ulm university universität
uulm

Channel Coding Inspired Contributions to Compressed Sensing

DISSERTATION

zur Erlangung des akademischen Grades eines

DOKTOR-INGENIEURS

(DR.-ING.)

der Fakultät für Ingenieurwissenschaften,
Informatik und Psychologie der Universität Ulm

von

Henning Alexander Zörlein

geboren in Crailsheim

Gutachter:	Prof. Dr.-Ing. Martin Bossert Prof. Dr.-Ing. Norbert Görtz
Amtierende Dekanin:	Prof. Dr. phil. Tina Seufert

Ulm, 28.08.2015

Preface

This dissertation presents recent results of my research activities at the Institute of Communications Engineering (formerly Telecommunications and Applied Information Theory), Ulm University. Parts of the results have been published within the following articles in conference proceedings and scientific journals:

[1] Dejan E. Lazich, Henning Zörlein, and Martin Bossert. Low coherence sensing matrices based on best spherical codes. In *Proc. 9th International ITG Conference on Systems, Communications and Coding (SCC)*, Munich, Germany, January 2013.

[2] Henning Zörlein, Dejan E. Lazich, and Martin Bossert. On the noise-resilience of OMP with BASC-based low coherence sensing matrices. In *Proc. 10th International Conference on Sampling Theory and Applications (SampTA)*, pages 468–471, Bremen, Germany, July 2013.

[3] Henning Zörlein, Akram Faisal, and Martin Bossert. Dictionary adaptation in sparse recovery based on different types of coherence. In *Proc. 2nd International Workshop on Compressed Sensing applied to Radar (CoSeRa)*, Bonn, Germany, September 2013. URL: `http://arxiv.org/abs/1307.3901`.

[4] Henning Zörlein, Thomas Leichtle, and Martin Bossert. Sparsity aware simplex algorithms for sparse recovery. In *Proc. 10th International ITG Conference on Systems, Communication and Coding (SCC)*, Hamburg, Germany, February 2015.

[5] Mostafa Mohamed, Shrief Rizkalla, Henning Zörlein, and Martin Bossert. Deterministic compressed sensing with power decoding for complex Reed–Solomon codes. In *Proc. 10th International ITG Conference on Systems, Communication and Coding (SCC)*, Hamburg, Germany, February 2015.

[6] Henning Zörlein and Martin Bossert. Coherence optimization and best complex antipodal spherical codes, 2015. URL: `http://arxiv.org/abs/1404.5889`.

My work has been supported by the German research council *Deutsche Forschungsgemeinschaft* (DFG) under Grant Bo 867/27-1. This work was performed on the computational resource bwUniCluster funded by the Ministry of Science, Research and Arts and the Universities of the State of Baden-Württemberg, Germany, within the framework program bwHPC.

Acknowledgments

This thesis would not have been possible without the support of many people and I would like to express my gratitude to all of them.

First of all, I would like to thank my supervisor Prof. Dr.-Ing. Martin Bossert, for creating such a perfect working environment within his institute, where he gave me the opportunity to research on compressed sensing for the last four years. He established an ideal atmosphere, which combines freedom, motivation, support, and entertainment. Thank you for this wonderful time and for adding me to your doctoral family.

Furthermore, I would like to express my gratitude to Prof. Dr.-Ing. Norbert Görtz from Technische Universität Wien for reviewing my thesis and for coming the long way along the Danube to Ulm. Further, I would like to thank Prof. Dr.-Ing. Christian Waldschmidt and Prof. Dr.-Ing. Klaus Dietmayer for being part of the colloquium.

Special thanks to my longtime mentor Prof. Dr. Dejan Lazich, who shared his great wisdom and experience with me on many occasions and continuously supported me during my research on compressed sensing.

I would also like to thank all my current and former colleagues at the institute, who shared all joys and pains of being a doctoral candidate, as well as all other staff members, who made the time such enjoyable. Especially, I would like to thank everyone who proofread parts of the thesis which doubtlessly improved the readability.

During the period of this dissertation, several students wrote their Master's theses under my supervision. It was a pleasure to accompany them during this time. By the open discussions within the supervision-meetings they naturally participated to the findings presented in this thesis.

Finally, I would like to thank my friends and especially my whole family for their everlasting support, encouragement, and patience during all these years and particularly during the time of writing.

Henning Alexander Zörlein
Ulm, August 2015

Abstract

This thesis presents three independent contributions to the large and juvenile field of Compressed Sensing (CS). Thereby, a close connection to the mature field of channel coding is established with an interdisciplinary motivation. Both fields are united by their search for a unique sparsest solution to an underdetermined System of Linear Equations (SLE).

Within the first contribution, Sparsity Aware Simplex Algorithms (SASAs) are provided which extend the well-known approach of Basis Pursuit (BP). The common concept of BP resembles a convex ℓ_1-relaxation to the non-deterministic polynomial-time hard sparse reconstruction problem. In case this relaxation does not lead to the sparsest solution, BP fails inevitably. By extending the famous simplex method for linear optimization, the sparsest solution might be found nevertheless. Thereby, the circumstance is exploited that the desired solution is contained in a degenerated vertex with high probability for systems in general position. All proposed SASAs favor such a degenerated solution against the ℓ_1-relaxation and improve thereby the recovery performance, where the variants differ in their trade-off between complexity and potential gain.

The second contribution generalizes a minimal distance maximization approach for real-valued spherical codes to vector spaces over complex numbers. Furthermore, an equivalence relation is introduced by the concept of antipodal spherical codes, which ensures that vector pairs of minimal coherence correspond to those of largest minimum distance for these antipodal codes as it is proven within this thesis. Consequently, this relation can be used to extend the aforementioned distance optimization approach to the problem of coherence minimization for the case of antipodal spherical codes. The resulting Best Antipodal Spherical Code (BASC) search approach obtains vector sets which improve significantly on previous numerical results and are often close to the theoretical limit. Since the coherence is an important and limiting factor in many applications, the proposed BASC search approach is not only relevant for CS, where coherence is commonly used as uniqueness assuring property for sensing matrices. Beyond the direct CS application with respect to optimized sensing matrices, the potential use for adapting a measurement matrix to a given dictionary is also investigated. Thereby, the influence of two different coherence-based adaptation strategies is examined.

The application of Complex-valued Reed–Solomon (CRS) codes as a deterministic CS scheme is described in details within the third contribution. Previous results in this direction have been extended, whereby particular focus has been put on noise resilience, as this is an often criticized weakness of CRS-based approaches. Within this chapter, power decoding methods have been adapted for three types of error locator algorithms: Peterson's, Berlekamp–Massey, and extended Euclidean. Together with two error evaluator algorithms, namely Gorenstein–Zierler and Forney's, the potential application in corresponding deterministic CS schemes has been considered. In order to counter the observed noise sensitivity, an iterative erasure and evaluation algorithm can be used which aims to determine the correct error locations. In contrast to conventional channel codes over finite fields, a low-degree Padé-approximation can be used for complex vector spaces to obtain reliability-like information on the error locations, which is subsequently used by continuity assisted decoding in order to decode even beyond the power decoding radius.

Contents

List of Symbols and Acronyms **xi**

1. Motivation and Overview **1**

2. Principles of Channel Coding **5**
2.1. Linear Block Codes over Finite Fields 6
2.1.1. Finite Fields and Their Discrete Fourier Transform 6
2.1.2. Algebraic Coding with Linear Block Codes 7
2.2. Reed–Solomon Codes 8
2.2.1. Definition and Properties 9
2.2.2. Decoding 11

3. Underdetermined Systems of Linear Equations **17**
3.1. Common Approaches to Underdetermined SLE 18
3.1.1. Minimum Norm Solution Provided by Moore–Penrose Pseudoinverse . . 18
3.1.2. Linear Optimization 19
3.2. Existence of Unique Sparse Solutions 20
3.2.1. Dimensionality of Intersecting Hyperplanes 20
3.2.2. Sparsity Provides Uniqueness 21

4. Principles of Compressed Sensing **23**
4.1. Reconstruction Algorithms 24
4.1.1. Convex Relaxation 24
4.1.2. Iterative Methods 25
4.2. Uniqueness Assuring Properties 27
4.2.1. Coherence 27
4.2.2. Restricted Isometry Property 28
4.3. Sensing Matrices 29
4.3.1. Sensing Matrices for General Reconstruction 29
4.3.2. Deterministic Sensing Matrices for Dedicated Reconstruction 30
4.4. Variants of Compressed Sensing 31
4.4.1. Noisy Scenarios 31
4.4.2. Sparsifying Dictionaries 32
4.5. Summary and Overview 33

5. Sparsity Aware Simplex Algorithms **35**
5.1. Simplex Method 36
5.1.1. Simplex Algorithm 36
5.1.2. Two-Phase Method 38

5.2. Sparse Solutions through Degeneracy 39
5.2.1. Path Oriented Search Variants 40
5.2.2. Closeness of Convex Relaxation 42
5.3. Sparsity Aware Simplex Methods for Compressed Sensing 43
5.3.1. Sparsity Aware Two-Phase Simplex Method 44
5.3.2. Potential of Sparsity Awareness in Compressed Sensing 44
5.4. Summary and Overview 46

6. Coherence Optimization by Best Antipodal Spherical Codes 49
6.1. Coherence of Vector Sets 50
6.1.1. Theory on Coherence Optimization 50
6.1.2. Existing Approaches to Coherence Optimization 54
6.2. Best Antipodal Spherical Codes for Coherence Optimization 55
6.2.1. Equivalence of Coherence and Distance Optimization 56
6.2.2. Search for Best Antipodal Spherical Codes 57
6.3. Applications in Compressed Sensing 68
6.3.1. Sensing Matrix Optimization 68
6.3.2. Measurement Matrix Adaptation 70
6.4. Summary and Overview 73

7. Compressed Sensing Based on Complex-Valued Reed–Solomon Codes 75
7.1. Complex-Valued Reed–Solomon Codes 76
7.1.1. Definition and Properties 76
7.1.2. Connection to Compressed Sensing 77
7.2. Robust Decoding of CRS Codes for Compressed Sensing 79
7.2.1. Basic Error Location and Evaluation in Noisy Environments 80
7.2.2. Verification and Iterative Improvement on Error Locations 86
7.3. Continuity Assisted Decoding beyond the Power Decoding Radius 89
7.3.1. Low-Degree Padé Approximation Provides Reliability-Like Information 89
7.3.2. Exploiting Continuity for CRS-Based Compressed Sensing 93
7.4. Summary and Overview 95

8. Concluding Remarks 99

Appendices 101

Bibliography 125

List of Symbols and Acronyms

Common Notation

$\mathbb{C}$	Field of complex numbers
$\mathbb{K}$	Arbitrary field
$\mathbb{N}$	Field of natural numbers
$\mathbb{N}_0$	Field of natural numbers including zero
$\mathbb{R}$	Field of real numbers
$\mathbb{R}_{\geq 0}$	Field of non-negative real numbers
$\mathbb{K}^m$	All vectors of length m with elements from $\mathbb{K}$
$\mathbb{K}^{m \times n}$	All matrices with m rows and n columns whose elements are from $\mathbb{K}$
$\mathbb{K}[z]$	Set of polynomials with coefficients from $\mathbb{K}$
$\mathcal{W} = \{\ldots\}$	Set consisting of the elements described by $(\ldots)$
$[a, b]$	Interval ranging from a to b (including a and b)
$\emptyset$	An empty set
$a \in \mathbb{K}$	Scalar
$a = a^{\mathrm{R}} + \mathrm{j}a^{\mathrm{I}} \in \mathbb{C}$	Complex number can be decomposed into real part $a^{\mathrm{R}} = \mathrm{Re}(a)$ and imaginary part $a^{\mathrm{I}} = \mathrm{Im}(a)$
$\mathrm{j} = \sqrt{-1}$	Imaginary unit
$\sigma \in \mathbb{R}$	Standard deviation
$\mathrm{e} = 2.718281828459045235\ldots$	Euler's number
$\pi = 3.14159265358979323\ldots$	Pi
$\boldsymbol{a}(z) = \sum_{i=0}^{n-1} a_i z^i$	Polynomial
$\boldsymbol{a} = (a_0, a_1, \ldots, a_{n-1})^{\mathrm{T}} \in \mathbb{K}^n$	Vector of length n
$\boldsymbol{a}_{\mathcal{W}} \in \mathbb{K}^{\#\mathcal{W}}$	Vector consisting out of the elements of $\boldsymbol{a}$ whose indices correspond to the set $\mathcal{W}$
$\boldsymbol{A} \equiv \mathscr{F}[\boldsymbol{a}]$	Discrete Fourier Transform (DFT) of $\boldsymbol{a}$
$\boldsymbol{W} \in \mathbb{K}^{m \times n}$	Matrix with m rows of length n
$\boldsymbol{W}_{\mathcal{W}} \in \mathbb{K}^{m \times \#\mathcal{W}}$	Matrix consisting out of the columns of $\boldsymbol{W}$ whose indices correspond to the set $\mathcal{W}$
$\boldsymbol{w}_i \in \mathbb{K}^m$	The i-th column of $\boldsymbol{W}$
$W_{j,i} \in \mathbb{K}$	Element in the j-th row and i-th column of $\boldsymbol{W}$

$\boldsymbol{D} \in \mathbb{K}^{n \times n}$	DFT matrix
$\boldsymbol{I} \in \mathbb{K}^{n \times n}$	Identity matrix
$\boldsymbol{U} \in \mathbb{K}^{m \times m}$	Left unitary square matrix from Singular Value Decomposition (SVD)
$\boldsymbol{V} \in \mathbb{K}^{n \times n}$	Right unitary square matrix from SVD
$\boldsymbol{\Sigma} \in \mathbb{R}_{\geq 0}^{m \times n}$	Rectangular matrix from SVD with singular values $\Sigma_{i,i}$ on the diagonal
$\lvert \cdot \rvert$	Absolute value of $(\cdot)$
$\lVert \cdot \rVert_{\ell_p}$	The ℓ_p-norm of $(\cdot)$ according to (4.3) on page 24
$\widehat{(\cdot)}$	An estimation on $(\cdot)$
$\widetilde{(\cdot)}$	An intermediate result on $(\cdot)$
$(\cdot)^{-1}$	Inverse of $(\cdot)$
$(\cdot)^{\dagger}$	Moore–Penrose pseudoinverse of $(\cdot)$
$(\cdot)^{\mathrm{T}}$	Transpose of $(\cdot)$
$(\cdot)^{\mathrm{H}}$	Hermitian conjugate of $(\cdot)$
$\#\mathcal{W}$	Cardinality of the set $\mathcal{W}$
$\mathscr{F}\left[\boldsymbol{a}\right] \equiv \boldsymbol{A}$	DFT of $\boldsymbol{a}$
$\deg \boldsymbol{a}(z) \in \mathbb{N}$	Degree of the polynomial $\boldsymbol{a}(z)$
$\det \boldsymbol{W} \in \mathbb{R}$	Determinant of the matrix $\boldsymbol{W}$
$\ker \boldsymbol{W}$	Null space of the matrix $\boldsymbol{W}$
$\max \mathcal{W}$	Maximum among the elements in the set $\mathcal{W}$
$\min \mathcal{W}$	Minimum among the elements in the set $\mathcal{W}$
$\mathcal{O}(\boldsymbol{f}(z))$	Big O notation: growth is asymptotically limited to that of $\boldsymbol{f}(z)$
$\operatorname{rank} \boldsymbol{W} \in \mathbb{N}$	Rank of the matrix $\boldsymbol{W}$
$\mathrm{Re}(a) = a^{\mathrm{R}} \in \mathbb{R}$	The real part of $a = a^{\mathrm{R}} + \mathrm{j}a^{\mathrm{I}}$
$\mathrm{Im}(a) = a^{\mathrm{I}} \in \mathbb{R}$	The imaginary part of $a = a^{\mathrm{R}} + \mathrm{j}a^{\mathrm{I}}$
$\boldsymbol{a}'(z)$	Formal derivative of $\boldsymbol{a}(z)$
$\langle \boldsymbol{a}, \boldsymbol{b} \rangle = \boldsymbol{a}^{\mathrm{H}} \boldsymbol{b} \in \mathbb{C}$	Inner product between $\boldsymbol{a}$ and $\boldsymbol{b}$
$\lVert \boldsymbol{a} \rVert = \sqrt{\langle \boldsymbol{a}, \boldsymbol{a} \rangle} \in \mathbb{R}_{\geq 0}$	The norm of $\boldsymbol{a}$
$[u/w]_{\boldsymbol{f}}(z) \in \mathbb{K}[z]$	Padé approximation w.r.t. $\boldsymbol{f}(z)$ of order $[u/w]$

Notation for Finite Fields

$\mathbb{F}_q$	Finite field of order q
$\mathbb{F}_{q^m}$	Extension field of $\mathbb{F}_q$ of degree m
${}^{\mathbb{F}_q[z]}\!/_{(z^n-1)}$	Ring of all polynomials over $\mathbb{F}_q$ of degree $< n$

$\alpha \in \mathbb{F}_q$	Primitive element
$p \in \mathbb{N}$	Prime
$q = p^i \in \mathbb{N}$	Power of a prime

Channel Coding Based Notation

$\mathcal{C} \subset \mathbb{K}^n$	(Linear) Block code
$\mathcal{C}^\perp \subset \mathbb{K}^n$	Dual code to $\mathcal{C}$
$\mathcal{IRS}$	Interleaved Reed–Solomon code
$\mathcal{RS}$	Reed–Solomon code
$\mathcal{E}$	Set of erasures
$\mathcal{U} = \{u_1, u_2 \ldots, u_\tau\}$	Set of error positions
$d_{\min} \in \mathbb{N}$	Minimum (Hamming) distance of $\mathcal{C}$
$k \in \mathbb{N}$	Dimension of $\mathcal{C}$
$l_{\max} \in \mathbb{N}$	Maximum integer fulfilling the inequality (2.10)
$n \in \mathbb{N}$	Length of $\mathcal{C}$
$r \in \mathbb{R}$	Coderate of $\mathcal{C}$
$\Delta_t \in \mathbb{K}$	Discrepancy for a sequence of length t in Berlekamp–Massey (BM) type algorithms
$\nu \in \mathbb{N}$	Assumed number of errors
$\tau \in \mathbb{N}$	Number of errors
$\tau_{\max} \in \mathbb{N}$	Maximal number of (almost sure) unique decoding in power decoding
$\tau_{\mathrm{m}} = \tau_{\max}^{(l_{\max})} \in \mathbb{N}$	Maximal number of (almost sure) uniquely decodable errors with power decoding
$\boldsymbol{c} \in \mathcal{C}$	Codeword
$\boldsymbol{e} \in \mathbb{K}^n$	Error vector
$\boldsymbol{i} \in \mathbb{K}^k$	Information vector
$\boldsymbol{r} \in \mathbb{K}^n$	Received vector
$\boldsymbol{s} \in \mathbb{K}^{(n-k)}$	Syndrome
$\boldsymbol{T}$	Vector based on $\boldsymbol{s}$ according to Equations (2.14) and (2.17) on page 11 and on page 12
$\boldsymbol{T}_\tau$	Vector with the first τ entries of $\boldsymbol{T}$ used by Peterson type algorithm

$\boldsymbol{\Lambda}(z) = 1 + \Lambda_1 z + \cdots + \Lambda_\tau z^\tau$	Error locator polynomial
$\boldsymbol{\lambda}(z) = \mathscr{F}^{-1}[\boldsymbol{\Lambda}(z)]$	Inverse Discrete Fourier Transform (IDFT) of error locator polynomial $\boldsymbol{\Lambda}(z)$
$\boldsymbol{\Omega}(z)$	Error evaluator polynomial
$\boldsymbol{\Lambda} = (\Lambda_\tau, \ldots, \Lambda_1)^{\mathrm{T}}$	Vector based on $\boldsymbol{\Lambda}(z)$ according to Equations (2.14) and (2.17) on page 11 and on page 12
$\boldsymbol{G} \in \mathbb{K}^{n \times k}$	Generator matrix of $\mathcal{C}$
$\boldsymbol{H} \in \mathbb{K}^{n \times n-k}$	Parity check matrix of $\mathcal{C}$ and generator matrix of the dual code $\mathcal{C}^\perp$
$\boldsymbol{S}$	Matrix based on $\boldsymbol{s}$ according to Equations (2.14) and (2.17) on page 11 and on page 12
$\boldsymbol{M}_\nu$	Matrix based on $\boldsymbol{S}$ used by Peterson type algorithm
$\boldsymbol{v}[z] \in \mathbb{K}[z]^{l_{\max}+1}$	Vector of polynomials
$\boldsymbol{M}[z] \in \mathbb{K}[z]^{(l_{\max}+1)\times(l_{\max}+1)}$	Matrix of polynomials based on (2.24) on page 15
$\mathcal{M}$	Solution space corresponding to the row-space of the basis $\boldsymbol{M}[z]$
$\deg \boldsymbol{v}[z] \in \mathbb{N}$	Degree of $\deg \boldsymbol{v}[z]$
$\mathrm{LP}\,\boldsymbol{v}[z] \in \mathbb{N}$	Leading position of $\boldsymbol{v}[z]$
$\mathrm{LT}\,\boldsymbol{v}[z] \in \mathbb{K}[z]$	Leading term of $\boldsymbol{v}[z]$
$\mathrm{w}(\boldsymbol{e}) \in \mathbb{N}$	Hamming weight of a vector $\boldsymbol{e}$
$(\cdot)^{(l)}$	Dependency on the current level l of powering
$(\cdot)^{\langle l \rangle}$	Component-wise l-powered vector or polynomial

