{1 + c^{2n}\left(1 - \frac{1}{c_2^{2n}}\right) - c_2^{2n}}.
\end{aligned}$$

Now take the limit for growing c_2 and fixed ϕ, which yields

$$\lim_{c_2 \to \infty} \lambda(c/c_2, c_2, \phi) = 0 \tag{A.12}$$
$$\text{for all} \quad \phi \in [0, 2\pi] \quad \text{and} \quad c < 1.$$

6. First consider the case $c_1 \cdot c_2 < 1$ and rewrite λ in another form with the trigonometric identity $\sin^2 x + \cos^2 x = 1$ and obtain

$$\lambda(c_1, c_2, \phi) = \frac{\dfrac{(\cos(n\phi) - (c_1c_2)^n)^2 + \sin^2(n\phi)}{(1-c_1^2)(1-c_2^2)}}{\dfrac{(\cos\phi - c_1c_2)^2 + \sin^2\phi}{(1-c_1^{2n})(1-c_2^{2n})}}. \tag{A.13}$$

Equation (A.13) shows that λ is zero if and only if

$$\sin(n\phi) = \cos(n\phi) - (c_1c_2)^n = 0.$$

Let us assume that $\sin(n\phi) = 0$, which implies that $\phi = k \cdot (\pi/n)$. Now consider that $|\cos(k\pi)| = 1$ and the only case that $\cos(n\phi) - (c_1c_2)^n = 0$ is when $c_1c_2 = 1$ what was excluded explicitly. In other words for c_1c_2 different than 1, there are no two vectors z_1 and z_2 which are orthogonal to each other.

■

Some results from above are visualized in Figure A.1.

A.2.3.1 Proof of Theorem 2.13

From Theorem A.1 notice know that $\lambda(c_1, c_2, \phi) \geqslant \kappa(c_1, c_2, \phi)$ for all c_1, c_2 and ϕ. Algorithm 2.3 produces a matrix $\boldsymbol{V}$ so that the minimum angle between two elements in its first row is $2\pi/m$. Moreover those elements are located either on the circle with radius c or the circle with radius $1/c$. If two elements share the same circle, their minimum angle between each other is $4\pi/m$. If not they enclose the angle $2\pi/m$. This yields the estimate

$$\begin{aligned}\mu(\boldsymbol{V})^2 \geqslant \max\{&\kappa(1/c, 1/c, 4\pi/m), \kappa(c, c, 4\pi/m),\\ &\kappa(c, 1/c, 2\pi/m), \kappa(1/c, c, 2\pi/m)\},\end{aligned} \tag{A.14}$$

which, together with the symmetry of κ as stated in fact 4 in Theorem A.1 and that λ is the square of the inner product of two vectors with Vandermonde structure, becomes what was asserted.

A.2.3.2 Proof of Theorem 2.14

Let $\boldsymbol{V}$ be the output of Algorithm 2.3 for given c, n and m, which implies the first row of $\boldsymbol{V}$ is placed on two circles with radii c and $1/c$. Then the angle between two elements in the first row of $\boldsymbol{V}$ which are not on the same circle is an integer multiple of $2\pi/m$. Elements on the same circle enclose an angle which is an integer multiple of $4\pi/m$.
Case 1: Suppose that $m < 2n$. From the monotony of η and its symmetry

$$\eta(c_1, c_2, \pi - \phi) = \eta(c_1, c_2, \pi + \phi) \quad \text{for} \quad \phi \in (0, \pi)$$

obtain

$$\eta(c, c, 4\pi/m) \geqslant \eta(c, c, k \cdot 4\pi/m) \geqslant \lambda(c, c, k \cdot 4\pi/m)$$

for $k \in \{1, \ldots, \lfloor m/2 \rfloor\}$ and

$$\eta(1/c, c, 2\pi/m) \geqslant \eta(1/c, c, k2\pi/m)$$

for $k \in \{1, \ldots, m\}$. Since the above estimates address all possible values λ can take for $\boldsymbol{V}$ and the maximum over all values λ attains the square of the coherence of $\boldsymbol{V}$ the proof for $m < 2n$ is finished.

Case 2: Suppose now that $4n \geqslant m \geqslant 2n$. From the monotony of η it follows that

$$\begin{aligned}\lambda(1/c, c, 2\pi/m) &> \eta(1/c, c, \pi/n) \geqslant \eta(1/c, c, k \cdot \pi/n) \\ &\geqslant \lambda(1/c, c, k \cdot 2\pi/m) \\ &\quad \text{for} \quad k \in \{2, \ldots, m\}. \end{aligned} \tag{A.15}$$

Moreover, $4n > m \geqslant 2n$ yields

$$\frac{\pi}{n} < \frac{4\pi}{m} \leqslant \frac{2\pi}{n}$$

and estimate

$$\begin{aligned}\eta(c, c, 4\pi/m) &\geqslant \eta(c, c, k \cdot 4\pi/m) \geqslant \lambda(c, c, k \cdot 4\pi/m) \\ &\quad \text{for} \quad k \in \{1, \ldots, \lfloor m/2 \rfloor\}. \end{aligned} \tag{A.16}$$

These estimates address all possible values for λ and so the statement follows in this case.
Case 3: Suppose now that $m > 4n$. From the monotony of λ on $[0, 2\pi/n]$ and the fact that $4\pi/m$ and $2\pi/m$ are left of the first touching point of λ and η, namely π/n, estimate

$$\begin{aligned}\lambda(c, c, 4\pi/m) &\geqslant \lambda(c, c, k_1 \cdot 4\pi/m) \geqslant \lambda(c, c, \pi/n) \\ &= \eta(c, c, \pi/n) \geqslant \eta(c, c, k_2 \cdot 4\pi/m) \\ &\geqslant \lambda(c, c, k_2 \cdot 4\pi/m), \end{aligned} \tag{A.17}$$

for $k_1 \in \{1, \ldots, \lfloor m/(4n) \rfloor\}$ and $k_2 \in \{\lfloor m/(4n) \rfloor + 1, \ldots, \lfloor m/2 \rfloor\}$. With a similar argument estimate

$$\begin{aligned}\lambda(1/c, c, 2\pi/m) &\geqslant \lambda(1/c, c, k_3 \cdot 2\pi/m) \geqslant \lambda(1/c, c, \pi/n) \\ &= \eta(1/c, c, \pi/n) \geqslant \eta(1/c, c, k_4 \cdot 2\pi/m) \\ &\geqslant \lambda(1/c, c, k_4 \cdot 4\pi/m), \end{aligned}$$

for $k_1 \in \{1, \ldots, \lfloor m/(2n) \rfloor\}$ and $k_2 \in \{\lfloor m/(2n) \rfloor + 1, \ldots, m\}$. This again estimates all values λ ever takes and it concludes the proof.

A.3 Proof of Lemma 5.1

First, we use the facts $\|\boldsymbol{M}\|_F^2 = \mathrm{tr}(\boldsymbol{M}^{\mathrm{H}}\boldsymbol{M})$ and $\mathrm{tr}(\boldsymbol{X}\boldsymbol{Y}) = \mathrm{tr}(\boldsymbol{Y}\boldsymbol{X})$ to get

$$\begin{aligned}\eta(\boldsymbol{\Phi}, \boldsymbol{A}_i) &= \mathrm{tr}\left(\boldsymbol{A}_i^{\mathrm{H}}\boldsymbol{\Phi}^{\mathrm{H}}\boldsymbol{\Phi}\boldsymbol{A}_i\boldsymbol{A}_i^{\mathrm{H}}\boldsymbol{\Phi}^{\mathrm{H}}\boldsymbol{\Phi}\boldsymbol{A}_i\right) \\ &\quad - 2\,\mathrm{tr}\left(\boldsymbol{A}_i^{\mathrm{H}}\boldsymbol{A}_i\boldsymbol{A}_i^{\mathrm{H}}\boldsymbol{\Phi}^{\mathrm{H}}\boldsymbol{\Phi}\boldsymbol{A}_i\right) + \mathrm{tr}\left(\boldsymbol{A}_i^{\mathrm{H}}\boldsymbol{A}_i\boldsymbol{A}_i^{\mathrm{H}}\boldsymbol{A}\right) \\ &= \mathrm{tr}\left(\boldsymbol{A}_i^{\mathrm{H}}\boldsymbol{\Phi}^{\mathrm{H}}\boldsymbol{\Phi}\boldsymbol{A}_i\boldsymbol{A}_i^{\mathrm{H}}\boldsymbol{\Phi}^{\mathrm{H}}\boldsymbol{\Phi}\boldsymbol{A}_i\right) \\ &\quad - 2\,\mathrm{tr}\left(\boldsymbol{\Phi}\boldsymbol{A}_i\boldsymbol{A}_i^{\mathrm{H}}\boldsymbol{A}_i\boldsymbol{A}_i^{\mathrm{H}}\boldsymbol{\Phi}^{\mathrm{H}}\right) + \mathrm{tr}\left(\boldsymbol{A}_i^{\mathrm{H}}\boldsymbol{A}_i\boldsymbol{A}_i^{\mathrm{H}}\boldsymbol{A}\right).\end{aligned}$$

Now with two well known results from matrix calculus

$$\begin{aligned}&\nabla_{\boldsymbol{X}}\,\mathrm{tr}\left(\boldsymbol{M}^{\mathrm{H}}\boldsymbol{X}^{\mathrm{H}}\boldsymbol{X}\boldsymbol{M}\boldsymbol{M}^{\mathrm{H}}\boldsymbol{X}^{\mathrm{H}}\boldsymbol{X}\boldsymbol{M}\right) = \\ &\quad 4\boldsymbol{X}\boldsymbol{M}\boldsymbol{M}^{\mathrm{H}}\boldsymbol{X}^{\mathrm{H}}\boldsymbol{X}\boldsymbol{M}\boldsymbol{M}^{\mathrm{H}}\end{aligned}$$

and $\nabla_{\boldsymbol{X}}\,\mathrm{tr}\left(\boldsymbol{X}\boldsymbol{M}\boldsymbol{X}^{\mathrm{H}}\right) = 2\boldsymbol{X}\boldsymbol{M}$, we conclude the statement.