Notation for SLEs

τ-$\mathcal{CS}$	A τ-dimensional Cartesian subspace
$\mathcal{H}_i$	The i-th hyperplane
$\mathcal{S}_i$	The i-th subspace
$m \in \mathbb{N}$	Number of equations for the System of Linear Equations (SLE)
$n \in \mathbb{N}$	Number of unknowns for the SLE
$\boldsymbol{b} \in \mathbb{K}^m$	Vector describing the constant terms of an SLE
$\boldsymbol{c} \in \mathbb{K}^n$	Objective function of an Linear Program (LP)
$\boldsymbol{x} \in \mathbb{K}^n$	Unknown vector of an SLE

$\boldsymbol{A} \in \mathbb{K}^{m \times n}$	Matrix describing the coefficients of an SLE
$\dim \mathcal{S}_i$	Dimension of the subspace $\mathcal{S}_i$
$\bigcap_{i=1}^{m} \mathcal{S}_i$	Intersection of the subspaces $\mathcal{S}_1, \ldots, \mathcal{S}_m$

Compressed Sensing Specific Notation

$l \in \mathbb{N}$	Length of measured vector
$m \in \mathbb{N}$	Number of measurements
$n \in \mathbb{N}$	Length of sparse vector
$\sigma_{\boldsymbol{\eta}} \in \mathbb{R}$	Standard deviation of the noise $\boldsymbol{\eta}$
$\boldsymbol{\beta} \in \mathbb{K}^m$	Measured vector
$\boldsymbol{\mu} \in \mathbb{K}^m$	Measured vector affected by noise
$\boldsymbol{\eta} \in \mathbb{K}^m$	Noise vector
$\boldsymbol{\chi} \in \mathbb{K}^n$	Sparse vector
$\boldsymbol{\zeta} \in \mathbb{K}^l$	Non-sparse vector
$\boldsymbol{\phi}_i \in \mathbb{K}^l$	The i-th row of the measurement matrix $\boldsymbol{\Phi}$
$\boldsymbol{\psi}_i \in \mathbb{K}^l$	The i-th column of the dictionary $\boldsymbol{\Psi}$
$\boldsymbol{\Theta} \in \mathbb{K}^{m \times n}$	Sensing matrix
$\boldsymbol{\Phi} \in \mathbb{K}^{m \times l}$	Measurement matrix
$\boldsymbol{\Psi} \in \mathbb{K}^{l \times n}$	Dictionary
$\mathrm{M}(\boldsymbol{\Theta}) \in \mathbb{R}$	Coherence of the sensing matrix $\boldsymbol{\Theta}$
$\mathrm{M}(\boldsymbol{\Phi}, \boldsymbol{\Psi}) \in \mathbb{R}$	Coherence between measuring matrix $\boldsymbol{\Phi}$ and dictionary $\boldsymbol{\Psi}$

Notation for Sparsity Aware Simplex Algorithms

$a_{i,j} \in \mathbb{R}$	Elements in the simplex tableau initially corresponding to the coefficients in $\boldsymbol{A}$,cf. (5.3)
$b_i \in \mathbb{R}$	Elements in the simplex tableau initially corresponding to the elements in $\boldsymbol{b}$,cf. (5.3)
$c \in \mathbb{N}$	Selected column
$d \in \mathbb{N}$	Distance between sparse and Basis Pursuit (BP) solution
$k \in \mathbb{N}$	Number of equations for the simplex tableau
$l \in \mathbb{N}$	Number of unknowns for the simplex tableau
$m \in \mathbb{N}$	Number of equations for the LP
$n \in \mathbb{N}$	Number of unknowns for the LP
$r \in \mathbb{N}$	Selected row
$\alpha \in \mathbb{R}$	Element in the simplex tableau corresponding to the value of the objective function,cf. (5.3)
$\alpha_i \in \mathbb{R}$	Elements in the simplex tableau initially corresponding to the objective function,cf. (5.3)
$\Sigma_0 \in \mathbb{R}$	Element for the secondary objective function within the two-phase method, cf. (5.5)
$\boldsymbol{b} \in \mathbb{R}^m$	Vector describing the constraints of an LP
$\boldsymbol{c} \in \mathbb{R}^n$	Objective function of an LP
$\boldsymbol{p} \in \mathbb{R}^k$	Vertex of the solution space
$\boldsymbol{x} \in \mathbb{R}^n_{\geq 0}$	Unknown vector of an LP
$\widetilde{\boldsymbol{x}} \in \mathbb{R}^n_{\geq 0}$	Slack variables for an LP in inequality form
$\boldsymbol{\Sigma} \in \mathbb{R}^n$	Elements for the secondary objective function within the two-phase method, cf. (5.5)
$\boldsymbol{A} \in \mathbb{R}^{m \times n}$	Matrix describing the constraints of an LP

Coherence and BSC Based Notation

$C_{\mathrm{s}}(m,n) = \{\boldsymbol{s}_p\}_{p=1}^n$	Spherical code: n points placed on Ω_m
Ω_m	The m-dimensional unit sphere
C_{bs}	Best spherical code
C_{bas}	Best antipodal spherical code
C_{bcas}	Best complex antipodal spherical code
C_{cas}	complex-valued antipodal spherical code
C_{ras}	real-valued antipodal spherical code

$\mathcal{G}(m,l)$	The Grassmannian space is the set of all l-dimensional subspaces of $\mathbb{K}^m$
$\mathcal{L} = \{\lambda_p\}_{p=1}^n$	Set of Lagrangian multipliers
$\mathcal{W} = \{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_n\}$	Set of n vectors
$a \in \mathbb{R}$	Frame bound
$b \in \mathbb{R}$	Frame bound
$K \in \mathbb{N}$	Number of approximation points
$l \in \mathbb{N}$	Dimensionality of a subspace
$m \in \mathbb{N}$	Dimensionality
$n \in \mathbb{N}$	Number of vectors
$s_{p_u} \in \mathbb{R}$	The u-th element in $\boldsymbol{s}_p$
$\alpha \in \mathbb{R}$	Damping factor
$\nu \in \mathbb{N}$	Free parameter in $\mathrm{g}(\cdot)$
$\lambda_p \in \mathbb{R}$	The p-th Lagrangian multiplier
$\boldsymbol{f}_p \in \mathbb{K}^m$	Superposition of all effective forces acting on $\boldsymbol{s}_p$
$\boldsymbol{\delta}_{pq} \in \mathbb{K}^m$	Effective force acting between $\boldsymbol{s}_p$ and $\boldsymbol{s}_q$
$\boldsymbol{s}_p \in \mathbb{K}^m$	The p-th codeword of $C_\mathrm{s}(m,n)$
$\boldsymbol{C} \in \mathbb{K}^{l \times l}$	Conference matrix
$\boldsymbol{G} \in \mathbb{K}^{n \times n}$	Gram matrix
$\boldsymbol{W} \in \mathbb{K}^{m \times n}$	Matrix with n vectors m dimensions
$\boldsymbol{P}(C_\mathrm{s}(m,n))$	Mapping based on the forces $\boldsymbol{f}_p$
$f(x)$	An abbreviated function, cf. page 59
$\mathrm{FP}(\{\boldsymbol{w}_i\}_1^n)$	Frame potential of a vector set
$\mathrm{g}(\cdot) \in \mathbb{R}$	Generalized potential function
$\mathrm{G}(\cdot) \in \mathbb{R}$	Lagrange function
$\mathrm{M}(\boldsymbol{W})$	Coherence of the vector set described by $\boldsymbol{W}$
$\delta_{pq} \in \mathbb{R}$	Norm of a vector difference between $\boldsymbol{s}_p$ and $\boldsymbol{s}_q$
$\phi(\cdot) \in \mathbb{R}$	An abbreviated function, cf. page 59
$(\cdot)^{(l)}$	State after the l-th iteration

CRS Specific Notation

$\mathcal{CRS}$	Complex-valued Reed–Solomon code
$\mathcal{ICRS}$	Interleaved Complex-valued RS code
$\mathcal{P}$	Set of positions corresponding to the smallest values of $\lvert\widehat{\lambda}_i\rvert$ s.t. $\mathcal{U} \in \mathcal{P}$
$\alpha = \mathrm{e}^{-\mathrm{j}2\pi/n} \in \mathbb{C}$	Element of order n

Acronyms

AMP	Approximate Message Passing
AWGN	Additive White Gaussian Noise
BASC	Best Antipodal Spherical Code
BCASC	Best Complex Antipodal Spherical Code
BCH	Bose–Chaudhuri–Hocquenghem
BCSC	Best Complex Spherical Code
BD	Blu-ray Disc
BM	Berlekamp–Massey
BMD	Bounded Minimum Distance
BP	Basis Pursuit
BPDN	Basis Pursuit DeNoising
BSC	Best Spherical Code
CAD	Continuity Assisted Decoding
CD	Compact Disc
CDMA	Code Division Multiple Access
CoSaMP	Compressive Sampling Matching Pursuit
CRS	Complex-valued Reed–Solomon
CS	Compressed Sensing
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
DRIP	Dictionary-adapted RIP
DSL	Digital Subscriber Line
DVB	Digital Video Broadcast
DVD	Digital Versatile Disc
EE	Extended Euclidean
ETF	Equiangular Tight Frame
GMD	Generalized Minimum-Distance
GRIP	Generalized RIP
GS	Guruswami–Sudan
GZ	Gorenstein–Zierler
ICRS	Interleaved Complex-valued Reed–Solomon
IDFT	Inverse Discrete Fourier Transform
IEE	Iterative Erasure and Evaluation
IHT	Iterative Hard Thresholding
IRS	Interleaved Reed–Solomon
LASSO	Least Absolute Shrinkage and Selection Operator

LDPC	Low-Density Parity-Check
LFSR	Linear Feedback Shift-Register
LP	Linear Program
MDS	Maximum Distance Separable
MIMO	Multiple-Input Multiple-Output
ML	Maximum Likelihood
MP	Matching Pursuit
MS	Mulders–Storjohann
MSE	Mean Squared Error
MUBs	Mutually Unbiased Bases
MWBE	Maximum Welch Bound Equality
NP	Non-deterministic Polynomial-time
NSP	Null Space Property
OFDM	Orthogonal Frequency-Division Multiplexing
OMP	Orthogonal Matching Pursuit
RIC	Restricted Isometry Constant
RIP	Restricted Isometry Property
RM	Reed–Muller
RMS	Root Mean Square
ROC	Restricted Orthogonality Constant
ROMP	Regularized Orthogonal Matching Pursuit
RS	Reed–Solomon
SASA	Sparsity Aware Simplex Algorithm
SLE	System of Linear Equations
SS	Schmidt–Sidorenko
StOMP	Stagewise Orthogonal Matching Pursuit
StRIP	Statistical RIP
SVD	Singular Value Decomposition
UNTF	Unit Norm Tight Frame
WBE	Welch Bound Equality

1

Motivation and Overview

Due to the plethora of research fields, similar solution approaches to common problems are often developed independently by different scientific groups. Therefore, interdisciplinary research approaches are of immediate importance in order to accelerate scientific progress by advantageous synergy effects. An example for one common research problem is the search for the sparsest (least number of non-zeros) solution to an underdetermined System of Linear Equations (SLE). Within the mature field of channel coding, such a system is often used to describe syndrome decoding and its sparsest (minimum Hamming weight) solution is likely to correspond to some unwanted error which needs to be identified for correction. The juvenile field of Compressed Sensing (CS), which is sometimes also denoted as compressive sampling, focuses also on such a sparse recovery problem. Consequently, there is natural interest in transferring knowledge from one field to another. In this spirit, the DFG (German Research Foundation) funded the project *"Methods of Channel Coding for Compressed Sensing"*[1], which provides the theme of this dissertation.

The first mentioned field of channel coding is based on the work of Shannon and Hamming [Sha48, Ham50], who laid out the foundations for reliable communication as we know it today. Good channel codes correspond to a high-dimensional embedding of some lower-dimensional information vector such that the added redundancy can be efficiently used for correcting potential errors (decoding). Typical research concentrates on the construction of such codes, theoretical bounds, as well as on efficient decoding algorithms, where commonly all operations are performed on finite fields. Modern, digital life would not be possible without channel coding, since it heavily depends on protecting communication, processing, and storage systems against errors [WB99, Bos99].

The second domain of this interdisciplinary dissertation is the field of CS. With the publications of Candes & Tao and Donoho [CT06, Don06a], both awarded by the IEEE Information Theory Society Paper Award 2008, CS has been established and received considerable attention ever since. Within CS, the sparsest high-dimensional representation which fits to some low-dimension measurement is commonly searched. Thereby, fields of characteristic zero, like complex and real vector spaces, are typically considered. Research considers potential performance bounds, matrix properties and corresponding constructions, efficient and robust recovery algorithms, as well as additional constraints introduced by specific practical aspects. Potential applications are not limited to magnetic resonance imaging [HHL11], radar [HS09, End10], or the single-pixel camera [DDT+08].

[1] Original title: *Methoden der Kanalcodierung für Compressed Sensing*, Bo 867/27-1.

Due to the strong connections between these research fields, it is not further surprising that several researchers proposed channel coding based methods for CS. Typically, such contributions focus on the construction of (deterministic) sensing matrices as well as corresponding dedicated recovery algorithms: e.g., [DeV07, PH08, HCS08, AHSC09, CHJ10a, AMM12].

The aforementioned DFG project resulted in three independent contributions to the general field of CS, which are presented in details within this thesis. The corresponding structure of the remaining chapters is as follows:

Chapter 2 introduces briefly the necessary foundations of channel coding. Thereby, the focus is on the principles of linear block codes over finite fields, where the relevant terms are introduced as well as the Discrete Fourier Transform (DFT) over finite fields. Subsequently, the concept of algebraic syndrome decoding is generally introduced. The well-known Reed–Solomon (RS) codes are defined with the help of the previously introduced DFT over finite fields. Based on the concept of interleaved RS codes, virtual (syndrome) extension (also commonly known as power decoding) is introduced for three types of error locator algorithms: Peterson's, Berlekamp–Massey, and extended Euclidean. Furthermore, two error evaluator algorithms, namely Gorenstein–Zierler and Forney's, are introduced which conclude this overview chapter.

Subsequently in **Chapter 3**, underdetermined SLE are discussed. In a first step, two common non-sparse approaches are introduced: A minimum norm solution can be obtained by the Moore–Penrose pseudoinverse. Linear optimization identifies an optimal solution based on a given (linear) objective function. In the second half of the chapter, sparsity is introduced as uniqueness providing criterion. Thereby, the solution space is described by intersecting hyperplanes and its dimensionality is determined for the typical case of systems in general position. Based on the corresponding observations, a necessary condition for the uniqueness (with high probability) of sufficiently sparse solutions is introduced. Furthermore, a common necessary and sufficient guarantee for unique sparse solutions is provided, which is based on the null space. This bound corresponds to decoding until half the minimum distance as it is known in channel coding. By this chapter, the foundation for CS is built with respect to the existence of unique sparse solutions. Furthermore, this chapter also forges the link to channel coding.

Chapter 4 introduces several principles of CS. The first focus is on feasible reconstruction algorithms which intend to recover the unique sparsest solution. Thereby, convex relaxation, with the prominent example of an ℓ_1-minimization in case of Basis Pursuit (BP), is introduced as a reconstruction type as well as iterative methods, where orthogonal matching pursuit is particularly highlighted. Both approaches are subsequently extended or used as reference. Afterwards, the coherence and the well-known restricted isometry property are introduced as uniqueness assuring properties and their interconnections are summarized. In contrast to the conditions of the previous chapter, these properties relate to guarantees for specific reconstruction algorithms. Subsequently, suitable sensing matrices are categorized based on their potential recovery approaches. Random, optimized, and deterministic matrix constructions are presented for general reconstruction algorithms. Furthermore, dedicated reconstructions based on certain deterministic matrix constructions are briefly summarized as well. Finally, two common variants of CS are explicitly introduced in order to apply subsequent contributions: Noisy CS scenarios introduce an additional non-sparse noise term which hinders reconstruction. Sparsifying dictionaries allow to extend the theme CS to non-sparse vectors which can be sparsely represented by a linear transform.

Based on these theoretical foundations, three contributions to the general field of CS are given in the subsequent chapters.

In **Chapter 5**, the well-known simplex method for linear optimization is extended for the specific setting of sparse recovery in the context of BP. In a first step, the conventional simplex method is introduced which can be separated into the actual simplex algorithm and the two-phase method. Subsequently, degeneracy is used as an indicator which allows for sparsity awareness. Based on this property, three path-oriented search variants are proposed which can be considered as trade-offs between additional complexity and improved reconstruction. Furthermore, closeness of the convex relaxation can be examined and utilized. Based on these concepts, variants for the sparsity aware two-phase method are proposed and examined for their recovery potential in the context of CS. This chapter is mainly based on [ZLB15].

Chapter 6 focuses on the more general problem of coherence optimization which is also of interest for CS since the coherence has been previously introduced as a uniqueness assuring property. In order to provide a reasonable theoretical basis, the coherence optimization problem is formally defined first, and subsequently, corresponding theoretical bounds are given. Furthermore, the problem of coherence optimization is connected to the research fields of frame theory and Grassmannian line packing which underlines its general importance. Additionally, a brief overview of existing (analytical and numerical) optimization approaches is given. Based in this foundation, the general concept of Best Antipodal Spherical Codes (BASCs) is introduced, where the equivalence of coherence and distance optimization is explicitly established and subsequently used for coherence optimization. The success of the proposed search approach is numerically evaluated and discussed. Finally, the potential of the given BASC-based search approach within the field of CS is demonstrated for two CS variants. Thereby, the straightforward application for sensing matrix optimization is considered as well as measurement matrix adaptation, which highlights the universality of the provided approach. This chapter combines and extends the results of [LZB13, ZLB13, ZAB13, ZB15].

The third contribution is described in **Chapter 7**. Thereby, a deterministic CS scheme based on Complex-valued Reed–Solomon (CRS) codes is presented, where noise robustness is explicitly considered. First, CRS codes and their properties are introduced and a connection to CS is established. Based thereon, robust decoding of CRS codes is proposed by adapted RS power decoding algorithms. In noisy scenarios, the reliability of the error locations are decreased. Therefore, an iterative erasure and evaluation scheme is used to improve the reconstruction performance as it is shown by corresponding evaluations. Based thereon, the concept of continuity assisted decoding extends this approach and allows to decode even beyond the power decoding radius with the help of a low-degree Padé-approximation. The reliability-like information obtained by this approximation is unique to fields with continuous norms, and therefore, not possible for conventional channel coding over finite fields. The corresponding approach is subsequently evaluated for its potential use in CS schemes. Within this chapter, the results of [MRZB15] have been extended.

Finally, a conclusion is given in **Chapter 8**.

2

Principles of Channel Coding

Claude Elwood Shannon laid the groundwork for modern digital communication, and therefore also for channel coding, with his seminal article in 1948 [Sha48]. Therein, he stated and proved the famous channel coding theorem which introduces the channel capacity as absolute bound on the code rate for which digital communication is nearly error-free possible for any noise level. However, the provided proof for the channel coding theorem is not constructive, and therefore, there is no general code construction provided which actually achieves the channel capacity. As argued by Massey in [Mas84], Shannon established the scientific basis of communications, and thereby a whole new field of science, with his information theory.

Starting with Hamming [Ham50], many researchers proposed new code constructions and discussed their properties in the subsequent decades. Within this development, two code classes evolved: convolutional and block codes. Well known examples of block codes are *Reed–Muller* (RM) [Mul54, Ree54], *Bose–Chaudhuri–Hocquenghem* (BCH) [Hoc59, BRC60], *Reed–Solomon* (RS) [RS60], *Low-Density Parity-Check* (LDPC) [Gal62, MN95], and *Hamming codes* [Ham50]. For the remainder of this chapter, RS codes over finite fields and their decoding algorithms are primarily considered. Besides the code construction, corresponding decoding algorithms pose an important and demanding research field. The needed computational complexity became a crucial property of a decoding algorithm. Conventional approaches are limited to a maximal error correcting radius of half the minimum Hamming distance. Recently, the potential to decode beyond the maximal error correcting radius by accepting multiple results or decoding failures (with small probability) has became an active field of research, e.g., [GS99, SSB10].

Since the focus of this thesis is on *Compressed Sensing* (CS), this chapter on channel coding is kept brief and compact. Within this chapter, the necessary notation is established and properties and algorithms are introduced for the later use in subsequent chapters. For a broader introduction to channel coding, the reader should refer to existing textbooks and overview articles, e.g., [MS88, Bos99, Bla03, JH04, Moo05, Rot06, BB13]. In Section 2.1, the fundamental concept of finite fields is provided and the basic notation is established. Additionally, the *Discrete Fourier Transform* (DFT) for finite fields is introduced which is subsequently used in Section 2.2 for the definition of RS codes. Based thereon, *Interleaved Reed–Solomon* (IRS) codes are defined and the concept of virtual extension is given which allows power decoding. Afterwards, (power) decoding algorithms for RS codes are provided as well. Since the complex-valued equivalents of RS codes are introduced in Chapter 7 on

page 75 for the application in CS, the definitions of this chapter are used as reference for the later discussion of *Complex-valued Reed–Solomon* (CRS) codes and their decoding algorithms.

2.1. Linear Block Codes over Finite Fields

The theory of finite fields goes back to Évariste Galois, a famous French mathematician of the 19th century [Gal30, Rot82]. To honor him, the term Galois field is also commonly used. In Section 2.1.1, the subsequently used notation for finite fields and their DFT is established. Afterwards, the basics of algebraic coding with linear block codes are presented in Section 2.1.2 on the facing page.

2.1.1. Finite Fields and Their Discrete Fourier Transform

This section provides the notation of finite fields and their DFT for the subsequent use in this thesis. Deeper details about the corresponding theory, as well as possible applications, can be found in standard literature, e.g., [LN97, MP13]. Channel coding related books provide also profound introductions, e.g., [Bos99, Bla03, Rot06].

The finite field of order q is denoted by $\mathbb{F}_q$, where $q = p^n$ is the n-th power of some prime p. Each finite field $\mathbb{F}_q$ contains subfields $\mathbb{F}_{p^m}$ for every m dividing n. In case of $m = 1$, the subfield $\mathbb{F}_p$ is denoted as prime (sub)field or basefield. The finite field $\mathbb{F}_q$ contains q elements and its characteristic is equal to p.[1] The field $\mathbb{F}_q$ can be further extended to the extension field $\mathbb{F}_{q^m}$, where m is the degree of the field extension.

The primitive element of $\mathbb{F}_q$ is denoted by α and generates the multiplicative group $\mathbb{F}_q^*$ by its powers: $\mathbb{F}_q^* = \left\{\alpha^i \;\forall\; i \in [0, q-2]\right\}$. Since $\mathbb{F}_q^*$ consists of the non-zero elements of $\mathbb{F}_q$, α can be used to describe the finite field:

$$\mathbb{F}_q = \{0\} \cup \mathbb{F}_q^* = \left\{0, \alpha^i \;\forall\; i \in [0, q-2]\right\} \tag{2.1}$$

The set of polynomials with coefficients from $\mathbb{F}_q$ is denoted by $\mathbb{F}_q[z]$. A polynomial $\boldsymbol{a}(z) \in \mathbb{F}_q[z]$ is commonly expressed by

$$\boldsymbol{a}(z) = \sum_{i=0}^{n-1} a_i z^i = a_0 + a_1 z + \ldots + a_{n-1} z^{n-1}, \quad a_i \in \mathbb{F}_q \;\forall\; i \in [0, n-2]\,,\; a_{n-1} \in \mathbb{F}_q^*, \tag{2.2}$$

where the degree of $\boldsymbol{a}(z)$ is $\deg \boldsymbol{a}(z) = n - 1$. Polynomial roots are given by $\boldsymbol{a}(z) = 0$. An irreducible polynomial over $\mathbb{F}_q$ cannot be written as product of polynomials over $\mathbb{F}_q$ with lower degree. The ring of all polynomials of degree smaller than n over $\mathbb{F}_q$ is denoted by $\mathbb{F}_q[z]/(z^n-1)$.

In [Pol71], the DFT over $\mathbb{F}_q$ is proposed, which is briefly introduced in the following. The set of all n-tuples with elements from $\mathbb{F}_q$ is denoted by $\mathbb{F}_q^n$. Subsequently, vectors are used to represent such tuples. For a vector $\boldsymbol{a} = (a_0, a_1, \ldots, a_{n-1})^{\mathrm{T}} \in \mathbb{F}_q^n$, the DFT $\boldsymbol{A} = \mathscr{F}[\boldsymbol{a}] \in \mathbb{F}_q^n$ is calculated elementwise by

$$A_i = \sum_{l=0}^{n-1} a_l \alpha^{li} \quad \forall\; i \in [0, n-1]\,. \tag{2.3}$$

[1] In comparison, the fields $\mathbb{R}$ and $\mathbb{C}$ are of characteristic zero.

The vectors $\boldsymbol{a}$ and $\boldsymbol{A}$ are denoted as time and frequency domain vector, respectively. With the help of polynomials, it is possible to represent the DFT efficiently as evaluation of the polynomial $\boldsymbol{a}(z)$ at the points α^i for $i \in [0, n-1]$, where the coefficients of $\boldsymbol{a}(z)$ are taken from $\boldsymbol{a}$ as described by (2.2):

$$A_i = \boldsymbol{a}\left(\alpha^i\right) \quad \forall\, i \in [0, n-1] \tag{2.4}$$

Therefore, the polynomials $\boldsymbol{a}(z)$ and $\boldsymbol{A}(z)$ are denoted as time and frequency domain polynomials, respectively. The *Inverse Discrete Fourier Transform* (IDFT) $\boldsymbol{a} = \mathscr{F}^{-1}[\boldsymbol{A}]$ over $\mathbb{F}_q$ can be similarly defined by

$$a_i = \frac{1}{n}\sum_{l=0}^{n-1} A_l \alpha^{-li} = \frac{1}{n}\boldsymbol{A}\left(\alpha^{-i}\right) \quad \forall\, i \in [0, n-1]\,. \tag{2.5}$$

For the later discussion, it is more convenient to express the DFT with the means of vector matrix multiplication: $\mathscr{F}[\boldsymbol{a}] :\ \boldsymbol{A} = \boldsymbol{D}\boldsymbol{a}$, where

$$\boldsymbol{D} = \begin{pmatrix} \alpha^0 & \alpha^0 & \alpha^0 & \cdots & \alpha^0 \\ \alpha^0 & \alpha^1 & \alpha^2 & \cdots & \alpha^{n-1} \\ \alpha^0 & \alpha^2 & \alpha^4 & \cdots & \alpha^{2(n-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ \alpha^0 & \alpha^{n-1} & \alpha^{2(n-1)} & \cdots & \alpha^{(n-1)(n-1)} \end{pmatrix}$$

is the DFT matrix. Since $\boldsymbol{D}$ has the form of a Vandermonde matrix, and is therefore of full rank (for α being a primitive element), an inverse $\boldsymbol{D}^{-1}$ exists [AL69] with

$$\boldsymbol{D}^{-1} = \frac{1}{n}\begin{pmatrix} \alpha^0 & \alpha^0 & \alpha^0 & \cdots & \alpha^0 \\ \alpha^0 & \alpha^{-1} & \alpha^{-2} & \cdots & \alpha^{-(n-1)} \\ \alpha^0 & \alpha^{-2} & \alpha^{-4} & \cdots & \alpha^{-2(n-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ \alpha^0 & \alpha^{-(n-1)} & \alpha^{-2(n-1)} & \cdots & \alpha^{-(n-1)(n-1)} \end{pmatrix}. \tag{2.6}$$

The scaling factor $1/n$ in Equations (2.5) and (2.6) is necessary to ensure $\boldsymbol{D}^{-1}\boldsymbol{D} = \boldsymbol{I}$. Consequently, the IDFT can be described by $\mathscr{F}^{-1}[\boldsymbol{A}] :\ \boldsymbol{a} = \boldsymbol{D}^{-1}\boldsymbol{A}$.

2.1.2. Algebraic Coding with Linear Block Codes

A linear block code $\mathcal{C}$ is defined as a subspace of $\mathbb{F}_q^n$. A codeword $\boldsymbol{c} \in \mathcal{C}$ is therefore a vector with elements over $\mathbb{F}_q$. The addition of two codewords results also in a codeword. If every cyclic shift of a codeword results in a valid codeword, the code is cyclic. The dimension of $\mathcal{C}$ is denoted by k and the length by n, therefore, $\mathcal{C}$ is also denoted as (n, k) code. The coderate r is given by $r = k/n \operatorname{ld} q$, which simplifies to $r = k/n$ for binary codes.

The Hamming weight $\mathrm{w}(\boldsymbol{c})$ of a vector is equal to the number of non-zero components in the vector. The Hamming weight of the difference between two distinct codewords $\boldsymbol{a}$

and $\boldsymbol{c}$ is denoted as Hamming distance $\mathrm{d}(\boldsymbol{a}, \boldsymbol{c}) = \mathrm{w}(\boldsymbol{a} - \boldsymbol{c})$. The smallest Hamming weight of any non-zero codeword $\boldsymbol{c} \in \mathcal{C} \setminus \{\boldsymbol{0}\}$ is denoted as minimum distance $d_{\min}$ of the linear block code $\mathcal{C}$. The minimum distance of any linear (n, k) code is bounded by the Singleton bound [Kom53, Jos58, Sin64]:

$$d_{\min} \leq n - k + 1 \tag{2.7}$$

A code satisfying the Singleton bound with equality is denoted as *Maximum Distance Separable* (MDS) code.

Since $\mathcal{C}$ is a subspace, it can be completely described as the column-space[2] of an $n \times k$ matrix $\boldsymbol{G}$, where the columns can be any set of basis vectors for $\mathcal{C}$. Such a matrix $\boldsymbol{G}$ is therefore denoted as generator matrix of $\mathcal{C}$. The encoding of an information vector $\boldsymbol{i} \in \mathbb{F}_q^k$ into some codeword $\boldsymbol{c}$ can be efficiently described by $\boldsymbol{c} = \boldsymbol{G}\boldsymbol{i}$. The set of all vectors orthogonal to $\mathcal{C}$ is again a subspace of $\mathbb{F}_q^n$ and called the dual code $\mathcal{C}^\perp$. The $n \times (n-k)$ generator matrix of $\mathcal{C}^\perp$ is denoted by $\boldsymbol{H}$. Since $\boldsymbol{H}^{\mathrm{T}}\boldsymbol{G} = \boldsymbol{0}$, some codeword $\boldsymbol{c} \in \mathcal{C}$ can be tested for validity by verifying $\boldsymbol{H}^{\mathrm{T}}\boldsymbol{c} = \boldsymbol{0}$. Therefore, $\boldsymbol{H}$ is also called the parity check matrix of $\mathcal{C}$. The application of the parity check matrix $\boldsymbol{H}$ to some received vector $\boldsymbol{r} \in \mathbb{F}_q^n$ results in the syndrome $\boldsymbol{s} \in \mathbb{F}_q^{n-k}$ which is given by $\boldsymbol{s} = \boldsymbol{H}^{\mathrm{T}}\boldsymbol{r}$. In case of $\boldsymbol{r} = \boldsymbol{c} + \boldsymbol{e}$, where $\boldsymbol{e} \in \mathbb{F}_q^n$ is some error vector and $\boldsymbol{c} \in \mathcal{C}$, the syndrome depends only on the error: $\boldsymbol{s} = \boldsymbol{H}^{\mathrm{T}}\boldsymbol{r} = \boldsymbol{H}^{\mathrm{T}}\boldsymbol{e}$. Therefore, the syndrome $\boldsymbol{s}$ provides information which can be used to decode the received word $\boldsymbol{r}$.

Assuming a received word $\boldsymbol{r}$ which has been affected by an error vector with $\mathrm{w}(\boldsymbol{e}) = \tau$ errors, the unique decoding of $\boldsymbol{r}$ is possible as long as $\tau \leq \lfloor (d_{\min} - 1)/2 \rfloor$. There are several possible decoding principles: For *Maximum Likelihood* (ML) decoding (also known as nearest neighbor decoding), the codeword $\boldsymbol{c}$ is returned which minimizes the Hamming distance $\mathrm{d}(\boldsymbol{r}, \boldsymbol{c})$ over all codewords. Due to its complexity, the optimal ML decoding is only feasible for small codes. In the case of *Bounded Minimum Distance* (BMD) decoding, the received word is only decoded if $\tau \leq \lfloor (d_{\min} - 1)/2 \rfloor$ errors are detected, elsewise, a failure is declared. More recent approaches aim to decode beyond $\lfloor (d_{\min} - 1)/2 \rfloor$, where the uniqueness of a decoding result cannot be guaranteed. Examples for suboptimal algorithms following this principle are list-decoders like the *Guruswami–Sudan* (GS) algorithm [GS99], where multiple possible codewords are returned, or schemes like power decoding [SSB10], where some small failure probability is accepted.

2.2. Reed–Solomon Codes

The well-known class of RS codes goes back to [Bus52, RS60]. These codes are widely used, e.g., for coding schemes used in deep space communication, f.i. in the Voyager and Galileo program; for digital communication, f.i. *Digital Video Broadcast* (DVB) and *Digital Subscriber Line* (DSL); as well as for storage systems, f.i. *Compact Discs* (CDs) and their successors *Digital Versatile Discs* (DVDs) and *Blu-ray Discs* (BDs) [WB99, Blu12]. In Section 2.2.1, the definition and properties of RS and IRS codes are given. Corresponding (power) decoding algorithms are provided in Section 2.2.2 on page 11.

[2] Opposed to the common notation in coding theory, column vectors are used within this thesis.

2.2.1. Definition and Properties

In the following, RS codes are defined. Subsequently, IRS codes are introduced in order to provide the concept of virtual extension. For the sake of brevity, only the definition of primitive RS codes is given. More general definitions can be found in [Bos99, Bla03]. However, the given variant is sufficient as basis for the later use in Chapter 7.

Reed–Solomon Codes

RS codes are cyclic linear (n, k) codes over the field $\mathbb{F}_q$ with length $n = q - 1$ and can be defined by the IDFT as set of polynomials:

$$\mathcal{RS}(n,k) = \left\{ \boldsymbol{c}(z) \mid c_i = \frac{1}{n} \boldsymbol{C}\left(\alpha^{-i}\right) \quad \forall\, i \in [0, n-1],\ \deg \boldsymbol{C}(z) < k \right\},$$

where $\boldsymbol{C}(z) \in {}^{\mathbb{F}_q[z]}\!/_{(z^n-1)}$ and α is the primitive element of $\mathbb{F}_q$. As before, the coefficients of $\boldsymbol{c}(z)$ can be equivalently represented as vectors, and consequently, a k-dimensional vector subspace of $\mathbb{F}_q^n$ is described by $\mathcal{RS}(n, k)$. By this definition, the generator matrix $\boldsymbol{G}$ equals a partial IDFT matrix

$$\boldsymbol{G} = \frac{1}{n} \begin{pmatrix} \alpha^0 & \alpha^0 & \alpha^0 & \cdots & \alpha^0 \\ \alpha^0 & \alpha^{-1} & \alpha^{-2} & \cdots & \alpha^{-(k-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ \alpha^0 & \alpha^{-(n-1)} & \alpha^{-2(n-1)} & \cdots & \alpha^{-(n-1)(k-1)} \end{pmatrix}.$$

Usually, the parity check matrix $\boldsymbol{H}$ is similarly defined to be a partial DFT matrix:

$$\boldsymbol{H} = \begin{pmatrix} \alpha^0 & \alpha^0 & \alpha^0 & \cdots & \alpha^0 \\ \alpha^k & \alpha^{(k+1)} & \alpha^{(k+2)} & \cdots & \alpha^{n-1} \\ \vdots & \vdots & & & \vdots \\ \alpha^{(n-1)k} & \alpha^{(n-1)(k+1)} & \alpha^{2(k+2)} & \cdots & \alpha^{(n-1)(n-1)} \end{pmatrix}$$

Both matrices still possess the Vandermonde structure and are, therefore, of full rank. RS codes are MDS, since they fulfill the Singleton bound with equality $d_{\min} = n - k + 1$ [Bla03, Theorem 6.2.1], cf. (2.7). As a consequence, up to

$$\tau_{\max} = \left\lfloor \frac{(n-k)}{2} \right\rfloor \tag{2.8}$$

errors can be uniquely decoded by RS codes.

Although BCH codes are not explicitly described here, it should be noted that they are closely connected to RS codes: Over finite fields, RS codes can be obtained from BCH codes by limiting the length. However, BCH codes over $\mathbb{F}_q$ are subfield-subcodes of RS codes over $\mathbb{F}_{q^m}$. For details about BCH codes and their connection to RS codes, refer to [Bos99].

Interleaved Reed–Solomon Codes and Virtual Extension

As the name suggests, an IRS code corresponds to an interleaved structure of several RS codes. Therefore, it can be considered as a set of codeword matrices, where each matrix itself consists out of RS codewords:

$$\mathcal{IRS}(n, k_1, \ldots, k_l) = \{(\boldsymbol{c}_1\ \boldsymbol{c}_2\ \cdots\ \boldsymbol{c}_l) : \boldsymbol{c}_i \in \mathcal{RS}(n, k_i)\}$$

Alternatively, a codeword can be interpreted as vector with elements from the extension field $\mathbb{F}_{q^l}$. With this representation, it can be shown that IRS codes are MDS in the homogeneous case of $k_i = k$ for all i [SSB09]. However, for heterogeneous IRS codes, this is generally not the case since the weakest RS code determines the minimum distance [SSB09]: $d = \min_{1\leq i\leq l} n - k_i + 1$. Consequently, the guaranteed correction radius is limited by

$$\tau \leq \min_{1\leq i\leq l} \left\lfloor \frac{n-k_i}{2} \right\rfloor.$$

For channels which introduce burst errors affecting the same positions in all RS codewords, IRS codes are especially suited [KL97]. Due to the collaboration, the correction radius can be increased to

$$\tau \leq \min\left\{ \frac{l}{(l+1)} \cdot \left(n - \frac{1}{l}\sum_{i=1}^{l} k_i \right), \min_{1\leq i\leq l} n - k_i \right\},$$

with a small failure probability [SSB09].

In [SSB10], a RS decoding scheme has been proposed which can be described as a virtual extension of an RS code to an IRS code. Instead of interleaving multiple codewords from different RS codes, only a single low rate code is used, where the received word $\boldsymbol{r}(z) = \boldsymbol{c}(z) + \boldsymbol{e}(z)$ with $\boldsymbol{c}(z) \in \mathcal{RS}(n, k)$ is powered component-wise with some positive integer l:

$$\begin{aligned}\boldsymbol{r}^{\langle l\rangle}(z) :=& \sum_{i=0}^{n-1} r_i^l z^i = \sum_{i=0}^{n-1} (c_i + e_i)^l z^i = \sum_{i=0}^{n-1} (c_i^l + e_i^{(l)}) z^i \\ =& \boldsymbol{c}^{\langle l\rangle}(z) + \boldsymbol{e}^{(l)}(z)\end{aligned} \tag{2.9}$$

As a consequence of the binomial distribution, $e_i = 0$ results in $e_i^{(l)} = 0$, while the converse is not necessarily true. Therefore, the indices of the non-zero coefficients of $\boldsymbol{e}^{(l)}(z)$ are a subset of those of $\boldsymbol{e}(z)$, and consequently, the errors are at the same positions in all (virtually) received words $\boldsymbol{r}^{\langle l\rangle}(z)$. Powering $\boldsymbol{c}(z)$ component-wise to $\boldsymbol{c}^{\langle l\rangle}(z)$ is equivalent to powering the whole frequency domain polynomial $\boldsymbol{C}(z)$ to $\boldsymbol{C}(z)^l$, due to the definition of the DFT (see (2.4) on page 7). Since $\boldsymbol{C}(z)$ is an RS code with $\deg \boldsymbol{C}(z) \leq k-1$, the relation $\deg \boldsymbol{C}(z)^l \leq l(k-1)$ is implied and results in

$$\boldsymbol{c}^{\langle l\rangle}(z) \in \mathcal{RS}(n, k^{(l)} := l(k-1)+1) \quad \text{with} \quad k^{(l)} \leq n. \tag{2.10}$$

It is possible to build an IRS codeword from a single RS codeword $\boldsymbol{c}(z) \in \mathcal{RS}(n, k)$:

$$(\boldsymbol{c}^{\langle 1\rangle}\ \boldsymbol{c}^{\langle 2\rangle}\ \cdots\ \boldsymbol{c}^{\langle l_{\max}\rangle}) \in \mathcal{IRS}(n, k, k^{(2)}, \ldots, k^{(l_{\max})}),$$

where the maximum l fulfilling the inequality in (2.10) is denoted by $l_{\max}$. Although there is only one RS code used, the resulting IRS code is heterogeneous since the virtual extension implicitly uses different RS codes [cf. (2.10)].

As a consequence, collaborative decoding of the $\mathcal{IRS}(n, k, k^{(2)}, \ldots, k^{(l_{\max})})$ code can be used to improve the decoding capability for the RS code, where an upper bound for $l_{\max}$ is given by [SSB10, Section II-E]:

$$l_{\max} \leq \frac{\sqrt{(k+3)^2 + 8(k-1)(n-1)} - (k+3)}{2(k-1)}$$

Using powers $l \leq l_{\max}$, it can also be shown that, except for a negligible probability of decoding failure, collaborative decoding can correct up to [SSB10, Equation (10)]

$$\tau_{\max}^{(l)} := \left\lfloor \frac{2ln - l(l+1)k + l(l-1)}{2(l+1)} \right\rfloor \leq \tau_{\max}^{(l_{\max})} \tag{2.11}$$

errors. Furthermore, it is shown in [SSB10] that the approach of virtual extensions can only enlarge the decoding radius for a code rate of $r \lesssim 1/3$. Since the scheme is based on powering polynomials, it is often also denoted as power decoding. Algorithms capable of power decoding are introduced in the subsequent section.