Appendix B

Definitions and Supplemental Material

B.1 Optimizing the Zero-Padding

To alleviate the drawback that the performance of the FFT depends on the prime factors of the processed dimension, we combine zero-padding with the FFT algorithm to increase the dimension n to a more feasible dimension in terms of its prime factors. To this end we define the mapping $\varphi : \mathbb{N} \to \mathbb{N}$, where $\varphi(n)$ is the number of multiply-and-accumulate (MAC) operations necessary to calculate $\boldsymbol{\mathcal{F}}_n \boldsymbol{x}$ for $\boldsymbol{x} \in \mathbb{C}^n$. Moreover, we define $\Phi : \mathbb{N} \to \mathbb{N}$ as $\Phi(n) = \min \operatorname{argmin}_{m \geqslant 2n-1} \varphi(m)$ and $\gamma_k : \mathbb{C}^n \to \mathbb{C}^{2n-1+k}$ as

$$\gamma_k(\boldsymbol{c}) = [c_1, \ldots, c_n, \boldsymbol{0}_k^{\mathrm{T}}, c_2, \ldots, c_n]^{\mathrm{T}}.$$

The key idea is now to embed $\boldsymbol{\Gamma}(\boldsymbol{c})$ into a larger circulant matrix $\boldsymbol{\mathcal{C}}_c(\boldsymbol{c}) \in \mathbb{C}^{N \times N}$, such that the possible bottleneck when calculating $\boldsymbol{\mathcal{F}}_n$ via an FFT is replaced by the shorter execution time of the FFT for $\boldsymbol{\mathcal{F}}_k$ for some $k \geqslant 2n-1$.

Following the formulation of the Cooley-Tukey algorithm for computing the FFT as found in [192], an algebraic matrix representation can be derived, which describes $\boldsymbol{\mathcal{F}}_n$ as composition of $\boldsymbol{\mathcal{F}}_\nu$, with $\nu \in \mathbb{N}$ being a prime factor of n, and $\boldsymbol{\mathcal{F}}_p$ with $p \in \mathbb{N} = n/\nu$ being the order of the square remainder matrix

$$\boldsymbol{\mathcal{F}}_n = \boldsymbol{P}_n^{(\nu)} \cdot (\boldsymbol{I}_\nu \otimes \boldsymbol{\mathcal{F}}_p) \cdot \boldsymbol{D}_n^{(\nu)} \cdot (\boldsymbol{\mathcal{F}}_\nu \otimes \boldsymbol{I}_p). \tag{B.1}$$

where $\boldsymbol{D}_n^{(\nu)}$ is a diagonal matrix of size n holding the FFT twiddle factors corresponding to ν and $\boldsymbol{P}_n^{(\nu)}$ being the stride permutation matrix which can be defined according to its effect on a vector via

$$\boldsymbol{P}_n^{(\nu)} \cdot [x_0, x_1, \ldots, x_{n-1}]^{\mathrm{T}} = [y_0, y_1, \ldots, y_{\nu-1}]^{\mathrm{T}},$$

where $y_i = [x_i, x_{i+p}, x_{i+2p}, \ldots, x_{i+(\nu-1)p}]$. Analyzing (B.1) we find a product of four square matrices of size $n \times n$ each. $\boldsymbol{P}_n^{(\nu)}$ can be implemented through clever memory access, requiring no additional ops, whereas $\boldsymbol{D}_n^{(\nu)}$ requires n MAC operations. The anterior Kronecker product resembles a block diagonal matrix with ν copies of the remainder matrix requiring $\nu \cdot \varphi(p)$ MAC ops and, given $\nu = n$ no operations at all, as it then equals to the identity matrix. The posterior Kronecker product is a structured sparse matrix, requiring $p \cdot \nu^2 = \nu \cdot n$ MAC ops. In total, (B.1) requires $\nu \cdot \varphi(p) + (\nu + 1)n$ MAC ops. Defining $\psi_n \in \mathbb{N}^w$ as the prime factor decomposition of n with w being the prime factor count, we recursively apply (B.1) over all $\psi_{n,i}$ resulting in a total complexity of

$$\varphi(n) = n \cdot (1 + w + \sum_{i=1}^{w} \psi_{n,i}). \tag{B.2}$$

The special case $\nu = 4$ needs to be considered carefully during the implementation of the aforementioned optimization problem. As all elements in $\boldsymbol{\mathcal{F}}_4$ are drawn from the set $(1, -1, i, -i)$ only simple additions are required in order to compute its action on a vector. Many current computing platforms are capable of exploiting this, resulting in an additional gain over the estimate $\varphi(4)$. The implementation used in this work employs a variant of Dijkstra's algorithm [193] for the search and rewards prime factors of four on grounds of the preceding consideration. To incorporate above finding about ϕ in our procedure, we define

$$\boldsymbol{\mathcal{C}}_n(\boldsymbol{c}) = \begin{cases} \boldsymbol{\Gamma}(\boldsymbol{c}), & \text{for} \quad \varphi(\Phi(n)) \geqslant \varphi(n) \\ \boldsymbol{\Gamma}(\gamma_{\Phi(n)-2n-1}(\boldsymbol{c})) & \text{otherwise.} \end{cases}$$

The case $\varphi(\Phi(n)) \geqslant \varphi(n)$ represents the fact that we cannot improve the execution time needed to apply $\boldsymbol{\mathcal{F}}_n$ by expanding the circulant matrix. To finalize the procedure note that for $k \in \mathbb{N}$ and $\boldsymbol{x} \in \mathbb{C}^n$ it holds that

$$\begin{bmatrix} \boldsymbol{y}_1 \\ \boldsymbol{z} \end{bmatrix} = \boldsymbol{\Gamma}(\gamma_k(\boldsymbol{c})) \cdot \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{0}_{n-1+k} \end{bmatrix}, \quad \text{where} \quad \boldsymbol{y}_1 = \boldsymbol{\Gamma}(\boldsymbol{c}) \cdot \boldsymbol{x}$$

with $\boldsymbol{y}_1 \in \mathbb{C}^n$ and $\boldsymbol{z} \in \mathbb{C}^{n-1+k}$. As such, the application of $\boldsymbol{\Gamma}(\gamma_k(\boldsymbol{c}))$ to a vector can be carried out efficiently according to (3.8).

B.2 Wirtinger Calculus

To calculate the partial derivatives of functions which depend on complex variables and map to $\mathbb{R}$ we make use of the so called Wirtinger calculus. To this end, let f be a function $f : \mathbb{C}^n \mapsto \mathbb{R}$, then its Wirtinger derivative is defined as

$$\frac{\partial f}{\partial \boldsymbol{x}} = \frac{1}{2}\left(\frac{\partial f}{\partial \boldsymbol{y}} - \jmath\frac{\partial f}{\partial \boldsymbol{z}}\right) \quad \text{and} \quad \frac{\partial f}{\partial \overline{\boldsymbol{x}}} = \frac{1}{2}\left(\frac{\partial f}{\partial \boldsymbol{y}} + \jmath\frac{\partial f}{\partial \boldsymbol{z}}\right),$$

where $\boldsymbol{y} = \operatorname{Re}\{\boldsymbol{x}\}$ and $\boldsymbol{z} = \operatorname{Im}\{\boldsymbol{x}\}$. Note that for $f(x) = \overline{x} = y - \jmath z$ we obtain

$$\frac{\partial \overline{x}}{\partial x} = \frac{1}{2}\left(\frac{\partial y - \jmath z}{\partial y} - \jmath\frac{\partial y - \jmath z}{\partial z}\right) = \frac{1}{2}\left(1 - \jmath(-\jmath)\right) = 0. \tag{B.3}$$

For this reason, in Wirtinger derivatives we can more or less treat x and $\overline{x}$ as independent variables. Also, it is straightforward to derive the following three simple rules

$$\frac{\partial \langle \boldsymbol{b}, \boldsymbol{a}\rangle}{\partial \overline{\boldsymbol{b}}} = \frac{\partial \boldsymbol{b}^{\mathrm{H}}\boldsymbol{a}}{\partial \overline{\boldsymbol{b}}} = \boldsymbol{a}, \quad \frac{\partial \langle \boldsymbol{a}, \boldsymbol{b}\rangle}{\partial \overline{\boldsymbol{b}}} = \frac{\partial \boldsymbol{a}^{\mathrm{H}}\boldsymbol{b}}{\partial \overline{\boldsymbol{b}}} = \boldsymbol{0} \tag{B.4}$$

and

$$\frac{\partial \|\boldsymbol{A}\boldsymbol{b} - \boldsymbol{c}\|^2}{\partial \overline{\boldsymbol{b}}} = \boldsymbol{A}^{\mathrm{H}}(\boldsymbol{A}\boldsymbol{b} - \boldsymbol{c}). \tag{B.5}$$

All three of them can easily be extended to the case where $\boldsymbol{a}$ and $\boldsymbol{b}$ are matrices, since both the inner product of matrices we use here and the Frobenius norm induced by it treat matrices as if they were vectors realigned into matrices.

Appendix C

Derivations

C.1 Derivation of the CRB in Section 4.1.5

Let $\hat{\boldsymbol{H}}(\boldsymbol{\omega}) = \operatorname{diag}\{\mathrm{e}^{-\jmath\boldsymbol{\omega}\tau_{x,y}(x_d,y_d,z_d))}\} \odot \boldsymbol{H}(\boldsymbol{\omega})$. With this definition at hand, the elements of the FIM can be computed by the blocks of $\boldsymbol{\Phi}\frac{\partial \boldsymbol{b}}{\partial \boldsymbol{u}^{\mathrm{T}}}$ given by

$$\boldsymbol{S}_i\boldsymbol{\mathcal{F}}\boldsymbol{\Sigma}_i\boldsymbol{\mathcal{F}}^{\mathrm{H}}\frac{\partial \boldsymbol{b}_i}{\partial \boldsymbol{p}} = a_1\boldsymbol{S}_i\boldsymbol{\mathcal{F}}\boldsymbol{\Sigma}_i\boldsymbol{\mathcal{F}}^{\mathrm{H}}\left(\hat{\boldsymbol{H}}(\boldsymbol{\omega})\odot \left(\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{p}} - \jmath\boldsymbol{\omega}\frac{\partial \tau_i(x_1,y_1,z_1)}{\partial \boldsymbol{p}}\right)\right) \tag{C.1}$$

$$\boldsymbol{S}_i\boldsymbol{\mathcal{F}}\boldsymbol{\Sigma}_i\boldsymbol{\mathcal{F}}^{\mathrm{H}}\frac{\partial \boldsymbol{b}_i}{\partial a} = \boldsymbol{S}_i\boldsymbol{\mathcal{F}}\boldsymbol{\Sigma}_i\boldsymbol{\mathcal{F}}^{\mathrm{H}}\frac{\boldsymbol{b}_i}{a} \tag{C.2}$$