2.2.2. Decoding

In the following, syndrome decoding of RS codes is considered, where the information contained in the syndrome $\boldsymbol{s}$ is used to decode the received word $\boldsymbol{r}$ by determining an error estimate $\widehat{\boldsymbol{e}}$. Since the frequency domain codeword polynomial is of limited degree with $\deg \boldsymbol{C}(z) \leq k-1$, the $n-k$ highest coefficients of $\mathscr{F}[\boldsymbol{c}(z) + \boldsymbol{e}(z)]$ are equal to those of the frequency domain error polynomial $\boldsymbol{E}(z)$ and provide the syndrome $\boldsymbol{S}(z)$ with $S_i = E_{i+k}\ \forall\, i \in [0, n-k-1]$.

The error locator polynomial $\boldsymbol{\Lambda}(z)$ is defined such that it has roots α^{-u_i} at all τ error locations $u_i \in \mathcal{U} = \{u_1, u_2 \ldots, u_\tau\}$, where $\mathcal{U}$ is also denoted as support set of $\boldsymbol{e}$:

$$\boldsymbol{\Lambda}(z) = \prod_{u_i \in \mathcal{U}} (1 - \alpha^{u_i} z) = 1 + \Lambda_1 z + \Lambda_2 z^2 + \cdots + \Lambda_\tau z^\tau \tag{2.12}$$

Due to the DFT, the coefficients $\lambda_{u_1}, \lambda_{u_2}, \ldots, \lambda_{u_\tau}$ of the polynomial $\boldsymbol{\lambda}(z) = \mathscr{F}^{-1}[\boldsymbol{\Lambda}(z)] = \lambda_0 + \lambda_1 z + \ldots \lambda_{n-1} z^{n-1}$ are zero. Since the coefficients in $\boldsymbol{\lambda}(z)$ and $\boldsymbol{e}(z)$ are complementary equal to zero, coefficient-wise multiplication yields $\lambda_i e_i = 0\ \forall\, i \in [0, n-1]$ and results in

$$\boldsymbol{\Lambda}(z)\boldsymbol{E}(z) = 0 \mod (z^n - 1). \tag{2.13}$$

A linear system of equations can be derived from (2.13), which includes only equations consisting of coefficients from the error locator polynomial $\boldsymbol{\Lambda}(z)$ or the syndrome, since $S_i = E_{i+k}$:

$$\underbrace{\begin{pmatrix} S_0 & S_1 & \cdots & S_{\tau-1} \\ S_1 & S_2 & \cdots & S_\tau \\ \vdots & \vdots & & \vdots \\ S_{n-k-\tau-1} & S_{n-k-\tau} & \cdots & S_{n-k-2} \end{pmatrix}}_{\boldsymbol{S}} \cdot \underbrace{\begin{pmatrix} \Lambda_\tau \\ \Lambda_{\tau-1} \\ \vdots \\ \Lambda_1 \end{pmatrix}}_{\boldsymbol{\Lambda}} = \underbrace{\begin{pmatrix} -S_\tau \\ -S_{\tau+1} \\ \vdots \\ -S_{n-k-1} \end{pmatrix}}_{\boldsymbol{T}} \tag{2.14}$$

This system of equations can be represented also by polynomials

$$\begin{aligned} \boldsymbol{\Lambda}(z)\boldsymbol{S}(z) &= \boldsymbol{\Omega}(z) \mod z^{n-k}, \\ \deg \boldsymbol{\Omega}(z) &< \deg \boldsymbol{\Lambda}(z) = \tau, \end{aligned} \tag{2.15}$$

where $\boldsymbol{\Omega}(z)$ is denoted as error evaluator polynomial. Often, (2.15) is referred to as key equation, since its solution reveals the decoding result.

Due to the virtual extension performed in power decoding, multiple syndromes can be obtained from the component-wise powered received polynomials $\boldsymbol{r}^{\langle l \rangle}(z)$, cf. (2.9) [SSB10]:

$$S_i^{(l)} = E_{k^{(l)}+i}^{(l)}, \quad i \in \left[0, n-k^{(l)}-1\right] \ \forall\ l \in [1, l_{\max}] \tag{2.16}$$

Therefore, the previously described concept of power decoding can also be considered as syndrome extension. With these syndromes, an extended linear system $\boldsymbol{S}\boldsymbol{\Lambda} = \boldsymbol{T}$ can be established based on (2.14):

$$\underbrace{\begin{pmatrix} \boldsymbol{S}^{(1)} \\ \boldsymbol{S}^{(2)} \\ \vdots \\ \boldsymbol{S}^{(l_{\max})} \end{pmatrix}}_{\boldsymbol{S}} \cdot \underbrace{\begin{pmatrix} \Lambda_\tau \\ \Lambda_{\tau-1} \\ \vdots \\ \Lambda_1 \end{pmatrix}}_{\boldsymbol{\Lambda}} = \underbrace{\begin{pmatrix} \boldsymbol{T}^{(1)} \\ \boldsymbol{T}^{(1)} \\ \vdots \\ \boldsymbol{T}^{(l_{\max})} \end{pmatrix}}_{\boldsymbol{T}}, \tag{2.17}$$

where

$$\boldsymbol{S}^{(l)} = \begin{pmatrix} S_0^{(l)} & S_1^{(l)} & \cdots & S_{\tau-1}^{(l)} \\ S_1^{(l)} & S_2^{(l)} & \cdots & S_\tau^{(l)} \\ \vdots & \vdots & & \vdots \\ S_{n-k^{(l)}-\tau-1}^{(l)} & S_{n-k^{(l)}-\tau}^{(l)} & \cdots & S_{n-k^{(l)}-2}^{(l)} \end{pmatrix}, \quad \boldsymbol{T}^{(l)} = \begin{pmatrix} -S_\tau^{(l)} \\ -S_{\tau+1}^{(l)} \\ \vdots \\ -S_{n-k^{(l)}-1}^{(l)} \end{pmatrix}.$$

As long as $\operatorname{rank} \boldsymbol{S} = \tau$, this system allows to decode beyond half the minimum distance.

In case of available reliability information, soft-decoding schemes, such as the *Koetter–Vardy* algorithm [KV03] or *Generalized Minimum-Distance* (GMD) [FJ66] decoding algorithms, can further increase the decoding capabilities. Erasure decoding combines the concepts of soft and hard decoding. Thereby, a set of erasures $\mathcal{E}$ is defined which marks missing (or potentially unreliable) received word coefficients $r_i,\ i \in \mathcal{E}$. These coefficients are considered erroneous and are, therefore, erased by setting $r_i = 0$. Since their positions are known, they can be considered as fixed factors $(1-\alpha^i z)$ in $\boldsymbol{\Lambda}(z)$ and only their values need to be determined. Thus, the number of unknowns is reduced, and consequently, more errors can be decoded with the available equations (two correct erasures allow to decode one additional error). The number of correctable errors is, therefore, smaller than $(d_{\min}-\#\mathcal{E})/2$ for the case of unique decoding. The concept of erasure decoding is applied, for example, in the GMD decoding algorithm, where the number of erased least reliable positions is iteratively increased. Thereby, a list of potential codewords is obtained.

Usually, decoding is divided into two steps: Error location, where the error positions are determined (and potential erasures considered), followed by error evaluation, where the corresponding values are determined for non-binary codes. Often, error location is also described as Padé-approximation, cf. Section 7.3.1 on page 89. In the following, three types of error location algorithms capable of performing power decoding are provided, where the name of the typification originates from the non-power decoding variants of the algorithms. Subsequently, error evaluation algorithms are given as well. A unified view on decoding algorithms can be found in [BB13].

Peterson Type Algorithm

The *Peterson* algorithm is one of the first RS decoding algorithms and goes back to [Pet60]. However, since the algorithm is not very efficient, it is rarely used in practice.

In a first step, the number of errors τ needs to be determined, which is typically done by exploiting $\operatorname{rank} \boldsymbol{S} = \tau$. For an assumed number of errors $\nu \leq \tau_{\max}^{(l_{\max})}$, the square matrix $\boldsymbol{M}_\nu$ can be built as the first ν rows of $\boldsymbol{S}$ from (2.17) for ν errors. Obviously, $\boldsymbol{M}_\nu$ is singular if too many errors are assumed ($\nu > \tau$). Therefore, by reducing ν stepwise from $\tau_{\max}^{(l_{\max})}$ until $\det \boldsymbol{M}_\nu \neq 0$, the rank of $\boldsymbol{S}$ can be determined. Consequently, the key equation can be solved by applying the inverse: $\boldsymbol{\Lambda} = \boldsymbol{M}_\tau^{-1} \boldsymbol{T}_\tau$, where $\boldsymbol{T}_\tau$ contains the first τ entries of $\boldsymbol{T}$.

Berlekamp–Massey Type Algorithms

The *Berlekamp–Massey* (BM) algorithm has been introduced by Berlekamp in [Ber68]. Subsequently, Massey provided a simplified description which illustrates the problem as synthesis of the shortest *Linear Feedback Shift-Register* (LFSR) capable of generating a prescribed finite sequence [Mas69]. In [SS06], the BM algorithm is generalized to multi-sequence synthesis, such that it can be used in power decoding [SSB10]. The resulting algorithm is also denoted as *Schmidt–Sidorenko* (SS) algorithm, due to their common principle, the term BM type is used to address both algorithms in the following.

Since the submatrices $\boldsymbol{S}^{(l)}$ in (2.17) possess Hankel structure, the corresponding linear system can be represented for $\nu \leq \tau_{\max}^{(l_{\max})}$ assumed errors by

$$S_i^{(l)} + \Lambda_1 S_{i-1}^{(l)} + \ldots + \Lambda_\nu S_{i-\nu}^{(l)} = 0 \quad \forall\, i \in \left[\nu, n - k^{(l)} - 1\right]. \tag{2.18}$$

In the context of LFSR synthesis, the resulting $\boldsymbol{\Lambda}(z)$ is also denoted as connection polynomial. The smallest integer ν fulfilling (2.18) is the linear complexity of the sequence $\boldsymbol{s}^{(l)}$. For a minimum length LFSR, the length of the connection polynomial equals the linear complexity of a given set of sequences. As soon as one minimum length LFSR is found for all sequences given by $\boldsymbol{S}^{(l)}\ \forall\ l \in [1, l_{\max}]$, the connection polynomial equals the locator polynomial, and therefore, the key equation is solved with $\nu = \tau$.

In order to determine a minimum length LFSR, the different lengths of the sequences need to be equalized with $S_i^{(l)} = 0$ for $i \in \left[n - k^{(l)}, n - k - 1\right]$ resulting in [SS06]:

$$S_{i+k^{(l)}-k}^{(l)} + \sum_{\xi=1}^{\nu} \Lambda_\xi S_{i+k^{(l)}-k-\xi}^{(l)} = 0 \quad \forall\, i \in \left[k - k^{(l)} + \nu, n - k^{(l)} - 1\right] \tag{2.19}$$

Based on this extension, the multi-sequence shift-register synthesis determines the minimum length LFSR by iteratively increasing and/or modifying the connection polynomial until all given sequences can be generated. For the description of the iterative process, the connection polynomial for a sequence of length t is denoted by $\boldsymbol{\Lambda}^{[t]}(z)$ and its length by ν_t. Based upon (2.19), the discrepancy $\Delta_t^{(l)}$ is introduced in order to evaluate whether $\boldsymbol{\Lambda}^{[t]}(z)$ can generate the sequence $\boldsymbol{s}^{(l)}$:

$$\Delta_t^{(l)} = s_{t+k^{(l)}-k}^{(l)} + \sum_{\xi=1}^{\nu_t} \Lambda_\xi^{[t]} s_{t+k^{(l)}-k-\xi}^{(l)} \tag{2.20}$$

Thus, $\Delta_t^{(l)} \neq 0$ indicates that the current connection polynomial $\boldsymbol{\Lambda}^{[t]}(z)$ is not able to generate the first t components of the sequence $\boldsymbol{s}^{(l)}$, and therefore, it needs to be modified. The length of $\boldsymbol{\Lambda}^{[t]}(z)$ might be increased according to $\nu_t = \max\{\nu_{t-1}, t - \nu_{t-1}\}$. The new connection polynomial is subsequently given by

$$\boldsymbol{\Lambda}^{[t]}(z) = \boldsymbol{\Lambda}^{[t-1]}(z) - \frac{\Delta_t^{(l)}}{\Delta_\zeta^{(l)}} z^{t-\zeta} \boldsymbol{\Lambda}^{[\zeta-1]}(z),$$

where the last length change took place at a sequence length of ζ. The complete SS algorithm is given in the Appendix (Algorithm A.1 on page 103) with the corresponding initial values according to [SS06].

Extended Euclidean Type Algorithms

The use of the Euclidean algorithm for channel coding goes back to Sugyiama et al. [SKHN75]. Instead of integers, polynomials are usually considered in the context of channel coding. The Euclidean algorithm can be described by repeated divisions for the polynomials $\boldsymbol{\rho}_{-1}(z)$ and $\boldsymbol{\rho}_0(z)$ with $\deg \boldsymbol{\rho}_0(z) \leq \deg \boldsymbol{\rho}_{-1}(z)$ [MS88]:

$$\begin{aligned}
\boldsymbol{\rho}_{-1}(z) &= \boldsymbol{q}_1(z)\boldsymbol{\rho}_0(z) + \boldsymbol{\rho}_1(z), && \deg \boldsymbol{\rho}_1 < \deg \boldsymbol{\rho}_0, \\
\boldsymbol{\rho}_0(z) &= \boldsymbol{q}_2(z)\boldsymbol{\rho}_1(z) + \boldsymbol{\rho}_2(z), && \deg \boldsymbol{\rho}_2 < \deg \boldsymbol{\rho}_1, \\
&\vdots && \vdots \\
\boldsymbol{\rho}_{i-2}(z) &= \boldsymbol{q}_i(z)\boldsymbol{\rho}_{i-1}(z) + \boldsymbol{\rho}_i(z), && \deg \boldsymbol{\rho}_i < \deg \boldsymbol{\rho}_{i-1}, \\
\boldsymbol{\rho}_{i-1}(z) &= \boldsymbol{q}_{i+1}(z)\boldsymbol{\rho}_i(z), &&
\end{aligned}$$

where the last non-zero remainder $\boldsymbol{\rho}_i(z) = \gcd(\boldsymbol{\rho}_{-1}(z), \boldsymbol{\rho}_0(z))$ is a greatest common divisor of $\boldsymbol{\rho}_{-1}(z)$ and $\boldsymbol{\rho}_0(z)$. In comparison to the Euclidean algorithm, the *Extended Euclidean* (EE) algorithm provides additionally the polynomials $\boldsymbol{u}(z)$ and $\boldsymbol{v}(z)$, such that

$$\gcd(\boldsymbol{\rho}_{-1}(z), \boldsymbol{\rho}_0(z)) = \boldsymbol{\rho}_{-1}(z)\boldsymbol{u}(z) + \boldsymbol{\rho}_0(z)\boldsymbol{v}(z), \tag{2.21}$$

with $\deg \boldsymbol{u}(z), \deg \boldsymbol{v}(z) < \deg \boldsymbol{\rho}_{-1}(z)$. Commonly, (2.21) is also known as Bézout's identity [Bé79]. The EE algorithm calculates iteratively with $\boldsymbol{u}_{-1}(z) = 0,\ \boldsymbol{u}_0(z) = 1,\ \boldsymbol{v}_{-1}(z) = 1$ and $\boldsymbol{v}_0(z) = 0$

$$\begin{aligned}
\boldsymbol{\rho}_i(z) &= \boldsymbol{\rho}_{i-2}(z) - \boldsymbol{q}_i(z)\boldsymbol{\rho}_{i-1}(z), \\
\boldsymbol{u}_i(z) &= \boldsymbol{u}_{i-2}(z) - \boldsymbol{q}_i(z)\boldsymbol{u}_{i-1}(z), \\
\boldsymbol{v}_i(z) &= \boldsymbol{v}_{i-2}(z) - \boldsymbol{q}_i(z)\boldsymbol{v}_{i-1}(z),
\end{aligned} \tag{2.22}$$

where $\boldsymbol{q}_i(z)$ is determined by the first equation in (2.22), such that $\deg \boldsymbol{\rho}_i < \deg \boldsymbol{\rho}_{i-1}$.

The EE algorithm can be applied in channel coding by solving the key equation (2.15) with $\boldsymbol{\rho}_{-1}(z) = z^{n-k}$ and $\boldsymbol{\rho}_0(z) = \boldsymbol{S}(z)$. Stopping the EE algorithm as soon as $\deg \boldsymbol{\rho}_i(z) < {}^{n-k}/_2$ and $\deg \boldsymbol{\rho}_{i-1}(z) \geq {}^{n-k}/_2$ results in $\boldsymbol{\Omega}(z) = \boldsymbol{\rho}_i(z)$ and $\boldsymbol{\Lambda}(z) = \boldsymbol{v}_i(z)$.[3] More details about EE algorithm and its application to channel coding can be found in corresponding textbooks, e.g., [MS88, Bos99, Moo05].

In order to perform power decoding, the known EE algorithm needs to be generalized. As it is pointed out in [Nie13], the *Mulders–Storjohann* (MS) algorithm [MS03] can be considered as such a generalization. Therefore, its application to power decoding according to [Nie13] is described briefly in the following. Subsequently, both algorithms are addressed as EE type due to their common principle.

For the description, vectors of polynomials $\boldsymbol{v}[z] = (\boldsymbol{v}_0(z), \ldots, \boldsymbol{v}_{l_{\max}}(z)) \in \mathbb{F}[z]^{l_{\max}+1}$ are used. The degree of such a polynomial vector is given by $\deg \boldsymbol{v}[z] = \max_{i \in [0, l_{\max}]} (\deg \boldsymbol{v}_i(z))$ and the leading position is denoted by $\mathrm{LP}\, \boldsymbol{v}[z] = \arg\max_{i \in [0, l_{\max}]} (\deg \boldsymbol{v}_i(z) = \deg \boldsymbol{v}[z])$. The polynomial at $\mathrm{LP}\, \boldsymbol{v}[z]$ is described as leading term $\mathrm{LT}\, \boldsymbol{v}[z] = \boldsymbol{v}_{\mathrm{LP}\, \boldsymbol{v}[z]}(z)$.

For the case of power decoding, there are $l_{\max}$ key equations, cf. (2.15) on page 12:

$$\begin{aligned} &\boldsymbol{\Lambda}(z)\boldsymbol{S}^{(l)}(z) = \boldsymbol{\Omega}^{(l)}(z) \mod z^{n-k^{(l)}} \\ &\deg \boldsymbol{\Omega}^{(l)}(z) < \deg \boldsymbol{\Lambda}(z) \end{aligned} \qquad \forall\ l \in [1, l_{\max}]\,, \tag{2.23}$$

where $\deg \boldsymbol{\Lambda}(z)$ should be minimal. A solution to (2.23) can be represented by the vector $(\boldsymbol{\Lambda}(z), \boldsymbol{\Omega}_1(z), \ldots, \boldsymbol{\Omega}_{l_{\max}}(z)) \in \mathbb{F}[z]^{l_{\max}+1}$ which lies in the solution space $\mathcal{M}$, where $\mathcal{M}$ can be described as row-space of the basis

$$\boldsymbol{M}[z] = \begin{pmatrix} 1 & \boldsymbol{S}^{(1)}(z) & \boldsymbol{S}^{(2)}(z) & \cdots & \boldsymbol{S}^{(l_{\max})}(z) \\ & z^{n-k^{(1)}} & & & \\ & & z^{n-k^{(2)}} & & \mathbf{0} \\ & \mathbf{0} & & \ddots & \\ & & & & z^{n-k^{(l_{\max})}} \end{pmatrix}. \tag{2.24}$$

A basis $\boldsymbol{P}[z] \in \mathbb{F}[z]^{(l_{\max}+1)\times(l_{\max}+1)}$ is in weak Popov form if the leading positions of all rows are different [MS03]. Consequently, as shown in [Nie13], the row $\boldsymbol{p}[z]$ of a basis in weak Popov form $\boldsymbol{P}[z]$ with $\mathrm{LP}\, \boldsymbol{p}[z] = 0$ contains the error locator polynomial $\boldsymbol{\Lambda}(z) = \boldsymbol{p}_0(z)$.

The weak Popov form can be reached by applying the Mulders–Storjohann algorithm to $\boldsymbol{M}[z]$ [MS03]. Therein, row reductions are performed until the weak Popov form is reached. These row reductions select two rows with $\mathrm{LP}\, \boldsymbol{p}_i[z] = \mathrm{LP}\, \boldsymbol{p}_j[z]$ and $\deg \boldsymbol{p}_i[z] \leq \deg \boldsymbol{p}_j[z]$. Subsequently, $\boldsymbol{p}_j[z]$ is replaced with $\boldsymbol{p}_j[z] - \xi z^{\zeta} \boldsymbol{p}_i[z]$, where $\xi \in \mathbb{F}$ and $\zeta \in \mathbb{N}_0$ are chosen such that the highest coefficient in the leading term $\mathrm{LT}\, \boldsymbol{p}_j[z]$ is canceled.