$$\boldsymbol{S}_i\boldsymbol{\mathcal{F}}\boldsymbol{\Sigma}_i\boldsymbol{\mathcal{F}}^{\mathrm{H}}\frac{\partial \boldsymbol{b}_i}{\partial \varphi} = \jmath\boldsymbol{S}_i\boldsymbol{\mathcal{F}}\boldsymbol{\Sigma}_i\boldsymbol{\mathcal{F}}^{\mathrm{H}}\boldsymbol{b}_i \tag{C.3}$$

with

$$\begin{aligned}
\frac{\partial \tau_i(x_1,y_1,z_1)}{\partial x_1} &= \frac{c_0}{2}\frac{(x_1-x_i)}{\sqrt{(x_1-x_i)^2+(y_1-y_i)^2+z_1^2}}\\
\frac{\partial \tau_i(x_1,y_1,z_1)}{\partial z_1} &= \frac{c_0}{2}\frac{z_1}{\sqrt{(x_1-x_i)^2+(y_1-y_i)^2+z_1^2}}\\
\frac{\partial g_i x_1,y_1,z_1}{\partial x_1} &= -2\frac{x_1-x_i}{(z_1\tan(\theta))^2}\cdot g_i(x_1,y_1,z_1)\\
\frac{\partial g_i x_1,y_1,z_1}{\partial z_1} &= -\frac{(x_1-x_i)^2-(y_1-y_i)^2}{z_1^3\tan(\theta)^2}\cdot g_i(x_1,y_1,z_1).
\end{aligned}$$

The derivatives with respect to y_1 follow from the derivatives with respect to x_1 in a similar fashion as above. The FIM can be formulated as a block matrix by separating the nuisance parameters, leading to

$$\boldsymbol{J} = \frac{2}{\sigma_n^2}\begin{bmatrix} \boldsymbol{J}_{\boldsymbol{p}} & \boldsymbol{J}_{\boldsymbol{p},(a,\varphi)} \\ \boldsymbol{J}_{\boldsymbol{p},(a,\varphi)}^{\mathrm{T}} & \boldsymbol{J}_{a,\varphi} \end{bmatrix}. \tag{C.4}$$

The CRB for the defect location is then given by

$$\boldsymbol{c} = [C_x, C_y, C_z]^{\mathrm{T}} = \operatorname{diag}\{\boldsymbol{C}_{\boldsymbol{p}}\}, \tag{C.5}$$

where $\boldsymbol{C}_{\boldsymbol{p}}$ is the upper left 3×3 block of $\boldsymbol{J}^{-1}$. Using the blocks defined in Equation (C.4), $\boldsymbol{C}_{\boldsymbol{p}}$ can be straightforwardly computed using the Schur complement as

$$\boldsymbol{Q} = \boldsymbol{J}_{a,\varphi} - \boldsymbol{J}_{\boldsymbol{p},(a,\varphi)}^{\mathrm{T}} \boldsymbol{J}_{p}^{-1} \boldsymbol{J}_{\boldsymbol{p},(a,\varphi)}$$

$$\boldsymbol{C}_{\boldsymbol{p}} = \frac{\sigma_n^2}{2}\left(\boldsymbol{J}_{\boldsymbol{p}}^{-1} + \boldsymbol{J}_{\boldsymbol{p}}^{-1} \boldsymbol{J}_{p,(a,\varphi)} \boldsymbol{Q}^{-1} \boldsymbol{J}_{\boldsymbol{p},(a,\varphi)}^{\mathrm{T}} \boldsymbol{J}_{\boldsymbol{p}}^{-1}\right).$$

C.2 Derivation of the ADMM in Section 5.2.6

Taking a close look at $\mathcal{L}$ in Equation (5.14) we see that the only derivatives which are not straightforward to calculate are those with respect to $\boldsymbol{u}$, since it is the defining tensor of the multilevel Toeplitz structure. In this case the expressions of interest are

$$\frac{\partial}{\partial \boldsymbol{u}} \langle \boldsymbol{A}, \boldsymbol{T}_{(\boldsymbol{n},d)}^{\mathbf{H}}(\boldsymbol{u}) \rangle \quad \text{and} \quad \frac{\partial}{\partial u_{\boldsymbol{e}}} \langle \boldsymbol{T}_{(\boldsymbol{n},d)}^{\mathbf{H}}(\boldsymbol{u}), \boldsymbol{T}_{(\boldsymbol{n},d)}^{\mathbf{H}}(\boldsymbol{u}) \rangle$$

for a given Hermitian matrix $\boldsymbol{A}$. To this end, for given $n \in \mathbb{N}$ and $p \in [n] \cup -[n] \cup \{0\}$, we define $\boldsymbol{S}_n^p$ via

$$\left[\boldsymbol{S}_n^p\right]_{(k,\ell)} = 1 \quad \text{for} \quad \ell - k = p - n.$$

Note that $\boldsymbol{S}_n^p$ has 1 only on a shifted diagonal and $\boldsymbol{S}_n^0 = \boldsymbol{I}_n$. Now, we can rewrite

$$\boldsymbol{T}_{(\boldsymbol{n},d)}^{\mathbf{H}}(\boldsymbol{u}) = \boldsymbol{T}_{\text{upper}} + \boldsymbol{T}_{\text{lower}} \tag{C.6}$$

where $\boldsymbol{T}_{\text{upper}}$ and $\boldsymbol{T}_{\text{lower}}$ are constructed by explicitly unraveling the recursive definition of $\boldsymbol{T}_{(\boldsymbol{n},d)}^{\mathbf{H}}(\boldsymbol{u})$ while keeping the Hermitian symmetry in mind. So they are defined as

$$\boldsymbol{T}_{\text{upper}} = \sum_{\boldsymbol{i} \in \mathcal{N}} \left(\boldsymbol{S}_{N_1}^{i_1 - 1} \otimes \boldsymbol{S}_{N_2}^{i_2 - N_2} \otimes \cdots \otimes \boldsymbol{S}_{N_d}^{i_d - N_d}\right) u_{\boldsymbol{i}} \tag{C.7}$$

$$\boldsymbol{T}_{\text{lower}} = \boldsymbol{T}^{\text{H}}_{\text{upper}}, \tag{C.8}$$

where $\mathcal{N} = [N_1, 2N_2 - 1, \ldots, 2N_d - 1]$. Next, we calculate for given multi-index $\boldsymbol{i} \in \mathcal{N}$

$$\frac{\partial}{\partial u_{\boldsymbol{i}}} \langle \boldsymbol{A}, \boldsymbol{T}^{\text{H}}_{(\boldsymbol{n},d)}(\boldsymbol{u}) \rangle = \tag{C.9}$$

$$= \left\langle \boldsymbol{A}, \boldsymbol{S}_{N_1}^{i_1 - 1} \otimes \boldsymbol{S}_{N_2}^{i_2 - N_2} \otimes \cdots \otimes \boldsymbol{S}_{N_d}^{i_d - N_d} \right\rangle. \tag{C.10}$$

For a shorter notation we define the operator

$$\mathfrak{D}_{\boldsymbol{n},d} : \mathbb{C}^{M \times M} \to \mathbb{C}^{N_1 \times 2N_2 - 1 \cdots \times 2N_d - 1}$$

via

$$\boldsymbol{A} \mapsto \mathfrak{D}_{\boldsymbol{n},d}(\boldsymbol{A}) = \left(\frac{\partial}{\partial u_{\boldsymbol{i}}} \langle \boldsymbol{A}, \boldsymbol{T}_{\boldsymbol{n},d}(\boldsymbol{u}) \rangle \right)_{\boldsymbol{i} \in \mathcal{N}} =$$
$$= \left(\left\langle \boldsymbol{A}, \boldsymbol{S}_{N_1}^{i_1 - 1} \otimes \boldsymbol{S}_{N_2}^{i_2 - N_2} \otimes \cdots \otimes \boldsymbol{S}_{N_d}^{i_d - N_d} \right\rangle \right)_{\boldsymbol{i} \in \mathcal{N}}.$$

This operator results in a tensor with the same dimensions as $\boldsymbol{u}$ and each entry at position $\boldsymbol{i} \in \mathcal{N}$ represents the sum of the elements in $\boldsymbol{A}$ which occur at the same position as $\boldsymbol{u}_{\boldsymbol{i}}$ in $\boldsymbol{T}^{\text{H}}_{(\boldsymbol{n},d)}(\boldsymbol{u})$.

Now for some $\boldsymbol{i} \in \mathcal{N}$ and $\boldsymbol{u} = \boldsymbol{v} + \jmath \boldsymbol{w}$ we can also calculate (Note that here we identify the multi-index $\boldsymbol{i}$ with the the tensor of order d which has zeros everywhere except a single 1 at position $\boldsymbol{i}$.):

$$\frac{\partial}{\partial u_{\boldsymbol{i}}} \langle \boldsymbol{T}^{\text{H}}_{(\boldsymbol{n},d)}(\boldsymbol{u}), \boldsymbol{T}^{\text{H}}_{(\boldsymbol{n},d)}(\boldsymbol{u}) \rangle = \frac{\partial}{\partial u_{\boldsymbol{i}}} \left\| \boldsymbol{T}^{\text{H}}_{(\boldsymbol{n},d)}(\boldsymbol{u}) \right\|_F^2 =$$
$$= \frac{\partial}{\partial u_{\boldsymbol{i}}} \left(\sum_{\boldsymbol{i}' \in \mathcal{N}} \left[\left\| \boldsymbol{T}^{\text{H}}_{(\boldsymbol{n},d)}(\boldsymbol{i}') \right\|_F^2 v_{\boldsymbol{i}'}^2 - \left\| \boldsymbol{T}^{\text{H}}_{(\boldsymbol{n},d)}(\boldsymbol{i}') \right\|_F^2 w_{\boldsymbol{i}'}^2 \right] \right) =$$
$$= 2 f_{\boldsymbol{n}}(\boldsymbol{i}) \overline{u}_{\boldsymbol{i}},$$

where $f_{\boldsymbol{n}}(\boldsymbol{i})$ represents the number of occurrences of $u_{\boldsymbol{i}}$ in the Hermitian multilevel Toeplitz matrix $\boldsymbol{T}^{\text{H}}_{(\boldsymbol{n},d)}(\boldsymbol{u})$.