Since the error locator polynomials obtained by the previously presented algorithms provide only the error positions, the error values need to be determined subsequently by error evaluation algorithms. In the following, brief descriptions of the algorithms by *Gorenstein–Zierler* (GZ) and Forney are provided.

[3] The polynomials obtained by the EE algorithm are unique to within a factor [Bos99].

Gorenstein–Zierler Algorithm

Due to the definition of the syndrome, the following linear matrix equation can be set up:

$$\underbrace{\begin{pmatrix} \alpha^{u_1 k} & \alpha^{u_2 k} & \cdots & \alpha^{u_\tau k} \\ \alpha^{u_1 (k+1)} & \alpha^{u_2 (k+1)} & \cdots & \alpha^{u_\tau (k+1)} \\ \vdots & \vdots & & \vdots \\ \alpha^{u_1 (n-1)} & \alpha^{u_2 (n-1)} & \cdots & \alpha^{u_\tau (n-1)} \end{pmatrix}}_{\boldsymbol{W}} \cdot \underbrace{\begin{pmatrix} e_{u_1} \\ e_{u_2} \\ \vdots \\ e_{u_\tau} \end{pmatrix}}_{\boldsymbol{e}} = \underbrace{\begin{pmatrix} S_0 \\ S_1 \\ \vdots \\ S_{n-k-1} \end{pmatrix}}_{\boldsymbol{s}} \tag{2.25}$$

This overdetermined linear system can be solved by the GZ algorithm which utilizes matrix inversion: $\widehat{\boldsymbol{e}} = \boldsymbol{W}_\tau^{-1} \boldsymbol{s}$, where the first τ rows of $\boldsymbol{W}$ form a square matrix and are denoted as $\boldsymbol{W}_\tau$ [GZ61]. Since $\boldsymbol{W}_\tau$ is still a Vandermonde matrix, it is invertible for the correct error positions. Due to the use of a general matrix inversion, the GZ algorithm is of high complexity, and therefore, usually not considered in practice.

Forney's Algorithm

The algorithm by Forney [FJ65] utilizes the Vandermonde structure of $\boldsymbol{W}$ in (2.25) for error evaluation [Moo05]. By utilizing the error evaluator polynomial, cf. (2.15) on page 12, Forney's algorithm allows to calculate the error values by [Rot06]

$$\widehat{e}_{u_i} = -\alpha^{-u_i (k-1)} \frac{\boldsymbol{\Omega}(\alpha^{-u_i})}{\boldsymbol{\Lambda}'(\alpha^{-u_i})} \quad \forall\ u_i \in \mathcal{U}, \tag{2.26}$$

where $\boldsymbol{\Lambda}'(z)$ denotes the formal derivative of $\boldsymbol{\Lambda}(z)$. For the complete derivation of (2.26), the reader is referred to the recommended coding textbooks.

3

Underdetermined Systems of Linear Equations

A collection of several linear equations over a set of unknowns is commonly denoted as *System of Linear Equations* (SLE). The theory of SLEs is fundamental for the whole subject of linear algebra and its applications can be found in a plethora of fields. Since SLEs are of such fundamental relevance, there are many good text-books covering this subject, e.g., [Mey00, Lay12].

In principle, an SLE over real or complex numbers can have either infinitely many, a single, or no solution at all. In case of no solution, the system is called inconsistent since at least two equations contradict, cf. the well-known Kronecker–Capelli theorem [Cap92, Kro03, Mil10, Pra94]. For the common case of general position [Yal68, p. 164], which is assumed hereafter, overdetermined SLEs have less unknowns than equations and are therefore inconsistent. In most applications, the least squares solution which fits best to the given equations is targeted in these cases. In contrast to this, a unique solution can be identified as soon as every unknown variable is matched by one linearly independent equation. In case of an underdetermined system, there are more unknowns than equations, and consequently, infinite many solutions are possible. Within this chapter, such underdetermined SLEs are mainly considered.

Underdetermined systems have already been implicitly established over finite fields by Equations (2.14) and (2.17) within the domain of channel coding in Section 2.2.2. For this case, the number of non-zero coefficients τ of the error locator polynomial is minimal. For many applications, such τ-sparse solutions are of interest. This circumstance actually led to the success and popularity of CS, where sparsity is an important key property. Both scientific fields are unified in the search of the sparsest solution to underdetermined SLEs.

In the following, common non-sparse approaches to underdetermined SLEs over a real or complex vector space, which are subsequently used within this thesis, are given in Section 3.1. Thereby, the solution with minimal Euclidean norm is considered, which can be obtained by the Moore–Penrose pseudoinverse. Afterwards, the concept of linear optimization is briefly introduced which allows to find an optimal solution with respect to a given linear objective function. In Section 3.2 on page 20, sparsity is used as criterion to identify one out of infinite many solutions. Therefore, the solution space is described by intersecting hyperplanes and its dimensionality is determined for typical cases. Based on these derivations, it is shown subsequently that sufficiently sparse solutions are unique with high probability. Furthermore, a necessary and sufficient condition for the existence of a unique sparsest solution is provided which is based on the null space. These observations provide the foundation for CS, which is introduced in detail within the subsequent chapter.

3.1. Common Approaches to Underdetermined SLE

In the following, the n unknowns of an underdetermined SLE are represented by the vector $\boldsymbol{x} \in \mathbb{K}^n$ with $\mathbb{K} \in \{\mathbb{C}, \mathbb{R}\}$. The corresponding m equations are given by a coefficient matrix $\boldsymbol{A} \in \mathbb{K}^{m \times n}$ and by a vector $\boldsymbol{b} \in \mathbb{K}^m$ representing the constant terms, where $m < n$:

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$$

As it has been stated before, there are infinite many solutions for underdetermined and consistent SLEs. Depending on the actual problem described by the SLE, a certain solution among the infinite many might be more suited than others. Such examples can be stated as optimization problems, where an optimal solution is searched among the allowed solutions.

In the following, two common non-sparse approaches to underdetermined SLEs are provided as they are subsequently used in the thesis: The minimum norm solution to a given system can be obtained by the well-known Moore-Penrose pseudoinverse as it is described in Section 3.1.1. Beyond that, the more general scenario of linear optimization can be used to determine an optimal solution with respect to some given linear objective function as introduced in Section 3.1.2 on the facing page.

3.1.1. Minimum Norm Solution Provided by Moore–Penrose Pseudoinverse

Among the infinite many solutions of an SLE, the minimum norm solution is often especially interesting for engineering problems since it corresponds to the solution with minimal energy. With the help of the Moore–Penrose pseudoinverse

$$\begin{aligned} \boldsymbol{A}^\dagger &= \left(\boldsymbol{A}^{\mathrm{H}}\boldsymbol{A}\right)^{-1}\boldsymbol{A}^{\mathrm{H}} && \text{for } \operatorname{rank}\boldsymbol{A} = n \\ \boldsymbol{A}^\dagger &= \boldsymbol{A}^{\mathrm{H}}\left(\boldsymbol{A}\boldsymbol{A}^{\mathrm{H}}\right)^{-1} && \text{for } \operatorname{rank}\boldsymbol{A} = m, \end{aligned}$$

which has been independently described by Moore, Bjerhammar and Penrose [Moo20, Bje51, Pen55], this solution can be obtained:

$$\widehat{\boldsymbol{x}} = \boldsymbol{A}^\dagger\boldsymbol{b} = \underset{\boldsymbol{x}\in\mathbb{K}^n}{\operatorname{argmin}} \|\boldsymbol{x}\| \quad \forall \quad \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$$

In case the Moore–Penrose pseudoinverse is applied to an inconsistent SLE, a least squares solution

$$\widehat{\boldsymbol{x}} = \underset{\boldsymbol{x}\in\mathbb{K}^n}{\operatorname{argmin}} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|$$

is obtained, which is of minimal error. This circumstance is later used in Section 7.2.1 in order to cope with additional noise.

For details and further properties about the given pseudoinverse, refer to standard literature, e.g., [GVL96]. Since the solutions of the Moore–Penrose pseudoinverse are generally not sparse, they are not directly suited for CS. However, the pseudoinverse is later used for illustration and it is applied within several algorithms in order to obtain least squares solutions.

3.1.2. Linear Optimization

As mentioned before, the search for a particular solution among the infinite many of an underdetermined SLE can be considered as optimization problem, where the optimum is searched with respect to a given objective function. In the following, a linear objective function is considered which is typically expressed for all variables by the vector multiplication $\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}$ with $\boldsymbol{c} \in \mathbb{K}^n$. Such an optimization problem is, for the case of $\mathbb{K} = \mathbb{R}$, also known as *Linear Program* (LP). Often, the general case of inequalities is also used to describe such LPs.

In the literature, several forms and notations of LPs have been introduced, which are however not uniform. In the following, the definition of [BV09] is used, where two special cases of LPs are given:

- the *inequality form*

$$\min \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} \quad \text{s.t.} \quad \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}$$

- the *standard form*

$$\min \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} \quad \text{s.t.} \quad \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \text{ and } \boldsymbol{x} \geq \boldsymbol{0}$$

Thereby, the inequalities over vectors are taken component-wise.

Optimization problems, which are stated in another form, can be simply transformed: The sign of the inequalities can be inverted by multiplying with -1. An equality can be formulated by the intersection of two inequalities: $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \Leftrightarrow \boldsymbol{A}\boldsymbol{x} \geq \boldsymbol{b} \cap \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}$. Missing non-negativity constraints on the variables $\boldsymbol{x}$ can be avoided by the following substitutions:

given	substitution	constraint
$\boldsymbol{x} \in \mathbb{R}^n_{\leq 0}$	$-\boldsymbol{x}_1$	$\boldsymbol{x}_1 \in \mathbb{R}^n_{\geq 0}$
$\boldsymbol{x} \in \mathbb{R}^n$	$\boldsymbol{x}_1 - \boldsymbol{x}_2$	$\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathbb{R}^n_{\geq 0}$

A maximization of $\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}$ corresponds to a minimization of $\boldsymbol{c}_1^{\mathrm{T}}\boldsymbol{x}$, with $\boldsymbol{c}_1 = -\boldsymbol{c}$. By these transformations, an LP of a certain form can be translated into any other form.

There are several solvers for LPs available. The simplex method [DT97] is a well-known example, which is more closely presented in Chapter 5, where an extension for sparsity awareness is proposed. Another kind of popular solvers are the interior point algorithms for which the running-time is proven to be polynomial [Kar84], where the simplex method comes with an exponential running-time in the worst case [KM72].

Since LPs are known to be convex [BV09], all solvers provide the same optimal solution if existing. The LP-related research has therefore separated into two relatively distinct communities: The performance and robustness of solvers is an independent research field. On the other side, much research-effort is spent on the modeling of a given problem as LP, where the solver is merely applied as tool in the end of the process. Similarly, LPs can also be applied in CS, where a convex relaxation, the so called *Basis Pursuit* (BP), provides an LP as it is subsequently shown in Chapter 4. By reunifying both research strands, the previously mentioned simplex method is extended in order to provide sparsity awareness for CS in Chapter 5.

3.2. Existence of Unique Sparse Solutions

In the previous section, common non-sparse approaches for underdetermined SLEs have been presented. Within the context of channel coding, the sparsest solution to an underdetermined SLE is searched in Section 2.2.2. Sparse solutions to such SLEs are also of fundamental interest in CS as it is subsequently shown in Chapter 4. In order to pave the ground for the subsequent chapter on CS, two necessary conditions on the existence of a unique sparse solution are introduced within this section for SLEs over complex or real numbers. The first condition can be used for a probabilistic guarantee, while the second is also sufficient.

The following derivations and explanations of this section are mainly based on the unpublished lectures and tutorials of Lazich [Laz13, Laz14]. However, similar results and observations can also be found in [VB98, FR13].

3.2.1. Dimensionality of Intersecting Hyperplanes

A hyperplane is defined as an $(n-1)$-dimensional subspace of the n-dimensional ambient space, consequently, it corresponds to an $(n-1)$-flat. Within an SLE, each equation describes an individual hyperplane which contains all points fulfilling this equation. Thereby, each equation reduces the degree of freedom in the choice of a potential solution. Naturally, the intersection of all these hyperplanes describes the corresponding solution space of the SLE which is consequently an r-flat, where the actual value of r is determined by the SLE as subsequently shown.

In the following, it is assumed that all sets of points and subspaces are in general position if not stated otherwise, cf. [Yal68, p. 164], which essentially ensures that no set of p points is contained in a $(p-2)$-flat. Consequently, any p-flat is determined by at least $p+1$ points. The intersection of a p-flat and a q-flat within an n-dimensional space results in an r-flat. The dimensionality of this r-flat is discussed by Sommerville in [Som29, Ch. 1.12] and a more detailed description can be found in [Har97, Ch. 7]:

$$p + q - n \leq r \leq \min\{p, q\},$$

where the left equality is valid for the case of general position. Consequently, in case of $p + q < n$, both flats can be skew with no common point.

This can be extended to the case of m intersecting subspaces $\mathcal{S}_1, \ldots, \mathcal{S}_m$:

$$\sum_{i=1}^{m} \dim \mathcal{S}_i - n(m-1) \leq \dim \bigcap_{i=1}^{m} \mathcal{S}_i \leq \min_i \dim \mathcal{S}_i \tag{3.1}$$

For m hyperplanes $\mathcal{H}_1, \ldots, \mathcal{H}_m$, which are intersecting into an r-flat, this extends to

$$n - m \leq \dim \bigcap_{i=1}^{m} \mathcal{H}_i = r \leq n - 1. \tag{3.2}$$

This relation confirms the number of solutions for an SLE in general position: A unique solution exists only in case of $m = n$, no solution is expected for $n < m$, and infinitely many solutions are possible for the underdetermined case of $n > m$.

In the following section, this result is used in order to illustrate how sparsity introduces the uniqueness which is necessary to identify the sparsest solution.

3.2.2. Sparsity Provides Uniqueness

Necessary Sparsity Level for Uniqueness

A τ-dimensional Cartesian subspace τ-$\mathcal{CS}$ of an n-dimensional ambient space is described by the τ included coordinate axes. There are $\binom{n}{\tau}$ of such subspaces. For every τ-$\mathcal{CS}$, all included points have at most τ non-zero values. Consequently, all τ-sparse points are located in τ-dimensional Cartesian subspaces.

Based on these observations, it is clear that τ-sparse solutions to an SLE exist only in the s-flat which corresponds to the intersection of the solution space with all τ-$\mathcal{CS}$. Considering Equations (3.1) and (3.2), the dimensionality $s = \dim\left(\tau\text{-}\mathcal{CS} \cap \bigcap_{i=1}^{m} \mathcal{H}_i\right)$ of the resulting flat for τ-sparse solutions is bounded by

$$\tau - m \leq \ s \leq n - m.$$

Consequently, there are typically no sparse solutions with less than m non-zero components for underdetermined systems in general position. Furthermore, it is not possible to reconstruct a certain τ-sparse vector for $\tau > m$ only based on its sparsity since there are infinitely many solutions. For the special boundary case of $\tau = m$, there are $\binom{n}{\tau}$ different τ-sparse solutions.

On a first sight, this seems to contradict sparse recovery approaches like CS, where it is possible to recover unique sparse solutions. However, the existence of a sufficiently sparse solution with $\tau < m$ is presupposed in these cases. As a consequence from the previous derivations, it can be therefore argued that this solution is unique with high probability. Obviously, this is a necessary condition for a successful reconstruction, and therefore, fundamental to the sparse recovery problems which arise in the context of CS.

Necessary and Sufficient Condition by the Null Space

However, the previous condition is obviously not sufficient for all SLEs with sparse solutions as it is already implied by the assumption of general position. With the help of the null space $\ker \boldsymbol{A} = \{\boldsymbol{x} \in \mathbb{K}^n : \boldsymbol{A}\boldsymbol{x} = \boldsymbol{0}\}$ (sometimes also denoted as kernel) of a matrix $\boldsymbol{A} \in \mathbb{K}^{m \times n}$, a necessary and sufficient condition can be derived: As long as the null space does not contain any vector with 2τ or less non-zero components (with exception of $\boldsymbol{0}$), every τ-sparse vector is the unique sparsest solution. Since for two τ-sparse vectors $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, both resulting in the same vector $\boldsymbol{b} = \boldsymbol{A}\boldsymbol{x}_1 = \boldsymbol{A}\boldsymbol{x}_2$, the subtraction leads to $\boldsymbol{A}(\boldsymbol{x}_1 - \boldsymbol{x}_2) = \boldsymbol{0}$, where $\boldsymbol{x}_1 - \boldsymbol{x}_2$ has at most 2τ non-zero components. Since $\boldsymbol{0}$ is the only sparse vector in the null space, equality $\boldsymbol{x}_1 = \boldsymbol{x}_2$ is enforced. For details and further derivations regarding this null space observation, refer to [FR13]. Since this condition is sufficient for all τ-sparse vectors, it is generally very pessimistic as it is observed subsequently.

Sparse solutions to underdetermined SLEs represent the direct connection of channel coding and CS. Within this brief chapter on underdetermined SLEs, common non-sparse approaches to such linear systems have been presented, which are applied as intermediate step in subsequent algorithms or even extended for sparse recovery. For the existence of a unique sparsest solution, necessary general conditions have been established in a further step. Motivated by the potential existence of a unique sparsest solution to an underdetermined SLE, its efficient recovery leads to the principles of CS which are introduced in the subsequent chapter.

4

Principles of Compressed Sensing

Within the last decade, CS gained a considerable amount of attention due to the award-winning[1] publications of Donoho [Don06a], who coined the term CS, and Candes & Tao [CT06]. These contributions acted as catalyzer for the new research field of CS, which attracted scientists from a plethora of different research fields. The interest in this field has not ceased as the recent establishment of the new six-year priority program (SPP 1798): *Compressed Sensing in Information Processing (CoSIP)* by the DFG (German Research Foundation) reflects the vivid interest in CS [DFG14].

The earliest contributions to sparse recovery in general are commonly associated with de Prony [Pro95] from 1795. The first modern results arose in the context of ℓ_1-minimization in the mid 1960s [Log65] which has been independently used by geophysicists in the late 1970s [TBM79] as well. Within the 1990s, interest in sparse recovery has increased and the basis for CS has been established [DL92, MZ93, Nat95, CDS98]. For more detailed overviews regarding the history of CS refer to [DDEK12, FR13].

The fundamental setting within CS is commonly described by an underdetermined SLE, where the sensing matrix linearly connects measurements with an unknown sparse vector. By the means of CS, it is possible to recover this sparse vector from a small number of its measurements under certain conditions.

Research on CS has spread on several subfields, where the following list is naturally not exhaustive: Due to the underlying recovery problem, there is considerable interest in reconstruction algorithms, where the requirements, e.g., reliability and speed, vary with potential applications [Ran11, BT15]. Another important point in research are properties of the SLE which favor successful recovery [HN07a, WZP+15], and therefore, lead to the search for new constructions of potential sensing matrices [CHJ10a, AMM12]. There are also further extensions of CS theory leading to related research fields as f.i. matrix completion [CP10]. Of course, a considerable amount of research is driven by potential applications, e.g., magnetic resonance imaging [HHL11], radar [HS09, End10] and the famous single-pixel-camera [DDT+08].

In the following Section 4.1, different types of well-known reconstruction algorithms are introduced which are subsequently extended or used as reference. Afterwards, several uniqueness assuring properties and their interconnections are briefly introduced in Section 4.2 on page 27. Based thereon, suitable sensing matrices are categorized and discussed in Section 4.3 on page 29. Finally, two variants of CS are given in Section 4.4 on page 31, which provide the basis for subsequently proposed contributions to CS.

[1] IEEE Information Theory Society Paper Award 2008

4.1. Reconstruction Algorithms

As previously mentioned, the unique sparsest solution to an underdetermined SLE is commonly searched within CS. In the following, the sensing matrix is denoted by $\boldsymbol{\Theta} \in \mathbb{K}^{m \times n}$. The corresponding measurements are contained in a vector $\boldsymbol{\beta} \in \mathbb{K}^m$ and the unknown sparse vector is given by $\boldsymbol{\chi} \in \mathbb{K}^n$. Commonly, CS problems are considered over the real or complex-valued vector space $\mathbb{K} = \{\mathbb{R}, \mathbb{C}\}$. The resulting underdetermined system ($m < n$) is usually given by the means of linear algebra:

$$\boldsymbol{\Theta}\boldsymbol{\chi} = \boldsymbol{\beta} \tag{4.1}$$

Typically, the ℓ_0-"norm"[2] is used as mathematical measure for the sparsity of the unknown vector $\boldsymbol{\chi}$ by determining the cardinality of its support set $\mathcal{U}$:

$$\|\boldsymbol{\chi}\|_{\ell_0} = \#\{i : \ \chi_i \neq 0\} \tag{4.2}$$

Thereby, the ℓ_0-"norm" equals the Hamming weight $\mathrm{w}(\boldsymbol{\chi})$ which is commonly used in channel coding, cf. Section 2.1.2 on page 7. The ℓ_0-"norm" is an asymptotic extension of the ℓ_p-(quasi)-norms

$$\begin{aligned} \|\boldsymbol{\chi}\|_{\ell_p} &= \left(\sum_{i=1}^{n} |\chi_i|^p \right)^{\frac{1}{p}}, \quad 0 < p < \infty, \\ \|\boldsymbol{\chi}\|_{\ell_\infty} &= \max_i |\chi_i|. \end{aligned} \tag{4.3}$$

For $1 \leq p \leq \infty$, the expression ℓ_p describes a norm and for $0 < p < 1$ a quasi-norm since the triangle inequality is not satisfied. In consequence, a norm is convex in contrast to a quasi-norm, cf. [Ela10, FR13].