With this intuition at hand, we can easily infer that $(f_{\boldsymbol{n}}(\boldsymbol{i}))_{\boldsymbol{i} \in \mathcal{N}} = \mathfrak{D}_{\boldsymbol{n},d}(\boldsymbol{1})$, where $\boldsymbol{1} \in \mathbb{C}^{M \times m}$ is a matrix with all entries equal to 1.

Now, these rules together with (B.4) and (B.5) can be used to establish the following results about the partial derivatives of $\mathcal{L}$ which read as:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{W}} = \frac{\tau}{2} \boldsymbol{I}_K - \boldsymbol{\Lambda}_0 - \rho(\boldsymbol{V}_0 - \boldsymbol{W}), \tag{C.11}$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{u}} = \frac{\tau}{2} \boldsymbol{i}_1 - \mathfrak{D}_{\boldsymbol{n},d}(\hat{\boldsymbol{\Lambda}}) + \frac{\rho}{2} \left(\mathfrak{D}_{\boldsymbol{n},d}(\boldsymbol{1}) - 2\mathfrak{D}_{\boldsymbol{n},d}(\hat{\boldsymbol{V}}) \right), \tag{C.12}$$

$$\frac{\partial \mathcal{L}}{\partial \overline{\boldsymbol{Z}}} = \frac{1}{2}(\boldsymbol{\Phi}^{\mathrm{H}}\boldsymbol{\Phi}\boldsymbol{Z} - \boldsymbol{\Phi}^{\mathrm{H}}\boldsymbol{Y}) - \hat{\boldsymbol{\Lambda}} - \rho\left(\hat{\boldsymbol{V}} - \boldsymbol{Z}\right), \tag{C.13}$$

where $\boldsymbol{i}_1$ is the tensor of the same dimension as $\boldsymbol{u}$ with entries all equal to 0 except at the position of $u_{[1,\ldots,1]}$. With these three derivatives at hand we can proceed to formulate the explicit update steps for the ADMM iteration.

Bibliography

Publications with Contributions from the Author

[O1] S. Semper, F. Römer, T. Hotz, and G. Del Galdo. "Sparsity Order Estimation from a Single Compressed Observation Vector". In: *IEEE Transactions on Signal Processing* (2018) (cit. on pp. 5, 128).

[O2] C. Wagner and S. Semper. "Fast Linear Transformations in Python". In: *CoRR* (2017) (cit. on pp. 6, 97).

[O3] C. Wagner, S. Semper, and J. Kirchhof. "fastmat: Linear Transforms in Python". In: *SoftwareX (submitted)* (2021) (cit. on pp. 6, 97, 98).

[O4] S. Pawar, S. Semper, and F. Römer. "Combining Matrix Design for 2D DoA Estimation with Compressive Antenna Arrays Using Stochastic Gradient Descent". In: *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2019 (cit. on pp. 6, 7, 176, 202).

[O5] S. Semper and F. Römer. "ADMM for ND Line Spectral Estimation Using Grid-free Compressive Sensing from Multiple Measurements with Applications to DOA Estimation". In: *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2019 (cit. on pp. 6, 176).

[O6] S. Semper and T. Hotz. "Packing Bounds for Outer Products with Applications to Compressive Sensing". In: *Geometric Science of Information*. Springer International Publishing, 2019 (cit. on pp. 6, 23, 43–45, 57, 106).

[O7] S. Semper, M. Döbereiner, S. Pawar, M. Landmann, and G. Del Galdo. "eadf: Representation of far-field antenna responses in Python". In: *SoftwareX* (2020) (cit. on pp. 6, 184, 213).

[O8] S. Semper, J. Kirchhof, C. Wagner, F. Krieg, F. Römer, and G. Del Galdo. "Defect Detection From Compressed 3-D Ultrasonic Frequency Measurements". In: *2019 27th European Signal Processing Conference (EUSIPCO)*. 2019 (cit. on pp. 7, 106, 111, 116, 125, 148, 149).

[O9] J. Kirchhof, S. Semper, and F. Römer. "GPU-Accelerated Matrix-Free 3D Ultrasound Reconstruction for Nondestructive Testing". In: *2018 IEEE International Ultrasonics Symposium (IUS)*. 2018 (cit. on pp. 7, 106, 111, 113, 114, 136).

[O10] S. Semper, J. Kirchhof, C. Wagner, F. Krieg, F. Römer, A. Osman, and G. Del Galdo. "Defect Detection from 3D Ultrasonic Measurements Using Matrix-free Sparse Recovery Algorithms". In: *Proceedings of the 26th European Signal Processing Conference (EUSIPCO-2018)*. 2018 (cit. on pp. 7, 106, 111, 113, 157).

[O11] J. Kirchhof, S. Semper, C. W. Wagner, E. P'erez, F. Römer, and G. Del Galdo. "Frequency Sub-Sampling of Ultrasound Non-Destructive Measurements: Acquisition, Reconstruction and Performance". In: *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control* (2021) (cit. on pp. 7, 106, 110, 111).

[O12] C. Wagner, S. Semper, F. Römer, A. Schönfeld, and G. Del Galdo. "Hardware Architecture For Ultra-Wideband Channel Impulse Response Measurements Using Compressed Sensing". In: *Proceedings of the 28th European Signal Processing Conference (EUSIPCO-2020)*. 2020 (cit. on pp. 7, 106, 110).

[O13] M. Ibrahim, W. Al-Aqqad, F. Römer, M. Kaeske, S. Semper, R. S. Thomä, and G. Del Galdo. "Compressive Spatial Channel Sounding". In: *Proceedings of the 12th European Conference on Antennas and Propagation (EuCAP 2018)*. 2018 (cit. on p. 7).

[O14] C. Zhang, B. Zhu, S. Semper, A. Breitbarth, M. Rosenberger, and G. Notni. "A comparative investigation on the use of compressive sensing methods in computational ghost imaging". In: *Computational Imaging IV*. International Society for Optics and Photonics. SPIE, 2019 (cit. on p. 7).

[O15] E. Pérez, S. Semper, J. Kirchhof, F. Krieg, and F. Römer. “Compressed Ultrasound Computed Tomography in NDT (accepted)”. In: *2021 IEEE International Ultrasonics Symposium (IUS)*. 2021 (cit. on pp. 7, 106).

[O16] E. Pérez, J. Kirchhof, S. Semper, F. Krieg, and F. Römer. “Total Focusing Method with Subsampling in Space and Frequency Domain for Ultrasound NDT”. In: *2019 IEEE International Ultrasonics Symposium (IUS)*. 2019 (cit. on pp. 111, 113, 143, 156).

[O17] E. Pérez, J. Kirchhof, S. Semper, F. Krieg, and F. Römer. “Cramér-Rao Bounds for Flaw Localization in Subsampled Multistatic Multichannel Ultrasound Ndt Data”. In: *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2020 (cit. on pp. 111, 114).

[O18] S. Semper, F. Römer, T. Hotz, and G. Del Galdo. “Grid-Free Direction-of-Arrival Estimation with Compressed Sensing and Arbitrary Antenna Arrays”. In: *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2018)*. 2018 (cit. on p. 176).

Patents with Contributions from the Author

[P1] C. Wagner, F. Roemer, S. Semper, and G. D. Galdo. “METHOD FOR THE ACQUISITION OF IMPULSE RESPONSES, E.G. FOR ULTRA-WIDEBAND SYSTEMS”. Patent Application Number 19201805.9 (Submitted) (cit. on pp. 7, 110, 157).

References

[1] B. Chen, J. Wan, L. Shu, P. Li, M. Mukherjee, and B. Yin. "Smart Factory of Industry 4.0: Key Technologies, Application Case, and Challenges". In: *IEEE Access* (2018) (cit. on p. 1).

[2] G. Mattela, M. Tripathi, C. Pal, R. S. Dhiraj, and A. Acharyya. "An Efficient Pipeline for Distant Person Detection and Identification in 4K Video using GPUs". In: *2020 International Conference on COMmunication Systems NETworkS (COMSNETS)*. 2020 (cit. on p. 1).

[3] E. G. Larsson. "Massive MIMO for 5G: Overview and the road ahead". In: *2017 51st Annual Conference on Information Sciences and Systems (CISS)*. 2017 (cit. on p. 1).

[4] B. Duraisamy, T. Schwarz, and C. Wöhler. "On track-to-track data association for automotive sensor fusion". In: *2015 18th International Conference on Information Fusion (Fusion)*. 2015 (cit. on p. 1).

[5] M. Giammarini, D. Isidori, E. Concettoni, C. Cristalli, M. Fioravanti, and M. Pieralisi. "Design of Wireless Sensor Network for Real-Time Structural Health Monitoring". In: *2015 IEEE 18th International Symposium on Design and Diagnostics of Electronic Circuits Systems*. 2015 (cit. on p. 1).

[6] M. Lustig, D. L. Donoho, and J. M. Pauly. "Sparse MRI: The application of compressed sensing for rapid MR imaging". In: *Magnetic resonance in medicine* 6 (2007) (cit. on p. 3).

[7] M. Rossi, A. M. Haimovich, and Y. C. Eldar. "Spatial compressive sensing in MIMO radar with random arrays". In: *Information Sciences and Systems (CISS), 2012 46th Annual Conference on*. IEEE. 2012 (cit. on pp. 3, 178).

[8] K. Han, Y. Wang, B. Kou, and W. Hong. "Parameters estimation using a random linear array and compressed sensing". In: *International Congress on Image and Signal Processing (CISP)*. 2010 (cit. on pp. 3, 178).

[9] D. M. Malioutov, M. Cetin, and A. S. Willsky. "A sparse signal reconstruction perspective for source localization with sensor arrays". In: *IEEE Transactions on Signal Processing* 8 (2005) (cit. on pp. 3, 66).

[10] Y. Wang, Z. Tian, and C. Feng. "Sparsity Order Estimation and its Application in Compressive Spectrum Sensing for Cognitive Radios". In: *IEEE Transactions on Wireless Communications* 6 (2012) (cit. on pp. 3, 49).

[11] E. J. Candes, J. Romberg, and T. Tao. "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information". In: *IEEE Transactions on Information Theory* 2 (2006) (cit. on pp. 3, 113, 123).

[12] S. Foucart and H. Rauhut. *A mathematical introduction to compressive sensing*. Birkhäuser, 2013 (cit. on pp. 10, 17, 18, 20, 21, 24, 25, 30, 31, 35, 36, 43).