The reconstruction problem can consequently be described as ℓ_0-optimization:

$$\widehat{\boldsymbol{\chi}} = \arg\min_{\boldsymbol{\chi}} \|\boldsymbol{\chi}\|_{\ell_0} \quad \text{subject to} \quad \boldsymbol{\Theta}\boldsymbol{\chi} = \boldsymbol{\beta} \tag{4.4}$$

This straightforward approach is proven to be *Non-deterministic Polynomial-time* (NP)-hard, cf. [Nat95, FR13], and therefore, intractable for large problems. Consequently, suboptimal approaches need to be considered.

In the following, the optimization problem (4.4) is approached by convex relaxation in Section 4.1.1, since it is known that such convex problems can be efficiently solved. As an alternative, iterative methods, which typically aim to solve the sparse optimization directly, are briefly discussed in the subsequent Section 4.1.2 on the next page.

4.1.1. Convex Relaxation

A suboptimal approach to the NP-hard ℓ_0-minimization is the well-known *Basis Pursuit* (BP) [CDS98], which replaces the ℓ_0-"norm" in (4.4) by a convex ℓ_1-norm resulting in the simple optimization problem:

$$\widehat{\boldsymbol{\chi}} = \arg\min_{\boldsymbol{\chi}} \|\boldsymbol{\chi}\|_{\ell_1} \quad \text{subject to} \quad \boldsymbol{\Theta}\boldsymbol{\chi} = \boldsymbol{\beta} \tag{4.5}$$

[2] Mathematically, this is actually neither a norm nor a quasi-norm since it is not homogeneous [FR13].

Since the ℓ_1-norm equals the summation of vector component magnitudes

$$\|\boldsymbol{\chi}\|_{\ell_1} = \sum_{i=1}^{n} |\chi_i| \ ,$$

the BP described in (4.5) can be transformed into an LP, cf. Section 3.1.2 on page 19. Thereby, the absolute value function need to be considered in the description, which can be achieved for example by the following translation into a standard form LP, where two non-negative vectors $\boldsymbol{\chi_1}$ and $\boldsymbol{\chi_2}$ are used therefore:

$$\begin{aligned}
\text{BP}: \quad & \widehat{\boldsymbol{\chi}} = \arg\min_{\boldsymbol{\chi}} \|\boldsymbol{\chi}\|_{\ell_1} && \text{s.t.} \quad \boldsymbol{\Theta\chi} = \boldsymbol{\beta} \\
\text{LP (standard form)}: \quad & \min_{\boldsymbol{x}} \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} && \text{s.t.} \quad \boldsymbol{Ax} = \boldsymbol{b} \text{ and } \boldsymbol{x} \geq \boldsymbol{0} \\
& \boldsymbol{A} = [\boldsymbol{\Theta}, -\boldsymbol{\Theta}] \\
& \boldsymbol{c} = \boldsymbol{1} \\
& \boldsymbol{b} = \boldsymbol{\beta} \\
& \boldsymbol{x}^{\mathrm{T}} = \left[\boldsymbol{\chi}_1^{\mathrm{T}}, \boldsymbol{\chi}_2^{\mathrm{T}}\right] \\
& \widehat{\boldsymbol{\chi}} = \boldsymbol{\chi}_1 - \boldsymbol{\chi}_2
\end{aligned} \tag{4.6}$$

There are several potential algorithms available to solve the LP given by (4.6). Typically these approaches are divided into simplex methods and interior-point algorithms. In Chapter 5, more details on reconstruction algorithms are given, where the focus is on the simplex method which is extended to be sparsity aware in order to favor the sparse solutions of (4.4).

As subsequently discussed in Section 4.4.1 on page 31, there might be additional perturbations, e.g., $\boldsymbol{\mu} = \boldsymbol{\Theta\chi} + \boldsymbol{\eta}$, which need to be considered by the reconstruction algorithm. (Refer to Section 4.4.1 for further details on different perturbation models.) The presented BP is not suited for such noisy scenarios. However, there are also convex relaxations available for these cases. For example, there is *Basis Pursuit DeNoising* (BPDN) [CDS98]:

$$\widehat{\boldsymbol{\chi}} = \arg\min_{\boldsymbol{\chi}} \frac{1}{2}\|\boldsymbol{\mu} - \boldsymbol{\Theta\chi}\|_{\ell_2}^2 + \lambda\|\boldsymbol{\chi}\|_{\ell_1}, \tag{4.7}$$

where $\lambda \geq 0$ is a free parameter controlling the similarity to BP. Furthermore, there are several other similar approaches, which provide the same solutions under certain circumstances: There are the well-known *Least Absolute Shrinkage and Selection Operator* (LASSO) [Tib96] or quadratically constrained BP. Refer to [FR13] for an overview and more details on the individual links between the approaches.

4.1.2. Iterative Methods

In contrast to the previously introduced convex relaxation, iterative methods aim to solve the intractable sparse recovery problem (4.4) directly. During the last decades, several iterative approaches have been introduced and extended. In the following, the focus will be on the *Orthogonal Matching Pursuit* (OMP) [PRK93], which is a well-known extension of the classical *Matching Pursuit* (MP) [MZ93] and subsequently used as reference. Further extensions and approaches are briefly mentioned and corresponding references are given at the end of this section.

Orthogonal Matching Pursuit (OMP)

The OMP has been independently introduced by several researchers, e.g., [CBL89, PRK93, DMZ94]. It is a greedy algorithm since the best fitting column of $\boldsymbol{\Theta}$ is added to a support set of active columns in each iteration until a given stopping criterion is met. This support set is subsequently used to determine a least squares solution to the recovery problem by an orthogonal projection via Moore–Penrose pseudo inverse. The whole procedure is formally given in Algorithm 4.1 according to [DDEK12, FR13]. There are several stopping criteria possible, e.g., the number of iterations or the norm on the residual $\boldsymbol{\beta} - \boldsymbol{\Theta}\widetilde{\boldsymbol{\chi}}$, as it has been used for the subsequent simulations (cf. Appendix H.1.2 on page 120).

Algorithm 4.1: Orthogonal Matching Pursuit algorithm [PRK93]

Input : sensing matrix $\boldsymbol{\Theta}$, measurement vector $\boldsymbol{\beta}$
Output: estimated sparse vector $\widehat{\boldsymbol{\chi}} = \widetilde{\boldsymbol{\chi}}$ supported on the set $\widehat{\mathcal{U}} = \widetilde{\mathcal{U}}$

1 $\widetilde{\mathcal{U}} \leftarrow \emptyset$, $\widetilde{\boldsymbol{\chi}} \leftarrow \mathbf{0}$ /* initialization */
2 **while** *stopping criterion is not met* **do**
3 $\quad \widetilde{\mathcal{U}} \leftarrow \widetilde{\mathcal{U}} \cup \arg\max_{i \in [1,n]} |\langle \boldsymbol{\theta}_i, \boldsymbol{\beta} - \boldsymbol{\Theta}\widetilde{\boldsymbol{\chi}} \rangle|$ /* add next best index to the support set */
4 $\quad \widetilde{\boldsymbol{\chi}}_{\widetilde{\mathcal{U}}} \leftarrow \boldsymbol{\Theta}^{\dagger}_{\widetilde{\mathcal{U}}} \boldsymbol{\beta}$ /* determine least squares solution for current support set */

In contrast to conventional MP, all coefficients corresponding to the support set $\widetilde{\mathcal{U}}$ are updated in every iteration by the orthogonal projection. However, this projection step comes with considerably increased computational effort which is of course especially intense for vectors $\boldsymbol{\chi}$ with larger support sets $\mathcal{U}$ since many iterations are required in such cases. Another disadvantage of OMP is that once an index is added to the support set $\widetilde{\mathcal{U}}$, it cannot be removed. This is especially disadvantageous if the sparsity is used as stopping criterion. These drawbacks are addressed by several OMP extensions, where the most common are *Compressive Sampling Matching Pursuit* (CoSaMP) [NT09], *Regularized Orthogonal Matching Pursuit* (ROMP) [NV08] and *Stagewise Orthogonal Matching Pursuit* (StOMP) [DTDS12].

Further Approaches

Besides the previously described MPs (including the OMP extensions), there are further promising approaches to the sparse recovery problem. For example, there are thresholding based approaches, which are usually quite fast due to the typicall threshold operation. A well-known example is the *Iterative Hard Thresholding* (IHT) [BD09]. Another popular approach, which is called *Approximate Message Passing* (AMP) [DMM09], gathered recently considerable attention and led to several extensions [Ran11, SS12, MG15]. It combines thresholding with *believe propagation* in graphical models based on BP and LASSO [DMM10a, DMM10b].

The choice of an ideal algorithm depends naturally on several factors: There is the actual system itself including its dimensions and the used number field. The available resources influence the selection of a certain algorithm as well as potential a-priori information and necessary reliability or quality of the reconstruction. Within this thesis, BP and OMP have been considered mainly as reference due to their generality and popularity.

4.2. Uniqueness Assuring Properties

With the increased interest in CS, research on uniqueness assuring properties has intensified. This led to a variety of interesting properties and corresponding extensions. Opposed to the necessary conditions on the existence of unique solutions stated previously in Section 3.2.2 on page 21, the following properties arose in the context of CS and can be directly related to guarantees for certain reconstruction algorithms.

In the following Section 4.2.1, the coherence is introduced as important uniqueness assuring property which is of central interest in Chapter 6. Subsequently, the famous *Restricted Isometry Property* (RIP) is briefly introduced in Section 4.2.2 on the following page as it is later used as reference.

4.2.1. Coherence

For the case of a sensing matrix which consists out of two concatenated bases, the coherence of the sensing matrix

$$\mathrm{M}(\boldsymbol{\Theta}) = \max_{i \neq j} \frac{|\langle \boldsymbol{\theta}_i, \boldsymbol{\theta}_j \rangle|}{\|\boldsymbol{\theta}_i\| \|\boldsymbol{\theta}_j\|} \quad \forall\, i, j \in [1, n]\,, \tag{4.8}$$

has been introduced as property for sparse recovery by BP in [DH01, EB02]. Subsequently, the results have been extended to the case of sensing matrices which consist out of multiple bases as well as general sensing matrices [GN03, DE03, Fuc04, Fuc03].

Often, the coherence is also expressed by the maximal off-diagonal value of the corresponding Gram matrix $\boldsymbol{G}$:

$$\mathrm{M}(\boldsymbol{\Theta}) = \max_{i \neq j} G_{i,j} \quad \text{with } \boldsymbol{G} = \boldsymbol{\Theta}^{\mathrm{H}} \boldsymbol{\Theta}$$

The coherence is trivially bounded by $0 \leq \mathrm{M}(\boldsymbol{\Theta}) \leq 1$ for normalized sensing matrices, as it can be orthogonal or some columns can be collinear. For the relevant non-trivial case of overcomplete sensing matrices, corresponding lower bounds on the coherence are given in Section 6.1.1 on page 50.

Extending the result of [DH01], it is shown in [DE03, Fuc04] that BP is successful as its result is unique and equivalent to the ℓ_0-minimization (4.4) for

$$\|\boldsymbol{\chi}\|_{\ell_0} \leq \frac{1 + \frac{1}{\mathrm{M}(\boldsymbol{\Theta})}}{2}. \tag{4.9}$$

Consequently, the number of non-zero components in the sparse vector $\boldsymbol{\chi}$ can be higher for sensing matrices of lower coherence. As shown in [Tro04], the condition, which is given by (4.9), is also valid for a reconstruction by OMP.

For the case of sensing matrices built by the concatenation of several bases, similar bounds on the sparsity are given in [FN03, GN03]. Therein, a separate tight bound for the uniqueness of the ℓ_0-minimization is given, which implies a non-bridgeable gap between the optimal reconstruction and its convex ℓ_1-relaxation.

The coherence of a given sensing matrix $\boldsymbol{\Theta}$ can be computed with low complexity. This circumstance makes the coherence an attractive criterion for the construction of sensing matrices as it is subsequently described.

4.2.2. Restricted Isometry Property

The RIP is one of the most known uniqueness assuring properties and has been introduced by Candès and Tao in [CT06, CT05]. It is actually based on *Restricted Isometry Constants* (RICs) δ_τ, which are defined as smallest non-negative quantity for which

$$(1-\delta_\tau)\|\boldsymbol{\chi}\|^2 \leq \|\boldsymbol{\Theta}\boldsymbol{\chi}\|^2 \leq (1+\delta_\tau)\|\boldsymbol{\chi}\|^2 \tag{4.10}$$

is fulfilled for all τ-sparse vectors $\boldsymbol{\chi}$. As a direct consequence, the singular values of any column submatrix $\boldsymbol{\Theta}_\mathcal{T}$, with $\#\mathcal{T} \leq \tau$, are restricted to the interval $[1-\delta_\tau, 1+\delta_\tau]$, cf. [CT05, FR13]. This circumstance is subsequently used for Monte Carlo experiments.

An RIP of a sensing matrix $\boldsymbol{\Theta}$ is an upper bound on the RICs[3] and requires that every subset of τ or less columns behaves approximately like an orthonormal system [CRTV05]. Within [CT05], it has been shown that the τ-sparse solution to the ℓ_0-minimization is unique as long as

$$\delta_{2\tau} < 1. \tag{4.11}$$

Similarly, the first RIP for the uniqueness of the BP solution is given in [CT05] as well. Corresponding RIPs have been proposed and refined by several researchers [CRTV05, Can08, FL09, CWX10, Fou10]. A more recent sufficient RIP for exact recovery of all τ-sparse vectors via BP is given in [CZ13] with

$$\delta_{2\tau} < \frac{1}{3}. \tag{4.12}$$

For the case of a quadratically constrained BP, similar bounds can be given which consider also the achieved error [FR13]. Since OMP might choose a wrong column in its first stage, it is impossible to provide an RIP which guarantees success within τ iterations [Don06b]. This can be avoided by performing more iterations (by using a different stopping criterion) [FR13] or by removing columns as the CoSaMP does for which RIPs exist [NT09].

As it can be seen by the numerical values in Equations (4.11) and (4.12), there is a gap observable which implies that BP cannot achieve the reconstruction performance of the direct ℓ_0-minimization. Since the BP is merely a convex relaxation, this is not further surprising.

It is proven in [BDMS13, TP14] that determining the RIC for a matrix is actually an NP-hard problem. However, it is possible to infer a lower bound on the RIC by Monte Carlo experiments. Based on the previous connection between the range restriction on the singular values of the column submatrix $\boldsymbol{\Theta}_\mathcal{T}$ and the definition of RICs [cf. (4.10)], the maximal occurring δ_τ can be obtained by Monte Carlo experiments over a large set of submatrices. Thereby, the corresponding ranges of singular values for several different subsets $\mathcal{T}$ lead to an empirical lower bound on the RIC.

The RIC of a column-normalized sensing matrix $\boldsymbol{\Theta}$ is connected to the previously introduced coherence by [CXZ09]

$$\delta_\tau \leq (\tau - 1)\ \mathrm{M}(\boldsymbol{\Theta}), \tag{4.13}$$

which is often used to construct sensing matrices with good RIP.

There are several extensions to the RIP, which are designed for special applications or are less restrictive, e.g., the *Dictionary-adapted RIP* (DRIP) [CENR11], the *Generalized RIP* (GRIP) [HN07b], or the *Statistical RIP* (StRIP) [CHJ10a].

[3] In some publications, *Restricted Orthogonality Constants* (ROCs) [CT05] are also used to express RIPs.

Another notable example for a condition on the suitability of a sensing matrix is the *Null Space Property* (NSP) [CDD09]. The RIP is used thereby for several proofs regarding reconstruction guarantees for BP and BPDN. In contrast to the necessary null space based condition of Section 3.2.2, the NSP considers also vectors which are approximatively sparse.

4.3. Sensing Matrices

As pointed out in the previous section about uniqueness assuring properties, a well-conditioned sensing matrix is crucial for a successful recovery. There are several variants to create suitable sensing matrices for which an overview is given in the following.

Potential sensing matrices $\mathbf{\Theta} \in \mathbb{K}^{m \times n}$ can be categorized into two groups based on the possible reconstruction approaches: All sensing matrices can be used in combination with the general reconstruction algorithms provided in Section 4.1. Beyond that, there are also deterministic matrices which are intentionally constructed such that they possess exploitable structures, which allow dedicated reconstruction algorithms. However, certain applications might enforce restrictions on the sensing matrix which prevent or at least limit deterministic approaches.

Within Section 4.3.1, the common approaches for sensing matrices are briefly summarized which includes random matrices, as well as potential optimization strategies, and explicit constructions. Deterministic sensing matrices are discussed in Section 4.3.2 on the following page, which allow dedicated recovery algorithms.

4.3.1. Sensing Matrices for General Reconstruction

Sensing matrices for general reconstruction are either based on random distributions, the result of a matrix optimization with respect to a uniqueness assuring property, or even deterministically constructed. In the following, these three categories are briefly summarized.

Random Matrices

The milestone publications [Don06a, CT06] focus on random sensing matrices. Thereby, the columns of an orthonormal basis or some tight frame are selected uniformly at random and combined to sensing matrices which allow to reconstruct sufficiently sparse vectors with high probability by BP. The authors of [CT05] extend their previous results by introducing the earlier described RIP which allows to provide reconstruction guarantees in case of Gaussian matrices (whose elements are independent and identically distributed Gaussian with zero mean and a fixed variance). These results have been subsequently generalized, e.g., to Bernoulli random variables and related distributions [BDDW08], or to the wider class of sub-gaussian distributions [Ver12]. A brief overview on random sensing matrices is provided in [DDEK12].

As an advantage, random sensing matrices can be typically obtained for any desired size. Furthermore, randomized sensing matrices can be applied in several applications which allows to give probabilistic reconstruction guarantees [Rom09, TLD+10]. As a drawback, there is the non-zero risk of obtaining an ill-conditioned sensing matrix. Furthermore, these matrices might be demanding in memory and computational requirements. The usage of pseudo-random number generators can reduce the memory demands since only a relatively small

seed needs to be stored. It is known that these random matrices fulfill uniqueness assuring properties with high probability, however, there is the potential of obtaining improved matrices by applying optimization strategies with respect to such beneficial properties.

Optimized Matrices

While the RIP is usually used to promote random sensing matrices, the computable coherence is commonly considered as criterion for sensing matrix optimization. Often, the previously given connection (4.13) is used to derive also guarantees on the RIP for such coherence based optimization approaches. However, such RIP guarantees are not competitive to those of random matrices [FR13].

Within Chapter 6 on page 49, coherence optimization is discussed separately in general. Therein, several numerical approaches are summarized in Section 6.1.2, which can be used to obtain low-coherence sensing matrices. Furthermore, a novel algorithm is provided in Section 6.2, which is based on the maximization of certain spherical codes with respect to their minimal distance.

Optimized matrices come typically with the advantage of improved reconstruction performance. The optimization process itself might be time-consuming, however, it is usually performed off-line during development, and therefore, corresponding complexity is of lower relevance. However, the memory requirements of such optimized matrices are potentially higher than for pseudo random variants, since the full matrix has to be stored compared to a potentially small random seed.

Deterministic Matrices

In order to avoid probabilistic performance guarantees and to reduce the potential memory consumption, several deterministic approaches for sensing matrices have been proposed. Typically, a minimal coherence is targeted by such approaches. Therefore, such sensing matrices correspond to vector sets which are obtained by analytical approaches for which an overview is given in Section 6.1.2 on page 54. As for the previously mentioned optimized matrices, RIP guarantees are typically provided by the given connection to the coherence, cf. (4.13). Thus, these guarantees share the same drawbacks.

Deterministic sensing matrices have the advantage of known and proven reconstruction guarantees. The matrices might be computed if needed and do not have to be stored in memory. However, they exist only for a limited range of matrix sizes. Furthermore, practical limitations might prevent the application of such approaches as previously mentioned.

4.3.2. Deterministic Sensing Matrices for Dedicated Reconstruction

Opposed to the deterministic matrices of the previous section, certain uniqueness assuring properties are not primarily targeted in this section. Here, the focus is on introducing exploitable structures, which can be utilized during the reconstruction by dedicated recovery algorithms. Thereby, the focus can be on either computational speed, memory consumption, or reconstruction performance. Of course, general recovery algorithms can still be applied, however, the performance is typically limited since beneficial properties are often sacrificed in order to establish the deterministic structures.

As aforementioned, many theoretical results in CS are based on randomness in the sensing process. There had been a similar situation in the early time of channel coding: Random codes have been used as an argument in order to prove Shannon's channel coding theorem [Sha48, Bos12]. However, for practical applications, non-random (deterministic) codes are used since they allow specialized decoding algorithms, which exploit the given code structure and allow efficient implementations. Therefore, it is not surprising that several researchers proposed coding based methods for deterministic CS [DeV07, PH08, HCS08, AHSC09, CHJ10a, AMM12]. Subsequently, CRS codes are used in Chapter 7 on page 75 as a deterministic CS scheme.