[13] M. Elad. *Sparse and Redundant Representations - From Theory to Applications in Signal and Image Processing*. Springer, 2010 (cit. on pp. 10, 19, 23).

[14] D. Werner. *Functional analysis. (Funktionalanalysis.)* German. Springer-Lehrbuch. Berlin: Springer., 2011 (cit. on pp. 10, 11).

[15] Y. C. Eldar. *Sampling theory: Beyond bandlimited systems*. Cambridge University Press, 2015 (cit. on pp. 11, 22, 113).

[16] G. M. Gibson, S. D. Johnson, and M. J. Padgett. "Single-pixel imaging 12 dates on: a review". In: *Opt. Express* 19 (2020) (cit. on p. 13).

[17] K. Gedalyahu and Y. C. Eldar. "Time delay estimation: Compressed sensing over an infinite union of subspaces". In: *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*. 2010 (cit. on p. 16).

[18] D. L. Donoho and M. Elad. "Optimally sparse representation in general (nonorthogonal) dictionaries via l1 minimization". In: *Proceedings of the National Academy of Sciences* 5 (2003) (cit. on p. 20).

[19] D. L. Donoho. "Compressed sensing". In: *IEEE Transactions on Information Theory* 4 (2006) (cit. on p. 20).

[20] Y. de Castro and F. Gamboa. "Exact reconstruction using Beurling minimal extrapolation". In: *Journal of Mathematical Analysis and Applications* 1 (2012) (cit. on p. 20).

[21] E. J. Candès, Y. C. Eldar, D. Needell, and P. Randall. "Compressed sensing with coherent and redundant dictionaries". In: *Applied and Computational Harmonic Analysis* 1 (2011) (cit. on pp. 26, 113).

[22] E. Crespo Marques, N. Maciel, L. Naviner, H. Cai, and J. Yang. "A Review of Sparse Recovery Algorithms". In: *IEEE Access* (2019) (cit. on p. 27).

[23] Y. Yang and M. Pesavento. "A Unified Successive Pseudoconvex Approximation Framework". In: *IEEE Trans. on Signal Processing* 13 (2017) (cit. on pp. 27, 33).

[24] H. Rauhut. "Compressive sensing and structured random matrices". In: *Theoretical foundations and numerical methods for sparse recovery* (2010) (cit. on p. 27).

[25] D. L. Donoho and Y. Tsaig. "Fast Solution ofℓ_1-Norm Minimization Problems When the Solution May Be Sparse". In: *IEEE Transactions on Information Theory* 11 (2008) (cit. on p. 27).

[26] L. Scharf and C. Demeure. *Statistical Signal Processing: Detection, Estimation, and Time Series Analysis*. Addison-Wesley series in electrical and computer engineering. Addison-Wesley Publishing Company, 1991 (cit. on pp. 28, 45, 46, 179).

[27] B. Mamandipoor, D. Ramasamy, and U. Madhow. "Newtonized Orthogonal Matching Pursuit: Frequency Estimation Over the Continuum". In: *IEEE Transactions on Signal Processing* 19 (2016) (cit. on pp. 28, 33).

[28] D. L. Donoho, Y. Tsaig, I. Drori, and J.-L. Starck. "Sparse Solution of Underdetermined Systems of Linear Equations by Stagewise Orthogonal Matching Pursuit". In: *IEEE Transactions on Information Theory* 2 (2012) (cit. on p. 29).

[29] D. Needell and J. Tropp. "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples". In: *Applied and Computational Harmonic Analysis* 3 (2009) (cit. on p. 29).

[30] A. Beck and M. Teboulle. "A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems". In: *SIAM Journal on Imaging Sciences* 1 (2009) (cit. on pp. 30, 32, 135).

[31] T. Hastie, R. Tibshirani, and M. Wainwright. *Statistical Learning with Sparsity: The Lasso and Generalizations*. Chapman & Hall/CRC, 2015 (cit. on p. 31).

[32] I. Daubechies, M. Defrise, and C. De Mol. "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint". In: *Communications on Pure and Applied Mathematics* 11 (2004) (cit. on p. 32).

[33] Y. E. NESTEROV. "A method for solving the convex programming problem with convergence rate $O(1/k^2)$". In: *Dokl. Akad. Nauk SSSR* (1983) (cit. on p. 32).

[34] A. Chambolle and C. H. Dossal. "On the convergence of the iterates of "FISTA"". In: *Journal of Optimization Theory and Applications* Issue 3 (2015) (cit. on p. 33).

[35] J. M. Bioucas-Dias and M. A. T. Figueiredo. "A New TwIST: Two-Step Iterative Shrinkage/Thresholding Algorithms for Image Restoration". In: *IEEE Transactions on Image Processing* 12 (2007) (cit. on p. 33).

[36] P. Boufounos, M. F. Duarte, and R. G. Baraniuk. "Sparse signal reconstruction from noisy compressive measurements using cross validation". In: *Statistical Signal Processing, 2007. SSP'07. IEEE/SP 14th Workshop on*. IEEE. 2007 (cit. on pp. 33, 49).

[37] A. M. Tillmann and M. E. Pfetsch. "The Computational Complexity of the Restricted Isometry Property, the Nullspace Property, and Related Concepts in Compressed Sensing". In: *IEEE Transactions on Information Theory* 2 (2014) (cit. on p. 34).

[38] M. Ibrahim, V. Ramireddy, A. Lavrenko, J. König, F. Römer, M. Landmann, M. Grossmann, G. D. Galdo, and R. S. Thomä. "Design and analysis of compressive antenna arrays for direction of arrival estimation". In: *Elsevier Signal Processing* (2017) (cit. on pp. 38, 203, 204).

[39] N. Ailon and H. Rauhut. "Fast and RIP-Optimal Transforms". In: *Discrete & Computational Geometry* 4 (2014) (cit. on p. 38).

[40] I. Haviv and O. Regev. "The Restricted Isometry Property of Subsampled Fourier Matrices". In: *Proc. of the 27th Annual ACM-SIAM Symp. on Discrete Algorithms*. 2015 (cit. on pp. 38, 39, 113, 121, 128).

[41] F. Krahmer and R. Ward. "New and Improved Johnson–Lindenstrauss Embeddings via the Restricted Isometry Property". In: *SIAM Journal on Mathematical Analysis* 3 (2011) (cit. on pp. 41, 113, 121, 128).

[42] Dhillon, I. S. and Heath, Jr. R. W. and Strohmer, T. and Tropp, J. A. "Constructing Packings in Grassmannian Manifolds via Alternating Projection." In: *Experimental Mathematics* 1 () (cit. on pp. 42, 45, 57).

[43] H. Zörlein and M. Bossert. "Coherence Optimization and Best Complex Antipodal Spherical Codes". In: *IEEE Transactions on Signal Processing* 24 (2015) (cit. on pp. 43, 45).

[44] V. Abolghasemi, S. Ferdowsi, and S. Sanei. "A gradient-based alternating minimization approach for optimization of the measurement matrix in compressive sensing". In: *Signal Processing* 4 (2012) (cit. on p. 44).

[45] M. Mangia, F. Pareschi, R. Rovatti, and G. Setti. "Adapted Compressed Sensing: A Game Worth Playing". In: *IEEE Circuits and Systems Magazine* 1 (2020) (cit. on p. 44).

[46] G. Li, Z. Zhu, D. Yang, L. Chang, and H. Bai. "On Projection Matrix Optimization for Compressive Sensing Systems". In: *IEEE Transactions on Signal Processing* 11 (2013) (cit. on p. 44).

[47] M. Thill and B. Hassibi. "Low-Coherence Frames From Group Fourier Matrices". In: *IEEE Transactions on Information Theory* 6 (2017) (cit. on p. 45).

[48] P. Stoica, R. L. Moses, et al. *Spectral analysis of signals*. Pearson Prentice Hall Upper Saddle River, NJ, 2005 (cit. on p. 46).

[49] Y. Tevfik and A. Hüseyin. "A survey of spectrum sensing algorithms for cognitive radio applications". In: *Communications Surveys & Tutorials, IEEE* 1 (2009) (cit. on pp. 48, 176).

[50] M. F. Duarte, M. A. Davenport, D. Takhar, J. N. Laska, T. Sun, K. E. Kelly, R. G. Baraniuk, et al. "Single-pixel imaging via compressive sampling". In: *IEEE Signal Processing Magazine* 2 (2008) (cit. on pp. 48, 157).

[51] S. K. Sharma, S. Chatzinotas, and B. Ottersten. "Compressive Sparsity Order Estimation for Wideband Cognitive Radio Receiver". In: *IEEE Transactions on Signal Processing* 19 (2014) (cit. on pp. 49, 50).

[52] A. Lavrenko, F. Römer, G. Del Galdo, R. Thomä, and O. Arikan. "An empirical eigenvalue-threshold test for sparsity level estimation from compressed measurements". In: *Signal Processing Conference (EUSIPCO), 2014 Proceedings of the 22nd European*. 2014 (cit. on pp. 49, 50, 56, 61).

[53] A. Lavrenko, F. Römer, G. DelGaldo, and R. S. Thomä. "Sparsity order estimation for sub-Nyquist sampling and recovery of sparse multiband signals." In: *ICC*. IEEE, 2015 (cit. on pp. 49, 50, 128).

[54] A. Lavrenko, F. Römer, G. DelGaldo, T. R., and O. Arikan. "Detection of time-varying support via rank evolution approach for effective joint sparse recovery". In: *Proc. 23rd European Sig. Proc. Conf. (EUSIPCO-2015)*. 2015 (cit. on pp. 49, 50, 128).

[55] R. Ward. "Compressed sensing with cross validation". In: *IEEE Transactions on Information Theory* 12 (2009) (cit. on p. 49).

[56] D. M. Malioutov, S. Sanghavi, and A. S. Willsky. "Compressed sensing with sequential observations". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2008)*. 2008 (cit. on p. 49).

[57] M. E. Lopes. "Estimating Unknown Sparsity in Compressed Sensing". In: *CoRR* (2012) (cit. on pp. 50, 61–64, 67, 69).

[58] C. Ravazzi, S. M. Fosson, T. Bianchi, and E. Magli. "Signal sparsity estimation from compressive noisy projections via γ-sparsified random matrices". In: *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2016)*. 2016 (cit. on pp. 50, 64, 128, 231).

[59] P. Stoica and Y. Selen. "Model-order selection: a review of information criterion rules". In: *IEEE Signal Processing Magazine* 4 (2004) (cit. on pp. 50, 56).

[60] T.-J. Shan, M. Wax, and T. Kailath. "On spatial smoothing for direction-of-arrival estimation of coherent signals". In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 4 (1985) (cit. on p. 54).

[61] A. Quinlan, J.-P. Barbot, P. Larzabal, and M. Haardt. "Model Order Selection for Short Data: An Exponential Fitting Test (EFT)". In: *EURASIP J. Appl. Signal Process.* 1 (2007) (cit. on p. 56).

[62] M. F. Duarte and Y. C. Eldar. "Structured Compressed Sensing: From Theory to Applications". In: *IEEE Transactions on Signal Processing* 9 (2011) (cit. on p. 66).

[63] P. Knabner and L. Angermann. *Numerical Methods for Elliptic and Parabolic Partial Differential Equations*. Springer, New York, NY, 2003 (cit. on pp. 72, 95).

[64] T. A. Davis and Y. Hu. "The University of Florida Sparse Matrix Collection". In: *ACM Trans. Math. Softw.* 1 (2011) (cit. on p. 72).

[65] D. Ludwig. "The Radon Transform on Euclidean Space". In: *Communications on pure and applied mathematics* (1966) (cit. on p. 72).

[66] E. T. Quinto. "An Introduction to X-ray tomography and Radon Transforms". In: *Providence, RI: American Mathematical Society (AMS)* (2006) (cit. on p. 72).

[67] R. K. R. Yarlagadda and J. E. Hershey. *Hadamard matrix analysis and synthesis : with applications to communications and signal/image processing*. The Kluwer international series in engineering and computer science ; 383. Literaturverz. S. [117] - 120. Kluwer Acad. Publ., 1997 (cit. on p. 72).

[68] J. Cooley and J. Tukey. "An Algorithm for the Machine Calculation of Complex Fourier Series". In: *Mathematics of Computation* 90 (1965) (cit. on p. 76).

[69] S. G. Johnson and M. Frigo. "Implementing FFTs in Practice". In: *Fast Fourier Transforms*. Ed. by C. S. Burrus. Connexions, 2008. Chap. 11 (cit. on p. 76).

[70] M. Frigo. "A fast Fourier transform compiler". In: *Proc. 1999 ACM SIGPLAN Conf. on Programming Language Design and Implementation*. 5. ACM, 1999 (cit. on p. 76).

[71] M. Frigo and S. G. Johnson. "FFTW: An adaptive software architecture for the FFT". In: *Proc. 1998 IEEE Intl. Conf. Acoustics Speech and Signal Processing*. IEEE, 1998 (cit. on p. 76).

[72] M. R. Hestenes and E. Stiefel. "Methods of Conjugate Gradients for Solving Linear Systems". In: *Journal of Research of the National Bureau of Standards* 6 (1952) (cit. on pp. 76, 95).

[73] R. B. Lehoucq and D. C. Sorensen. "Deflation Techniques for an Implicitly Restarted Arnoldi Iteration". In: *SIAM Journal on Matrix Analysis and Applications* 4 (1996) (cit. on pp. 76, 137).

[74] S. Diamond and S. Boyd. "Convex Optimization with Abstract Linear Operators". In: *2015 IEEE International Conference on Computer Vision (ICCV)*. 2015 (cit. on p. 76).

[75] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. "Backpropagation Applied to Handwritten Zip Code Recognition". In: *Neural Computation* 4 (1989) (cit. on p. 76).

[76] Fino and Algazi. "Unified Matrix Treatment of the Fast Walsh-Hadamard Transform". In: *IEEE Transactions on Computers* 11 (1976) (cit. on pp. 81, 94, 167).

[77] P. Fernandes, B. Plateau, and W. J. Stewart. "Efficient Descriptor-Vector Multiplications in Stochastic Automata Networks." In: *J. ACM* 3 (1998) (cit. on pp. 84, 85, 89).

[78] P. J. Davis. *Circulant Matrices*. 1st ed. Pure & Applied Mathematics. John Wiley & Sons Inc, 1979 (cit. on p. 89).

[79] W. Chen, C. Smith, and S. Fralick. "A Fast Computational Algorithm for the Discrete Cosine Transform". In: *IEEE Trans. Commun.* (1977) (cit. on p. 94).

[80] S. Mallat. *A Wavelet Tour of Signal Processing, Third Edition: The Sparse Way*. 3rd. Academic Press, Inc., 2008 (cit. on p. 94).

[81] T. A. Davis and Y. Hu. "The University of Florida Sparse Matrix Collection". In: *ACM Trans. Math. Softw.* 1 (2011) (cit. on p. 95).

[82] H. A. van der Vorst. "Bi-CGSTAB: A Fast and Smoothly Converging Variant of Bi-CG for the Solution of Nonsymmetric Linear Systems". In: *SIAM Journal on Scientific and Statistical Computing* 2 (1992) (cit. on p. 95).

[83] C. C. Paige and M. A. Saunders. "LSQR: An Algorithm for Sparse Linear Equations and Sparse Least Squares". In: *ACM Trans. Math. Softw.* 1 (1982) (cit. on p. 95).

[84] P. S. Foundation. *Python*. Version 3. 12, 2018 (cit. on p. 97).

[85] T. E. Oliphant. *A guide to NumPy*. Trelgol Publishing USA, 2006 (cit. on p. 99).

[86] W. Gehrke. *Fortran 90 Language Guide*. Springer-Verlag, 1995 (cit. on p. 99).

[87] B. W. Kernighan and D. M. Ritchie. *The C Programming Language*. 2nd. Prentice Hall Professional Technical Reference, 1988 (cit. on p. 99).

[88] E. Jones, T. Oliphant, P. Peterson, et al. *SciPy: Open source scientific tools for Python*. 12, 2018 (cit. on pp. 99, 105).

[89] S. Behnel, R. Bradshaw, C. Citro, L. Dalcin, D. S. Seljebotn, and K. Smith. "Cython: The Best of Both Worlds". In: *Computing in Science Engineering* 2 (2011) (cit. on p. 99).

[90] R. Okuta, Y. Unno, D. Nishino, S. Hido, and C. Loomis. "CuPy: A NumPy-Compatible Library for NVIDIA GPU Calculations". In: *Proceedings of Workshop on Machine Learning Systems (LearningSys) in The Thirty-first Annual Conference on Neural Information Processing Systems (NIPS)*. 2017 (cit. on pp. 105, 133).

[91] M. Ravasi and I. Vasconcelos. "PyLops—A linear-operator Python library for scalable algebra and optimization". In: *SoftwareX* (2020) (cit. on p. 106).

[92] A. D. Padula, S. D. Scott, and W. W. Symes. "A Software Framework for Abstract Expression of Coordinate-Free Linear Algebra and Optimization Algorithms". In: *ACM Trans. Math. Softw.* 2 (2009) (cit. on p. 106).

[93] F. Krieg, J. Kirchhof, F. Romer, R. Pandey, A. Ihlow, G. del Galdo, and A. Osman. "Progressive Online 3-D SAFT Processing by Matrix Structure Exploitation". In: *2018 IEEE International Ultrasonics Symposium (IUS)*. IEEE, 2018 (cit. on p. 106).

[94] F. Krieg, J. Kirchhof, E. Pérez, T. Schwender, F. Römer, and A. Osman. "Locally Optimal Subsampling Strategies for Fullmatrix Capture Measurements in Pipe Inspection". In: *3rd International Symposium on Structural Health Monitoring and Nondestructive Testing (SHM-NDT)*. 2020 (cit. on p. 106).

[95] K. Weise, L. Poßner, E. Müller, R. Gast, and T. R. Knösche. "Pygpc: A sensitivity and uncertainty analysis toolbox for Python". In: *SoftwareX* (2020) (cit. on p. 106).

[96] M. Yadav, D. Sheldon, and C. Musco. "Faster Kernel Interpolation for Gaussian Processes". In: *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*. Ed. by A. Banerjee and K. Fukumizu. Proceedings of Machine Learning Research. PMLR, 2021 (cit. on p. 106).

[97] J. Krautkrämer and H. Krautkrämer. *Ultrasonic testing of materials*. Springer Science & Business Media, 2013 (cit. on p. 112).

[98] M. Spies, H. Rieder, A. Dillhöfer, V. Schmitz, and W. Müller. "Synthetic Aperture Focusing and Time-of-Flight Diffraction Ultrasonic Imaging—Past and Present". In: *Journal of NDE* (2012) (cit. on p. 112).

[99] C. Holmes, B. W. Drinkwater, and P. D. Wilcox. "Post-processing of the full matrix of ultrasonic transmit–receive array data for non-destructive evaluation". In: *NDT & E International* 8 (2005) (cit. on p. 112).

[100] B. Berthon, P. Morichau-Beauchant, J. Porée, A. Garofalakis, B. Tavitian, M. Tanter, and J. Provost. "Spatiotemporal matrix image formation for programmable ultrasound scanners". In: *Physics in Medicine & Biology* 3 (2018) (cit. on p. 112).

[101] E. Carcreff, S. Bourguignon, J. Idier, and L. Simon. "A linear model approach for ultrasonic inverse problems with attenuation and dispersion". In: *IEEE Trans. on Ultrasonics, Ferroelectrics, and Frequency Control* 7 (2014) (cit. on p. 112).

[102] J. Kirchhof, F. Krieg, F. Römer, A. Ihlow, A. Osman, and G. Del Galdo. "Sparse Signal Recovery for Ultrasonic Detection and Reconstruction of Shadowed Flaws". In: *IEEE ICASSP*. 2017 (cit. on p. 113).

[103] N. Quaegebeur and P. Masson. "Correlation-based imaging technique using ultrasonic transmit–receive array for non-destructive evaluation". In: *Ultrasonics* 8 (2012) (cit. on pp. 113, 114, 129).

[104] R. Demirli and J. Saniie. "Model-based estimation of ultrasonic echoes. Part I: Analysis and algorithms". In: *IEEE Trans. on Ultrasonics, Ferroelectrics, and Frequency Control* 3 (2001) (cit. on pp. 113, 156).

[105] C. Quinsac, A. Basarab, and D. Kouamé. "Frequency Domain Compressive Sampling for Ultrasound Imaging". In: *Advances in Acoustics and Vibration* (2012) (cit. on p. 113).