The previously mentioned drawbacks of deterministic sensing matrices for general algorithms apply here as well. Furthermore, such coding based deterministic CS schemes with dedicated coding-based reconstruction algorithms are typically rather sensitive to noise since robustness is no issue in coding theory where complexity reduction is of main interest. As it is shown in Chapter 7, this circumstance needs to be considered in corresponding approaches.

4.4. Variants of Compressed Sensing

With the fast growing interest, several related variants of CS have been proposed. Typically, basic CS problems are described by an underdetermined SLE, cf. (4.1). In the following, two basic variants are discussed which will be of interest in the subsequent chapters. In Section 4.4.1, the previously mentioned case of additional non-sparse noise is formally introduced. Subsequently, the scenario of sparsifying dictionaries is given in Section 4.4.2 on the following page.

4.4.1. Noisy Scenarios

Especially in the context of potential applications, noisy CS scenarios are typically discussed since measurement noise can never be completely avoided. Most commonly, the occurring noise is modeled as *Additive White Gaussian Noise* (AWGN). The additional noise terms can be considered at two different positions in a CS system:

$$\boldsymbol{\mu} = \boldsymbol{\Theta}\left(\boldsymbol{\chi} + \boldsymbol{\eta}_1\right) + \boldsymbol{\eta}_2,$$

where $\boldsymbol{\eta}_1 \in \mathbb{K}^n$ and $\boldsymbol{\eta}_2 \in \mathbb{K}^m$ represent the two possible additional terms. The effective noise $\boldsymbol{\Theta}\boldsymbol{\eta}_1$ corresponds to a filtered noise term and can also be modeled within $\boldsymbol{\eta}_2$. The parts of $\boldsymbol{\eta}_1$ which lie in the null space of $\boldsymbol{\Theta}$ do not influence the actual measurement and reconstruction and can therefore be neglected. The scaling of the noise for fair comparisons depends on the actual application, the noise sources occurring therein, as well as the constraints on the designed system. Consequently, there are many possible ways for comparison. In the following,

$$\boldsymbol{\mu} = \boldsymbol{\Theta}\boldsymbol{\chi} + \boldsymbol{\eta}$$

is used as noisy CS scenario. Refer to Appendix H.1 on page 119 for a precise definition how the components in $\boldsymbol{\eta}$ are obtained for the subsequent simulations. Several techniques for noisy scenarios can also be used for CS variants which describe the reconstruction of sparse approximations to non-sparse vectors [FR13].

Sometimes it is argued, why noiseless CS schemes are considered at all. Due to its generality, CS has connections to a plethora of different research fields. Naturally, each field comes with its own individual problems, requirements and assumptions. Especially for direct applications, noise is a major issue and needs to be addressed. However, in less practical research fields, the actual problem lies in solving the NP-hard sparse reconstruction and noise is of minor interest if at all. This discussion is closely related to the (almost philosophical) question on the importance of fundamental research.

4.4.2. Sparsifying Dictionaries

Another important CS variant utilizes so-called sparsifying dictionaries in order to allow the application of the CS methodology to a non-sparse vector $\boldsymbol{\zeta} \in \mathbb{K}^l$. Thereby, $\boldsymbol{\zeta}$ needs to have a sparse representation $\boldsymbol{\chi} \in \mathbb{K}^n$ with respect to a (possibly overdetermined $l \leq n$) basis denoted as dictionary $\boldsymbol{\Psi} \in \mathbb{K}^{l \times n}$:

$$\boldsymbol{\zeta} = \boldsymbol{\Psi}\boldsymbol{\chi}$$

A measurement matrix $\boldsymbol{\Phi} \in \mathbb{K}^{m \times l}$ can be applied on this non-sparse vector $\boldsymbol{\zeta}$ in order to obtain $\boldsymbol{\beta} = \boldsymbol{\Phi}\boldsymbol{\zeta}$, the so-called measurement with $\boldsymbol{\beta} \in \mathbb{K}^m$ and typically $m < l$. By combining both matrices into the sensing matrix $\boldsymbol{\Theta} = \boldsymbol{\Phi}\boldsymbol{\Psi}$, the well-known underdetermined SLE of CS is obtained, cf. (4.1). Since sparse representations are often applied in many practical scenarios, e.g., lossy compression [Mal08], such variants opened CS to a large variety of potential applications [Ela10].

This variant of CS raised several interesting problems and questions, e.g., whether a dictionary is already known or needs to be acquired by observations (dictionary learning [Ela10]), or which properties a suitable measurement matrix needs to possess [CENR11]. The latter question is often answered by two different yet similar properties. The authors of [CR07, CW08], argue that the rows of $\boldsymbol{\Phi}$ (the measurements) should be incoherent to the columns of $\boldsymbol{\Psi}$ (the elementary signals). This demand results in the following coherence-like description:

$$\mathrm{M}(\boldsymbol{\Phi}, \boldsymbol{\Psi}) = \max_{i,j} \frac{|\langle \boldsymbol{\phi}_i, \boldsymbol{\psi}_j \rangle|}{\|\boldsymbol{\phi}_i\| \|\boldsymbol{\psi}_j\|} \tag{4.14}$$

The intra row coherence of $\boldsymbol{\Phi}$ should be small as well since it is only reasonable to collect as much potentially different information as possible. However, it is also desirable to choose a potential measurement matrix such that the resulting sensing matrix $\boldsymbol{\Theta}$ suits the uniqueness assuring properties previously given in Section 4.2. Due to its computability, the sensing matrix coherence $\mathrm{M}(\boldsymbol{\Theta})$, cf. (4.8), is especially suited. Since both properties are based on the coherence, caution needs to be applied by reading the referred literature.[4] As an advantage of Gaussian measurement matrices, the resulting sensing matrix is again Gaussian for most typical dictionaries, and therefore, the earlier RIP-based probabilistic reconstruction guarantees can be directly applied [BDDW08].

The coherence optimization technique subsequently introduced in Section 6.2 on page 55 can be adapted in order to optimize the coherence between measurement matrix and dictionary, as shown within Section 6.3.2 on page 70. This allows to estimate the influence of both coherence-based criteria numerically as it is done in the corresponding section.

[4] In [CR07], it is referenced that [DH01, DE03] introduce coherence between measurement matrix and dictionary as in (4.14), however, the coherence of the sensing matrix (4.8) is actually described therein.

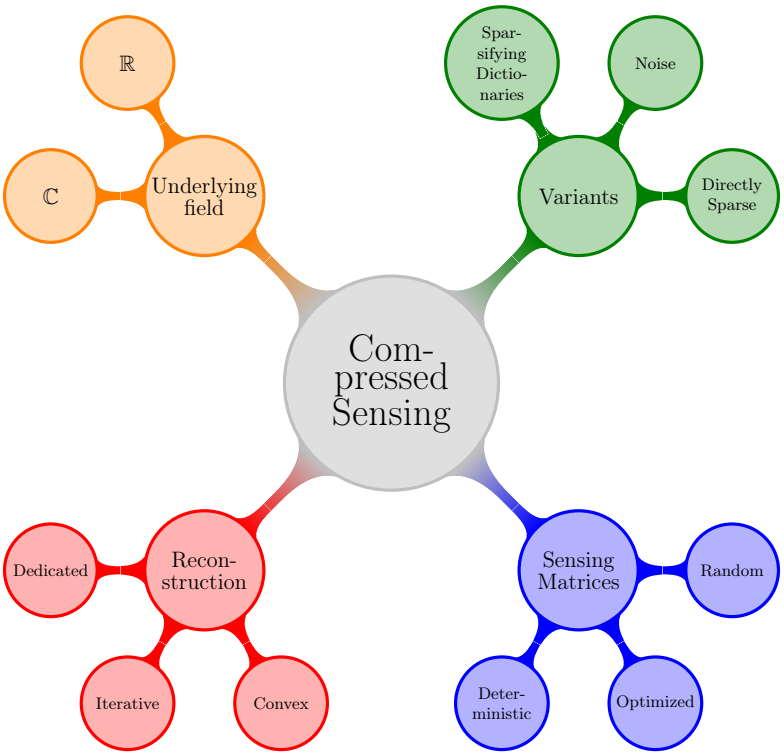


Figure 4.1.: Overview of several aspects in CS.

4.5. Summary and Overview

Within this chapter, several aspects of CS which are relevant for the subsequent contributions have been introduced. After establishing the CS problem as search for the sparsest solution to an underdetermined SLE, common reconstruction algorithms have been summarized in Section 4.1. Thereby, special focus is put on convex ℓ_1-norm relaxations and on iterative methods, where the OMP is of primary interest (among the plethora of iterative approaches) for the subsequent thesis. Two uniqueness assuring properties have been discussed in Section 4.2, where the computable coherence and the well-known RIP are highlighted. Potential constructions or sources for sensing matrices have been discussed altogether in Section 4.3. Thereby, matrices for general reconstruction algorithms are separated from special sensing matrices which allow also dedicated reconstruction algorithms. Finally, two basic variants on CS have been given in Section 4.4: noisy CS scenarios which consider additional non-sparse noise terms hindering the reconstruction, the CS variant of sparsifying dictionaries which allows to cover also non-sparse vectors by the CS theory as long as they can be sparsely represented with respect to some dictionary.

As it is clear from the current introductory chapter, CS resembles a large umbrella field for several different scenarios and objectives. In order to classify quickly the contributions of the subsequent chapters into the overall big picture of CS, the visualization of Figure 4.1 is introduced. Similar figures are used at the end of each contributing chapter, where the relevant aspects of the corresponding contribution are highlighted in order to allow a fast classification of the individual contribution. Since the approaches often differ for the underlying

field, it is indicated whether complex or real vector spaces are considered. The potentially applied reconstruction is separated into dedicated, iterative, and convex algorithms. For the used sensing matrices, deterministic, optimized, and random approaches are differentiated. And of course, the corresponding CS variant is provided, where the classical case of directly sparse vectors is possible as well as the previously introduced variants of noisy scenarios and sparsifying dictionaries.

5

Sparsity Aware Simplex Algorithms

Since the direct approach to the sparse recovery problem is commonly known to be NP-hard [Nat95, FR13], suboptimal approaches are considered, for example, those discussed in Section 4.1 on page 24. One of the prominent examples is the convex ℓ_1-relaxation called BP proposed by [CDS98]. Utilizing the ℓ_1-norm for sparse recovery has already an older history [Log65, DL92] with applications in geophysics [TBM79]. Besides the benefits for theoretical derivations, the possibility of transforming the ℓ_1-minimization into an LP (cf. Section 3.1.2 on page 19) contributed to the popularity of this optimization approach, since the well-known solvers for LPs can be applied.

An early and famous example for such an LP solver is the *simplex method* originally proposed by Dantzig [DT97] in 1947, which has been subsequently improved and revised by several researchers, e.g., [Bea54, Lem54, FG92]. The term simplex method had been coined by Motzkin who described the approach as a movement from one simplex to another [DT97]. Since there are interior point algorithms for which the running-time is proven to be polynomial [Kar84], those approaches are also very popular. However, simplex algorithms are often faster although their worst case running time is known to be exponential [KM72]. Within this chapter, the focus will be on the simplex method for which there are many good textbooks available providing detailed introductions, e.g., [NS95, DT97, BT97].

Subsequently, *Sparsity Aware Simplex Algorithms* (SASAs) are provided by utilizing degenerated vertices. Degeneracy is caused by redundant constraints and leads to solutions which contain one or more zeros [DT97]. In conventional simplex approaches, such vertices are usually unwanted, since they might lead to cycles which could prevent the algorithm from finding the solution to the optimization problem [DT97]. However, for sparse recovery problems, degenerated vertices are actually very attractive, since they are sparse by definition. For cases where the sparsest solution does not have minimal ℓ_1-norm, degenerated vertices allow therefore to improve on the reconstruction performance of a conventional BP. Consequently, this technique allows to investigate the "closeness" of the sparsest solutions with respect to the solution with minimal ℓ_1-norm for cases where those do not coincide and BP fails.

In the following Section 5.1, the conventional simplex method is introduced, where the simplex algorithm is applied within the so called two-phase method. Subsequently, degeneracy is used for sparsity awareness in Section 5.2 on page 39, where different variants for SASAs are proposed. These variants resemble a trade-off between additional complexity and improved reconstruction. Finally, within Section 5.3 on page 43, variants for the sparsity aware two-phase method are proposed and examined for their recovery potential in CS.

5.1. Simplex Method

One should note the difference between the simplex method and the simplex algorithm: For LPs in the inequality form (cf. Section 3.1.2 on page 19) with only $\leq$-relations ($\boldsymbol{b} \in \mathbb{R}^m_{\geq 0}$), the origin is a feasible initial solution and the simplex algorithm can be directly used to solve the linear optimization problem. In case of equality or $\geq$-relations (e.g., an LP in standard form, cf. Section 3.1.2), there is typically no such initial feasible solution known. As a consequence, the LP has to be extended in a first phase such that the simplex algorithm can be applied in order to find a feasible solution. Based on this solution, the LP can be reduced again and the simplex algorithm obtains in a second phase the optimal solution. The term simplex method describes thereby the general approach of solving LPs with the help of one or more instances of the simplex algorithm. The mentioned two-phase (simplex) method is subsequently used as an example for solving an LP in standard form. In Section 5.1.1, the simplex algorithm is introduced for LPs in inequality form. Subsequently, the two-phase method is explained in Section 5.1.2 on page 38 which allows to solve all forms of LPs. The following introduction is mainly based on [AHK+11, DT97, BT97].

5.1.1. Simplex Algorithm

The feasible solution space of an optimization problem in inequality form corresponds to a polytope, which is the intersection of all half-spaces given by the corresponding inequalities. Consequently, optimal solutions can only be located on the surface of this polytope, whereby the unique optimal solution corresponds to a certain vertex. Based on this observation, the fundamental idea of the simplex algorithm is to walk from one vertex to an improving vertex until the optimum is found. Figure 5.1 illustrates the concept for a minimization of $-x_3$.

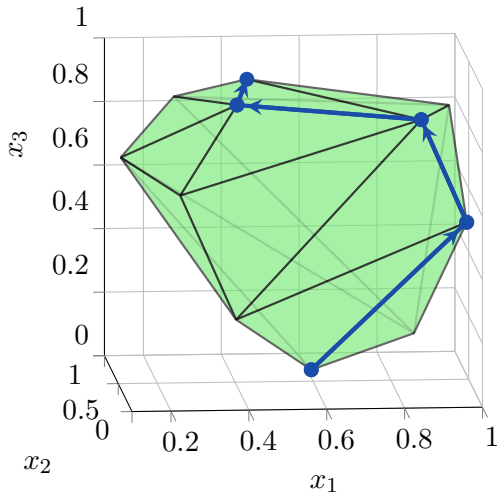


Figure 5.1.: Solution space of an LP in inequality form $\boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}$ with $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, $\boldsymbol{x} \in \mathbb{R}^n_{\geq 0}$, $\boldsymbol{b} \in \mathbb{R}^m$ visualized as polytope. Exemplary path of vertices covered by the simplex algorithm displayed in dark blue.

In order to apply the simplex algorithm, the constraints of an LP in inequality form $\boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}$ with $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, $\boldsymbol{x} \in \mathbb{R}^n_{\geq 0}$ and $\boldsymbol{b} \in \mathbb{R}^m_{\geq 0}$ need to be transformed into a system of linear

equations.[1] Therefore, slack variables $\widetilde{\boldsymbol{x}}$ are introduced into every inequality leading to

$$\boldsymbol{A}\boldsymbol{x} + \boldsymbol{I}\widetilde{\boldsymbol{x}} = \boldsymbol{b}, \tag{5.1}$$

with $\widetilde{\boldsymbol{x}} \in \mathbb{R}^m_{\geq 0}$ and $\boldsymbol{I}$ being the identity matrix. These additional variables can be interpreted as indicator for the degree of how close an inequality is to being equal. Naturally, at least one of the introduced slack variables is equal to zero on the surface of the polytope.

An under-determined linear system of m equations in $n + m$ variables [as in (5.1)] is said to be canonical if there are m variables (one for each equation) which occur in all equations only once with a 1 as coefficient. Consequently, the matrix representation of such a system contains m distinct canonical unit vectors as columns. This special form allows to identify directly a basic feasible solution: The non-zero components attain the values from $\boldsymbol{b}$ and are located according to the columns containing the canonical unit vectors. Such a basic feasible solution corresponds to a vertex of the polytope. For example, in (5.1), the unit columns identify $\widetilde{\boldsymbol{x}} = \boldsymbol{b}$ which corresponds to $\boldsymbol{x} = \boldsymbol{0}$ to the origin as currently selected vertex.

An LP, which comprises a system of linear equations in canonical form together with a corresponding objective function given by $\min_{\boldsymbol{x}} \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}$, can be represented by a simplex tableau

$$\left[\begin{array}{cc|c} \boldsymbol{A} & \boldsymbol{I} & \boldsymbol{b} \\ \hline \boldsymbol{c}^{\mathrm{T}} & \boldsymbol{0} & 0 \end{array}\right], \tag{5.2}$$

which is an extended matrix consisting out of known values from the original optimization problem. As argued before, the basic feasible solution corresponds to the vertex $\boldsymbol{p} \in \mathbb{R}^{n+m}$, which equals $\boldsymbol{p}^{\mathrm{T}} = \left[\boldsymbol{x}^{\mathrm{T}}, \widetilde{\boldsymbol{x}}^{\mathrm{T}}\right]$ for the given example.

For the subsequent description of the simplex algorithm, the components of a tableau are identified as:

$$\left[\begin{array}{ccc|c} a_{1,1} & \dots & a_{1,k} & b_1 \\ \vdots & \ddots & \vdots & \vdots \\ a_{l,1} & \dots & a_{l,k} & b_l \\ \hline \alpha_1 & \dots & \alpha_k & \alpha \end{array}\right], \tag{5.3}$$

where $k = n + m$ and $l = m$ for the current example of (5.2). As before, canonical unit vectors can be used to identify non-zero components of the current basic feasible solution $\boldsymbol{p}$, where the actual values are obtained from the corresponding entries in the right column of the tableau $(b_1 \dots b_l)$. The value of the objective function for the currently selected vertex $\boldsymbol{p}$ is given by $-\alpha$. Consequently, for the given minimization problem, the solution can only be improved if some positive value is added to α.

In order to walk from the current vertex to another (also denoted as simplex step), a new canonical unit vector has to be formed within a column of the tableau without changing the actual solution space. This can be accomplished by row operations (cf. Gaussian elimination). Consequently, the optimal solution is obtained as soon as $\alpha_i \geq 0$ for all $i \in [1, k]$, since only the selection of columns with $\alpha_i < 0$ improves on α. The vector component, which becomes 1 during the simplex step, is denoted as pivot element $a_{r,c}$. There are several pivoting strategies for selecting the corresponding pivot column c. In the following, two strategies are considered:

[1] Here, $\boldsymbol{b} \in \mathbb{R}^m_{\geq 0}$ is assumed. Negative values in $\boldsymbol{b}$ correspond to $\geq$-relations. Consequently, a transformation has to be performed as subsequently described in Section 5.1.2.

$$\begin{array}{ll} \text{Rule of Bland [Bla77]:} & c = \min\{i \mid \alpha_i < 0\} \\ \text{Smallest reduced cost:} & c = \underset{i}{\operatorname{argmin}}\{\alpha_i \mid \alpha_i < 0\} \end{array}$$

Obviously, the rule of Bland selects the column with the smallest index i, while the greedy strategy, denoted as smallest reduced cost, chooses the one with minimal α_i. Within column c, the pivot row r of the pivot element $a_{r,c}$ is selected by the minimum ratio test:

$$r = \underset{i \in [1,l]}{\operatorname{argmin}} \left\{ \frac{b_i}{a_{i,c}} \geq 0 \right\} \tag{5.4}$$

As soon as the pivot element $a_{r,c}$ has been identified, a simplex step can be performed by applying the corresponding row operations such that a new canonical unit vector is created in column c. The continuous application of such simplex steps until there is no better vertex reachable, i.e., $\alpha_i \geq 0$ for all $i \in [1, k]$, is denoted as simplex algorithm.

If the result of the minimum ratio test (5.4) is provided by ${}^{b_r}/_{a_{r,c}} = 0$, a degenerated vertex has been reached. Consequently, at least two equal ratios preceded such a vertex during the previous minimum ratio test. Such degenerated vertices are dangerous, since they might lead to cycles which can stall the algorithm as the value of the objective function is not further improved. It is possible to avoid such stalling cycles by a suitable choice for the pivot element. The rule of Bland is a simple and elegant example for an anti-cycling strategy [Bla77]. Since these degenerated vertices are given by $b_r = 0$, they contain additional zeros. Consequently, sparse solutions are located at such degenerated vertices. Therefore, degeneracy is subsequently used in Section 5.2 on the facing page to provide sparsity awareness.