[106] M. Mangia, F. Pareschi, R. Rovatti, and G. Setti. "Adapted Compressed Sensing: A Game Worth Playing". In: *IEEE Circuits and Systems Magazine* 1 (2020) (cit. on p. 113).

[107] T. Chernyakova and Y. Eldar. "Fourier-domain beamforming: the path to compressed ultrasound imaging". In: *IEEE Trans. on Ultrasonics, Ferroelectrics, and Frequency Control* 8 (2014) (cit. on p. 113).

[108] M. Lehne. "An Analog/Mixed Signal FFT Processor for Ultra-Wideband OFDM Wireless Transceivers". PhD thesis. Virginia Polytechnic Institute and State University, 2008 (cit. on p. 113).

[109] R. Tur, Y. C. Eldar, and Z. Friedman. "Innovation Rate Sampling of Pulse Streams With Application to Ultrasound Imaging". In: *IEEE Trans. on Signal Processing* 4 (2011) (cit. on pp. 113, 115).

[110] J. Spaulding, Y. C. Eldar, and B. Murmann. "Mixer-based sub-array beamforming for sub-Nyquist sampling ultrasound architectures". In: *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2015)*. 2015 (cit. on p. 113).

[111] S. Mulleti, K. Lee, and Y. C. Eldar. "Identifiability Conditions for Compressive Multichannel Blind Deconvolution". In: *IEEE Trans. on Signal Processing* (2020) (cit. on pp. 113, 115, 142, 143).

[112] A. Besson, R. E. Carrillo, O. Bernard, Y. Wiaux, and J.-P. Thiran. "Compressed delay-and-sum beamforming for ultrafast ultrasound imaging". In: *IEEE Int. Conference on Image Processing (ICIP)*. 2016 (cit. on p. 113).

[113] A. Ramkumar and A. K. Thittai. "Strategic Undersampling and Recovery Using Compressed Sensing for Enhancing Ultrasound Image Quality". In: *IEEE Trans. on Ultrasonics, Ferroelectrics, and Frequency Control* 3 (2020) (cit. on p. 113).

[114] A. Tuysuzoglu, J. M. Kracht, R. O. Cleveland, M. Çetin, and W. C. Karl. "Sparsity driven ultrasound imaging". In: *The Journal of the Acoustical Society of America* 2 (2012) (cit. on p. 113).

[115] N. Laroche, S. Bourguignon, E. Carcreff, J. Idier, and A. Duclos. "An inverse approach for ultrasonic imaging from full matrix capture data. Application to resolution enhancement in NDT." In: *IEEE Trans. on Ultrasonics, Ferroelectrics, and Frequency Control* (2020) (cit. on pp. 114, 147, 148).

[116] M. Karimi, P. Croaker, and N. Kessissoglou. "Acoustic scattering for 3D multi-directional periodic structures using the boundary element method". In: *The Journal of the Acoustical Society of America* 1 (2017) (cit. on p. 114).

[117] P. R. Bueno, M. V. W. Zibetti, and J. M. Maia. "Real-time ultrasound image reconstruction as an inverse problem on a GPU". In: *Journal of Real-Time Image Processing* (2018) (cit. on p. 114).

[118] S. M. Kay. *Fundamentals of Statistical Signal Processing: Estimation Theory*. Prentice-Hall, Inc., 1993 (cit. on p. 114).

[119] W. F. Walker and G. E. Trahey. "A fundamental limit on delay estimation using partially correlated speckle signals". In: *IEEE Trans. on Ultrasonics, Ferroelectrics, and Frequency Control* 2 (1995) (cit. on p. 114).

[120] F. Viola and W. F. Walker. "A Comparison of the Performance of Time-Delay Estimators in Medical Ultrasound". In: *IEEE Trans. on Ultrasonics, Ferroelectrics, and Frequency Control* 4 (2003) (cit. on p. 114).

[121] Y. Desailly, J. Pierre, O. Couture, and M. Tanter. "Resolution limits of ultrafast ultrasound localization microscopy". In: *Physics in Medicine & Biology* 22 (2015) (cit. on p. 114).

[122] Q. An and Y. Shen. "Camera Configuration Design in Cooperative Active Visual 3d Reconstruction: A Statistical Approach". In: *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2020)*. 2020 (cit. on p. 114).

[123] Y. Gershon, Y. Buchris, and I. Cohen. "Greedy Sparse Array Design for Optimal Localization under Spatially Prioritized Source Distribution". In: *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2020)*. 2020 (cit. on p. 114).

[124] P. Cawley. "Structural health monitoring: Closing the gap between research and industrial deployment". In: *Structural Health Monitoring* 5 (2018) (cit. on pp. 114, 141).

[125] A. Mamistvalov and Y. C. Eldar. "Sparse Convolutional Beamforming for Wireless Ultrasound". In: *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2020)*. 2020 (cit. on p. 115).

[126] B. Valeske, A. Osman, F. Römer, and R. Tschuncky. "Next Generation NDE Sensor Systems as IIoT Elements of Industry 4.0". In: *Research in Nondestructive Evaluation* (2020) (cit. on p. 115).

[127] X. P. Qing, S. J. Beard, A. Kumar, T. K. Ooi, and F.-K. Chang. "Built-in Sensor Network for Structural Health Monitoring of Composite Structure". In: *Journal of Intelligent Material Systems and Structures* 1 (2007) (cit. on p. 115).

[128] F. Lingvall, T. Olofsson, and T. Stepinski. "Synthetic aperture imaging using sources with finite aperture: Deconvolution of the spatial impulse response". In: *The Journal of the Acoustical Society of America* 1 (2003) (cit. on pp. 129, 141).

[129] L. Bluestein. "A linear filtering approach to the computation of discrete Fourier transform". In: *IEEE Trans. on Audio and Electroacoustics* 4 (1970) (cit. on p. 132).

[130] R. Lehoucq, D. Sorensen, and C. Yang. *ARPACK Users' Guide: Solution of Large-scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods*. Software, Environments, Tools. Society for Industrial and Applied Mathematics, 1998 (cit. on p. 137).

[131] G. Goertzel. “An Algorithm for the Evaluation of Finite Trigonometric Series”. In: *The American Mathematical Monthly* 1 (1958) (cit. on p. 143).

[132] J. Nickolls, I. Buck, M. Garland, and K. Skadron. “Scalable Parallel Programming with CUDA”. In: *Queue* 2 (2008) (cit. on p. 144).

[133] J. Ender. “On compressive sensing applied to radar”. In: *Signal Processing* 5 (2010) (cit. on p. 157).

[134] R. Baraniuk and P. Steeghs. “Compressive radar imaging”. In: *Radar Conference, 2007 IEEE*. IEEE. 2007 (cit. on p. 157).

[135] M. Lustig, D. L. Donoho, J. M. Santos, and J. M. Pauly. “Compressed Sensing MRI”. In: *IEEE Signal Processing Magazine* 2 (2008) (cit. on p. 157).

[136] T. Zhou, C. Tao, S. Salous, L. Liu, and Z. Tan. “Channel sounding for high-speed railway communication systems”. In: *IEEE Communications Magazine* 10 (2015) (cit. on p. 157).

[137] J. Sachs. “Ultra-Wideband Radar”. In: *Handbook of Ultra-Wideband Short-Range Sensing*. John Wiley & Sons, Ltd, 2012. Chap. 4 (cit. on pp. 158, 160).

[138] J. Sachs, P. Peyerl, and M. Rossberg. “A new UWB-principle for sensor-array application”. In: *IMTC/99. Proceedings of the 16th IEEE Instrumentation and Measurement Technology Conference (Cat. No.99CH36309)*. 1999 (cit. on pp. 158, 161, 162, 165, 169).

[139] G. Itzhak, N. Wagner, and E. Shoshan. “A sub-Nyquist radar prototype: Hardware and algorithms”. In: *IEEE Transactions on Aerospace and Electronic Systems, special issue on Compressed Sensing for Radar, Aug. 2012* () (cit. on p. 158).

[140] M. Mishali and Y. C. Eldar. “Sub-Nyquist Sampling”. In: *IEEE Signal Processing Magazine* 6 (2011) (cit. on p. 158).

[141] T. Thiasiriphet, M. Ibrahim, and J. Lindner. “Compressed Sensing for UWB medical radar applications”. In: *2012 IEEE International Conference on Ultra-Wideband*. 2012 (cit. on pp. 158, 164).

[142] J. A. Tropp, J. N. Laska, M. F. Duarte, J. K. Romberg, and R. G. Baraniuk. “Beyond Nyquist: Efficient sampling of sparse bandlimited signals”. In: *Information Theory, IEEE Transactions on* 1 (2010) (cit. on pp. 158, 164).

[143] J. N. Laska, S. Kirolos, M. F. Duarte, T. S. Ragheb, R. G. Baraniuk, and Y. Massoud. "Theory and Implementation of an Analog-to-Information Converter using Random Demodulation". In: *2007 IEEE International Symposium on Circuits and Systems*. 2007 (cit. on p. 158).

[144] D. V. Sarwate and M. B. Pursley. "Crosscorrelation properties of pseudorandom and related sequences". In: *Proceedings of the IEEE* 5 (1980) (cit. on p. 162).

[145] T. Chernyakova and Y. C. Eldar. "Exploiting FRI signal structure for sub-Nyquist sampling and processing in medical ultrasound". In: *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2015 (cit. on p. 164).

[146] A. Kulkarni and T. Mohsenin. "Low Overhead Architectures for OMP Compressive Sensing Reconstruction Algorithm". In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 6 (2017) (cit. on pp. 165, 169).

[147] M. Cohn and A. Lempel. "On fast M-sequence transforms (Corresp.)" In: *IEEE Transactions on Information Theory* 1 (1977) (cit. on p. 167).

[148] P. Galajda, S. Slovak, M. Sokol, M. Pecovsky, and M. Kmec. "Integrated M-Sequence Based Transceiver for UWB Sensor Networks". In: *Radioengineering* (2019) (cit. on p. 169).

[149] Y. Wang, G. Leus, and A. Pandharipande. "Direction estimation using compressive sampling array processing". In: *Statistical Signal Processing, 2009. SSP'09. IEEE/SP 15th Workshop on*. IEEE. 2009 (cit. on pp. 176, 178).

[150] M. B. Hawes and W. Liu. "Compressive sensing-based approach to the design of linear robust sparse antenna arrays with physical size constraint". In: *IET Microwaves, Antennas Propagation* 10 (2014) (cit. on pp. 176, 178).

[151] J.-F. Gu, W.-P. Zhu, and M. Swamy. "Compressed sensing for DOA estimation with fewer receivers than sensors". In: *2011 IEEE International Symposium of Circuits and Systems (ISCAS)*. 2011 (cit. on p. 178).

[152] S. Shakeri, D. D. Ariananda, and G. Leus. "Direction of arrival estimation using sparse ruler array design". In: *Signal Processing Advances in Wireless Communications (SPAWC), 2012 IEEE 13th International Workshop on*. IEEE. 2012 (cit. on p. 178).

[153] M. Ibrahim, F. Römer, R. Alieiev, G. Del Galdo, and R. S. Thomä. "On the estimation of grid offsets in CS-based direction-of-arrival estimation". In: *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2014)*. 2014 (cit. on p. 179).

[154] E. J. Candès and C. Fernandez-Granda. "Super-Resolution from Noisy Data". In: *Journal of Fourier Analysis and Applications* 6 (2013) (cit. on pp. 179, 182, 188).

[155] E. J. Candès and C. Fernandez-Granda. "Towards a Mathematical Theory of Super-resolution". In: *Communications on Pure and Applied Mathematics* 6 (2014) (cit. on pp. 179, 182).

[156] H. Krim and M. Viberg. "Two decades of array signal processing research: the parametric approach". In: *IEEE Signal Processing Magazine* 4 (1996) (cit. on pp. 179, 181).

[157] W. Tang, H. Jiang, and S. Pang. "Coprime Array Interpolation for Direction of Arrival Estimation Based on Atomic Norm Minimization". In: *2019 IEEE Radar Conference (RadarConf)*. 2019 (cit. on p. 180).

[158] Y. Wang, P. Xu, and Z. Tian. "Efficient channel estimation for massive MIMO systems via truncated two-dimensional atomic norm minimization". In: *2017 IEEE International Conference on Communications (ICC)*. 2017 (cit. on p. 180).

[159] Y. Tsai, L. Zheng, and X. Wang. "Millimeter-Wave Beamformed Full-Dimensional MIMO Channel Estimation based on Atomic Norm Minimization". In: *IEEE Transactions on Communications* 12 (2018) (cit. on pp. 180, 189).

[160] A. Govinda Raj and J. H. McClellan. "Single Snapshot Super-Resolution DOA Estimation for Arbitrary Array Geometries". In: *IEEE Signal Processing Letters* 1 (2019) (cit. on p. 180).

[161] R. S. Thomä, M. Landmann, and A. Richter. "RIMAX - A Maximum Likelihood Framework for Parameter Estimation in Multidimensional Channel Sounding". In: *Proceedings of the ISAP 2004*. 2004 (cit. on pp. 180, 222).

[162] M. Steinbauer, A. F. Molisch, and E. Bonek. "The double-directional radio channel". In: *IEEE Antennas and Propagation Magazine* 4 (2001) (cit. on pp. 180, 194, 222).

[163] R. Roy and T. Kailath. "ESPRIT-estimation of signal parameters via rotational invariance techniques". In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 7 (1989) (cit. on pp. 181, 198).

[164] R. Schmidt. "Multiple emitter location and signal parameter estimation". In: *IEEE Transactions on Antennas and Propagation* 3 (1986) (cit. on p. 181).

[165] F. Römer, M. Haardt, and G. DelGaldo. "Analytical Performance Assessment of Multidimensional Matrix- and Tensor-Based ESPRIT-Type Algorithms". In: *IEEE Transactions on Signal Processing* 10 (2014) (cit. on p. 181).

[166] C. Cortes and V. Vapnik. "Support-Vector Networks". In: *Machine Learning*. 1995 (cit. on p. 181).

[167] M. E. Tipping. "Sparse Bayesian Learning and the Relevance Vector Machine". In: *J. Mach. Learn. Res.* (2001) (cit. on p. 181).

[168] M. E. Tipping, A. Faul, J. J. T. Avenue, and J. J. T. Avenue. "Fast Marginal Likelihood Maximisation for Sparse Bayesian Models". In: *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*. 2003 (cit. on p. 181).

[169] Z. Yang, L. Xie, and C. Zhang. "Off-Grid Direction of Arrival Estimation Using Sparse Bayesian Inference". In: *IEEE Transactions on Signal Processing* 1 (2013) (cit. on p. 181).

[170] V. Chandrasekaran, B. Recht, P. A. Parrilo, and A. S. Willsky. "The Convex Geometry of Linear Inverse Problems". In: *Foundations of Computational Mathematics* 6 (2012) (cit. on pp. 182, 183, 202).

[171] V. Chandrasekaran, S. Sanghavi, P. A. Parrilo, and A. S. Willsky. "Rank-Sparsity Incoherence for Matrix Decomposition". In: *SIAM Journal on Optimization* 2 (2011) (cit. on p. 183).

[172] S. Jagabathula and D. Shah. "Inferring Rankings Using Constrained Sensing". In: *IEEE Transactions on Information Theory* 11 (2011) (cit. on p. 183).

[173] Z. Liu and L. Vandenberghe. "Interior-Point Method for Nuclear Norm Approximation with Application to System Identification". In: *SIAM Journal on Matrix Analysis and Applications* 3 (2010) (cit. on p. 184).

[174] M. Landmann and G. D. Galdo. "Efficient antenna description for MIMO channel modelling and estimation". In: *7th European Conference on Wireless Technology, 2004.* 2004 (cit. on pp. 184, 213).

[175] R. Heckel and M. Soltanolkotabi. "Generalized Line Spectral Estimation via Convex Optimization". In: *IEEE Transactions on Information Theory* 99 (2017) (cit. on pp. 185, 189, 202).

[176] G. Tang, B. N. Bhaskar, P. Shah, and B. Recht. "Compressed sensing off the grid". In: *IEEE Transactions on Information Theory* 11 (2013) (cit. on p. 188).

[177] C. Carathéodory and L. Fejér. "Über den zusammenhang der extremen von harmonischen funktionen mit ihren koeffizienten und über den picard-landau'schen satz". In: *Rendiconti del Circolo Matematico di Palermo (1884-1940)* () (cit. on p. 188).

[178] M. Grant and S. Boyd. *CVX: Matlab Software for Disciplined Convex Programming, version 2.1*. `http://cvxr.com/cvx`. 2014 (cit. on pp. 189, 190).

[179] M. Grant and S. Boyd. "Graph implementations for nonsmooth convex programs". In: *Recent Advances in Learning and Control*. Ed. by V. Blondel, S. Boyd, and H. Kimura. Lecture Notes in Control and Information Sciences. `http://stanford.edu/~boyd/graph_dcp.html`. Springer-Verlag Limited, 2008 (cit. on pp. 189, 190).

[180] M. Sánchez-Fernández, V. Jamali, J. Llorca, and A. Tulino. *Gridless Multidimensional Angle of Arrival Estimation for Arbitrary 3D Antenna Arrays*. 2020 (cit. on p. 189).

[181] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers". In: *Foundations and Trends® in Machine Learning* 1 (2011) (cit. on pp. 193, 194).

[182] Y. Li and Y. Chi. "Off-the-Grid Line Spectrum Denoising and Estimation With Multiple Measurement Vectors". In: *IEEE Transactions on Signal Processing* 5 (2016) (cit. on p. 194).

[183] B. N. Bhaskar, G. Tang, and B. Recht. "Atomic Norm Denoising With Applications to Line Spectral Estimation". In: *IEEE Transactions on Signal Processing* 23 (2013) (cit. on pp. 194, 195).

[184] F. Roemer, M. Ibrahim, N. Franke, N. Hadaschik, A. Eidloth, B. Sackenreuter, and G. Del Galdo. "Measurement matrix design for compressed sensing based time delay estimation". In: *2016 24th European Signal Processing Conference (EUSIPCO)*. 2016 (cit. on p. 203).

[185] V. Abolghasemi, S. Ferdowsi, and S. Sanei. "A gradient-based alternating minimization approach for optimization of the measurement matrix in compressive sensing". In: *Signal Processing* 4 (2012) (cit. on p. 203).

[186] G. Li, Z. Zhu, D. Yang, L. Chang, and H. Bai. "On Projection Matrix Optimization for Compressive Sensing Systems". In: *IEEE Transactions on Signal Processing* 11 (2013) (cit. on p. 203).

[187] B. Kılıç, A. Güngör, M. Kalfa, and O. Arıkan. "Adaptive Measurement Matrix Design in Compressed Sensing Based Direction of Arrival Estimation". In: *2020 28th European Signal Processing Conference (EUSIPCO)*. 2021 (cit. on p. 203).

[188] A. Lavrenko, S. Pawar, M. Ibrahim, F. Roemer, G. Del Galdo, and R. Thomae. "Combining Matrix Design for 2D DoA Estimation with Compressive Antenna Arrays". In: *WSA 2018; 22nd International ITG Workshop on Smart Antennas*. 2018 (cit. on pp. 204, 208, 210, 212).

[189] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller. "Efficient BackProp". In: *Neural Networks: Tricks of the Trade: Second Edition*. Ed. by G. Montavon, G. B. Orr, and K.-R. Müller. Springer Berlin Heidelberg, 2012 (cit. on p. 206).

[190] M. Landmann, M. Kaske, and R. S. Thoma. "Impact of Incomplete and Inaccurate Data Models on High Resolution Parameter Estimation in Multidimensional Channel Sounding". In: *IEEE Transactions on Antennas and Propagation* 2 (2012) (cit. on p. 222).

[191] W. Roth, G. Schindler, H. Fröning, and F. Pernkopf. "Training Discrete-Valued Neural Networks with Sign Activations Using Weight Distributions". In: *Machine Learning and Knowledge Discovery in Databases*. Ed. by U. Brefeld, E. Fromont, A. Hotho, A. Knobbe, M. Maathuis, and C. Robardet. Springer International Publishing, 2020 (cit. on p. 231).

[192] A. Cortés, I. Vélez, and F. Sevillano Juan. "Radik r^k FFTs: Matricial Representation and SDC/SDF Pipeline Implementation". In: *IEEE Transactions on Signal Processing* 7 (2009) (cit. on p. 247).

[193] E. W. Dijkstra. "A Note on Two Problems in Connexion with Graphs". In: *Numer. Math.* 1 (1959) (cit. on p. 248).