5.1.2. Two-Phase Method

In case of an LP which is not given in inequality form $\boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}$ with $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, $\boldsymbol{x} \in \mathbb{R}^n_{\geq 0}$ and $\boldsymbol{b} \in \mathbb{R}^m_{\geq 0}$, the previously described simplex algorithm cannot be directly applied: Since the missing slack variables cannot provide a system of linear equations in canonical form, no basic feasible solution can be obtained directly. However, it is possible to determine a valid starting vertex of the polytope by introducing an additional preparation step called the first simplex phase. For a general LP of the form

$$\min \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} \quad \text{s.t.} \quad \boldsymbol{x} \geq \boldsymbol{0}, \quad \begin{array}{rcl} \boldsymbol{A}_{\mathrm{le}}\boldsymbol{x} & < & \boldsymbol{b}_{\mathrm{le}} \\ \boldsymbol{A}_{\mathrm{eq}}\boldsymbol{x} & = & \boldsymbol{b}_{\mathrm{eq}} \\ \boldsymbol{A}_{\mathrm{ge}}\boldsymbol{x} & > & \boldsymbol{b}_{\mathrm{ge}} \end{array},$$

with $\boldsymbol{b}_{\mathrm{le}},\ \boldsymbol{b}_{\mathrm{eq}},\ \boldsymbol{b}_{\mathrm{ge}} \geq \boldsymbol{0}$, artificial variables are introduced for equations and $>$-inequalities within this first phase. With the help of these additional variables, an enlarged simplex tableau can be created:

$$\left[\begin{array}{ccccc|c} \boldsymbol{A}_{\mathrm{le}} & \boldsymbol{I}_{\mathrm{slack}} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{b}_{\mathrm{le}} \\ \boldsymbol{A}_{\mathrm{ge}} & \boldsymbol{0} & -\boldsymbol{I}_{\mathrm{slack}} & \boldsymbol{I}_{\mathrm{artif.}} & \boldsymbol{0} & \boldsymbol{b}_{\mathrm{ge}} \\ \boldsymbol{A}_{\mathrm{eq}} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I}_{\mathrm{artif.}} & \boldsymbol{b}_{\mathrm{eq}} \\ \hline \boldsymbol{c} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & 0 \\ \hline -\boldsymbol{\Sigma} & \boldsymbol{0} & \boldsymbol{1} & \boldsymbol{0} & \boldsymbol{0} & -\Sigma_0 \end{array}\right], \tag{5.5}$$

where $\boldsymbol{\Sigma}$ contains the summed elements from each column of $\boldsymbol{A}_{\text{ge}}$ and $\boldsymbol{A}_{\text{eq}}$. Similarly, Σ_0 contains the summed elements of $\boldsymbol{b}_{\text{ge}}$ and $\boldsymbol{b}_{\text{eq}}$. The previous objective function is now denoted as primary, while the last row resembles the secondary objective function which corresponds to a summation of the artificial variables.

The SLE, which is provided by (5.5), is obviously of canonical form due to the introduced artificial variables represented by $\boldsymbol{I}_{\text{artif.}}$ and due to the previously described slack variables $\boldsymbol{I}_{\text{slack}}$. Consequently, the simplex algorithm can be applied to find an optimal solution to the secondary objective function. Thereby, the primary objective function is ignored during the minimum ratio test, however, all row operations are performed on this row as well. At the end of the simplex algorithm, the secondary objective function reaches its minimum with 0, where no artificial variables are used. Consequently, they are not needed in order to determine the basic feasible solution. This allows to reduce the simplex tableau again by skipping the columns corresponding to artificial variables as well as the secondary objective function.

In a second phase, the simplex algorithm can be applied to the reduced simplex tableau and an optimization can be performed with respect to the primary objective function. The combination of both phases is commonly known as two-phase simplex method. There are also other methods available, e.g., the *Big M method* [DT97].

A sparse recovery problem can be expressed as LP in standard form as described in (4.6) on page 25. Since only equations are involved, the corresponding simplex tableau for the first phase is equal to [cf. (5.5)]

$$\left[\begin{array}{cc|c} \boldsymbol{A} & \boldsymbol{I} & \boldsymbol{b} \\ \hline \boldsymbol{c} & \boldsymbol{0} & 0 \\ \hline -\boldsymbol{\Sigma} & \boldsymbol{0} & -\Sigma_0 \end{array}\right].$$

For the element identification in (5.3), the parameters are $k = n + m$ and $l = m$, where the primary objective function is ignored for the identification.

In the subsequent section, the concept of sparsity awareness based on degeneracy is introduced. Thereby, the previously described simplex method is extended to be sparsity aware.

5.2. Sparse Solutions through Degeneracy

Since the BP is just a convex relaxation, the optimization of the derived LP is not the original goal but merely the chosen suboptimal approach. The solution space, which corresponds to the given under-determined system, is naturally the same for the sparse recovery problem as well as for the ℓ_1-minimization. As it has been previously mentioned, sparse solutions are found in degenerated vertices, which correspond by definition to the intersection of more than l hyperplanes. Therefore, it is only reasonable to look out for degenerated vertices during the walk on the surface described by the simplex algorithm.

For systems in general position (cf. Section 3.2.1 on page 20), such a degeneracy based approach is well legitimated: By extracting a vertex from a simplex tableau, only l components are not set to zero. Consequently, every solution corresponding to a vertex has at most l non-zero components. A degenerated vertex has at least one zero on the right hand side of the tableau, and therefore, the number of non-zero components is less than l. The degenerated vertex can be assumed to be unique for a linear system in general position since there exists with probability 1 no other solution of sparsity less than l (cf. Section 3.2.2).

In the following, two general approaches for the search of sparse solutions are provided. The first concept is based on the path which is chosen by the simplex algorithm, while the second is based on the closeness of the convex relaxation with respect to the sparse solution. Both concepts can also be combined and result in variants of SASAs.

5.2.1. Path Oriented Search Variants

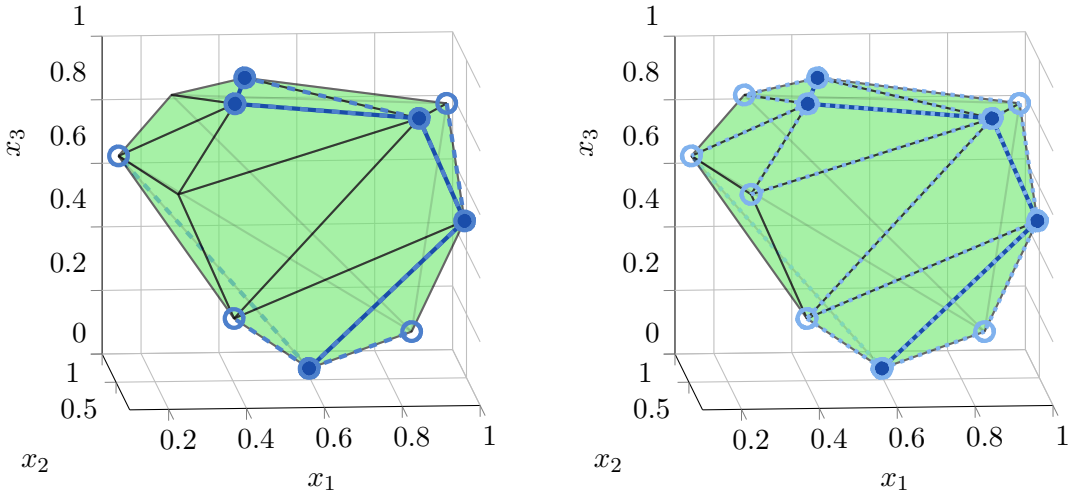
While the simplex algorithm walks from one vertex to another, the solution space can be explored for degenerated vertices up to different extends. Three exemplary variants of such searches along the path are provided in the following. Thereby, each variant represents a trade-off between potential reconstruction gain and complexity.

1. **Direct path:** This is the most straight forward variant. Each visited vertex on the path to the optimal solution is checked for degeneracy. If such a degenerated vertex is found, the obtained sparse solution is returned and the algorithm stops. In case there is no degenerated vertex found, the algorithm continues until the optimal solution of the LP is reached ($\alpha_i \geq 0 \ \forall \ 1 \leq i \leq k$). A vertex is checked for degeneracy by the minimum ratio test (5.4): As soon as $b_r/a_{r,c} = 0$ is obtained, a sparse solution is found. Since the minimum ratio test has to be performed anyway, no additional complexity is required for this variant. On the contrary, simplex steps might be even avoided because of an earlier termination, and consequently, computation time might be decreased.

2. **Improving neighbors:** In addition to the visited vertices, this variant checks all neighboring vertices which improve the current value of the objective function for degeneracy. As noted before, the corresponding columns are identified by $\alpha_i < 0$. The degeneracy is verified by searching for rows with equal ratios during the minimum ratio test for each improving column. In order to estimate the additional complexity, the number of improving columns has to be known. On average, this number can be roughly approximated to half the columns, which leads to $kl/2$ ratios which have to be calculated and compared. The additional computation time might be compensated by the increased probability for an earlier termination, if a sparse solution is found.

3. **All neighbors:** The second variant can be extended to check not only the improving but all neighboring vertices for degeneracy. On the first sight, it might sound implausible to consider non-improving vertices, however, it follows directly the general spirit of sparsity awareness: The objective function is defined by the ℓ_1-minimization, which is only the convex relaxation of the underlying sparse problem. Consequently, the sparsest solution can be found in a non-improving neighbor, as soon as it does not coincide with the convex relaxation. Since kl ratios have to be determined per simplex step, this variant comes with the highest complexity but has also an increased probability of finding the sparsest solution.

The difference of the three variants can be visualized by an exemplary polytope. Since the first variant examines only the vertices on the path, its visualization corresponds to Figure 5.1 on page 36. In addition to the vertices on the direct path, their neighbors are also checked during the second and third variant as depicted in Figure 5.2 for a minimization of $-x_3$. As it can be seen from the figure, some vertices might be checked several times during the third variant which may be avoided in corresponding implementations.



(a) Variant 2: Improving neighbors

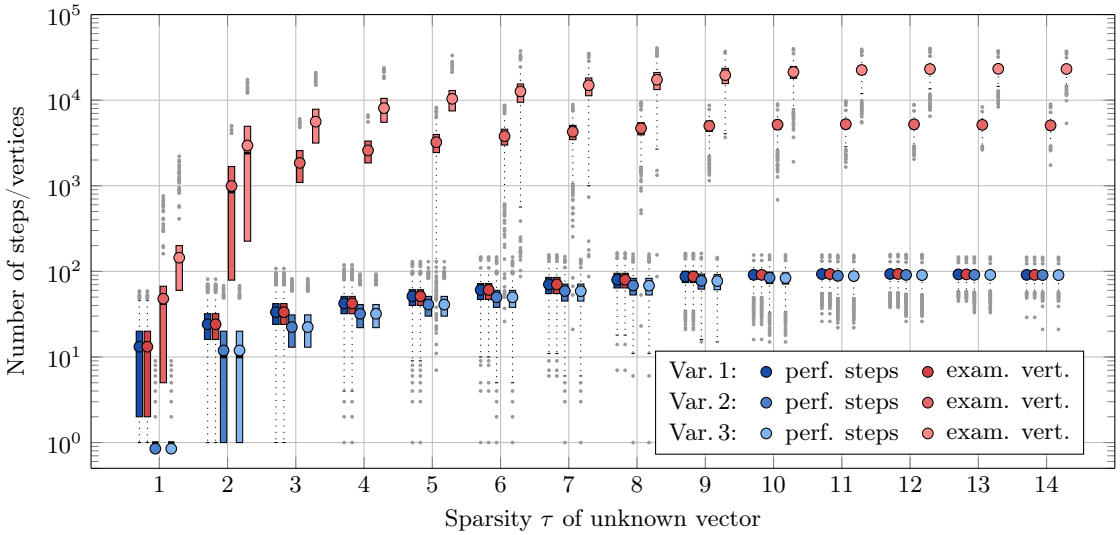
(b) Variant 3: All neighbors

Figure 5.2.: Visualization of path oriented search variants for sparsity aware simplex algorithms which examine neighboring vertices. The direct path (Variant 1) is given in dark blue (cf. Figure 5.1).

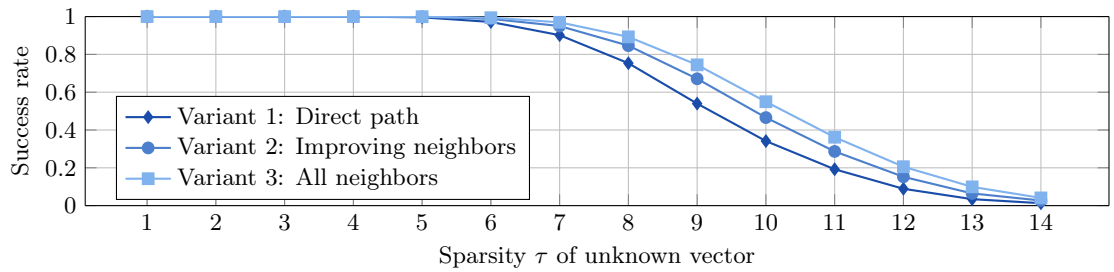
Further variants with different complexity trade-offs can be derived, for example by selecting all columns with $\alpha_i < \kappa \in \mathbb{R}$, where κ is a corresponding arbitrary threshold. Since the performance of such derived variants is bounded by the presented extremal cases, only these variants are considered for the remainder of this chapter.

A common parameter for all variants is the choice of the pivoting strategy: The rule of Bland is applied to avoid cycling in degenerated vertices and usually results in considerably longer paths. Since such an avoidance is not needed for the presented approach of sparsity awareness, the smallest reduced cost pivoting strategy is applied for the presented variants of sparsity aware simplex algorithms.

Within Figure 5.3, the proposed variants are evaluated. The complexity differences between the three provided search variants with respect to the number of performed simplex steps and examined vertices are visualized over the sparsity τ of the unknown vector for all variants in Figure 5.3a. Similarly, the success rates of determining the sparse solution are given in Figure 5.3b. For the underlying simulation, 10 000 realizations of CS-schemes with random sensing matrices of size 32×128, resulting in an LP with $n = 256$, $m = 32$, have been used. A reconstruction is considered to be successful here if the squared error of the underlying sparse vector is below 10^{-10}. Further simulation details can be found in Appendix H.2 on page 120. For the direct path search (Variant 1), there is only one vertex examined per simplex step, therefore, the numbers are equal. Since variants 2 and 3 examine their neighbors, these search approaches may terminate earlier, and therefore, their number of performed simplex steps is less than for the first variant, where the number of examined vertices is consequently significantly larger. Obviously, Variant 3 examines most vertices, since all neighbors are considered. Since a degenerated vertex might be found thereby, this Variant has the tendency to terminate earlier than Variant 2, as it can be seen at the outliers for sparsity $\tau = 9$. The success rate given in Figure 5.3b illustrates the performance gains



(a) Boxplots on the number of performed simplex steps (blue) and examined vertices (red)



(b) Success rate of the provided path oriented search variants

Figure 5.3.: Evaluating path oriented search variants. Simulation parameters: Appendix H.2.

of the presented path oriented search variants. Each of the given variants corresponds to a trade-off between complexity and improved reconstruction. The more vertices are visited and examined for sparse neighbors, the higher is the probability of finding a degenerated vertex which provides the sparsest solution.

5.2.2. Closeness of Convex Relaxation

The convex relaxation of BP is justified by the circumstance that the sparsest solution has also minimal ℓ_1-norm as long as it is sparse enough. As soon as the searched solution is not sufficiently sparse, the BP fails consequently. However, the desired degenerated vertex, which resembles the sparsest solution, might still be in the vicinity of the vertex corresponding to the smallest ℓ_1-norm. Searching the sparsest solution within the neighbors of the vertex obtained by BP resembles another approach to SASAs.

The underlying assumption can be investigated by applying several simplex steps to all columns of the corresponding simplex tableau. The obtained vertices are checked for degeneracy as described before. By recursively repeating this procedure for all vertices until the desired sparse solution is found, the distance d between both solutions can be determined

with respect to the number of necessary simplex steps (which equals the count of involved edges on the polytope). For a concrete CS-like scenario with sensing matrices of size 16×32 and a sparsity range of $1 \leq \tau \leq 8$, the closeness of the convex ℓ_1 relaxation within the solution space of the resulting LP with $n = 64$, $m = 16$ has been examined for cases where BP fails to obtain the sparsest solution. The corresponding distance distribution given in Table 5.1 has been determined over 10 000 random realizations of sensing matrices and sparse vectors. The corresponding simulation details can be found in Appendix H.2 on page 120. The sparsest solution coincides with the one of minimal ℓ_1-norm ($d = 0$) for sufficiently sparse

τ \ d	1	2	3	4	5
3	91.3%	8.7%	0	0	0
4	80.2%	17.6%	2.1%	0	0
5	69.5%	27.1%	3.3%	0	0
6	55.3%	34.7%	9.1%	0.9%	0
7	41.7%	39.0%	15.7%	3.2%	0.3%
8	29.2%	38.0%	22.9%	8.1%	1.6%

Table 5.1.: Relative distance distribution between the sparsest solution and the one with minimal ℓ_1-norm in case of a failed BP for an LP with $n = 64$, $m = 16$.

vectors ($\tau \leq 2$). As previously assumed, the resulting vertex is still located in the vicinity of the sparsest solution for sufficiently small values of non-zero components (cf. $3 < \tau \leq 7$). Consequently, examining the vicinity of the vertex determined by the BP for degeneracy is a potential approach for sparse recovery.

Checking the degeneracy of vertices up to different distances can be considered as variants of this closeness-based approach. As for the previously discussed schemes of sparsity awareness, such variants correspond to trade-offs between complexity and reconstruction performance. Since the number of neighboring vertices grows exponentially with the distance, only the direct neighbors ($d = 1$) are considered in the following.

The described approach to sparsity awareness, which is based on the closeness of the convex relaxation, can be combined with the previously introduced path oriented search variants. The resulting possibilities for a sparsity aware two-phase method are described subsequently in Section 5.3, where its potential for CS is evaluated as well.

5.3. Sparsity Aware Simplex Methods for Compressed Sensing

As described in Section 5.1.2 on page 38, the simplex method can be used for sparse recovery problems as they arise in the context of CS. Since degenerated vertices provide sparse solutions, corresponding search approaches were given within the preceding Section 5.2. In the following, variants of sparsity aware two-phase simplex methods are given which combine the described search strategies. Subsequently, the resulting sparsity aware simplex methods are evaluated for their potential in CS scenarios.

5.3.1. Sparsity Aware Two-Phase Simplex Method

The previously introduced variants of SASAs can be embedded into two-phase simplex methods. Thereby, one has to differ between first and second phase: In the first phase, a valid starting vertex is determined by the simplex algorithm. Since the corresponding objective function is not a convex relaxation of the sparse recovery problem, there is no point in examining non-improving vertices. This observation rules out the third variant of the path oriented search variants (all neighbors, cf. Section 5.2.1), as well as the search within the vicinity of the obtained optimum (cf. Section 5.2.2). The optimal vertex of the first phase is degenerated if the corresponding starting point for the second phase provides already a sparse solution itself, therefore, the first two path oriented search variants of Section 5.2.1 can be applied in the first phase as well. In case a degenerated vertex has been obtained within this first phase, its validity with respect to the reduced original problem $\boldsymbol{Ax} = \boldsymbol{b}$ has to be verified since artificial variables might have been used within this vertex. Even if a zero is assigned to an artificial variable, which is not implausible for a degenerated vertex, the resulting reduced system will be invalid. If the validity check is not passed, a conventional simplex algorithm can be applied in order to determine the starting vertex for the second phase.

Within the second phase, both search approaches can be applied: In case there is no degenerated vertex found during the (sparsity aware) simplex algorithm, the vicinity of the obtained vertex with minimal ℓ_1-norm can be checked for degenerated vertices. For the path oriented search variant in which all neighboring vertices are examined (Variant 3, cf. Section 5.2.1), a subsequent check of the direct vicinity is not necessary, since these neighbors have already been checked on the way.

The presented variants on sparsity aware two-phase simplex methods are illustrated as flowchart in Figure 5.4. The alternative path oriented search variants (cf. Section 5.2.1) are represented by blue boxes with rounded corners. In case a valid sparse solution is found in the first phase, the corresponding tableau is reduced and the path oriented SASAs return the result immediately in the second phase. As mentioned before, a conventional simplex algorithm has to determine a valid starting point for the second phase as soon as no valid sparse solution has been obtained. After the first phase has been completed, the three path oriented search variants can be utilized. In case a sparse solution is obtained therein, it is directly returned and the recovery procedure is completed. The green block represents the optional (dashed edges) examination of the vicinity close to the vertex representing the solution with minimal ℓ_1-norm, which may be performed if no sparse solution could be obtained by the path oriented search variants.

5.3.2. Potential of Sparsity Awareness in Compressed Sensing

The previously introduced two-phase sparsity aware simplex methods can basically be applied in every CS-scenario where the underlying BP is suitable. With the help of numerical simulations, the potential of sparsity awareness in CS is evaluated in the following.

The subsequently examined CS schemes are based on random sensing matrices of size 32×128 resulting in an LP with $m = 32$ equations and $n = 256$ unknowns. More details on the simulation can be found in Appendix H.2 on page 120. In Figure 5.5, the success rate of the sparse reconstruction is plotted over the sparsity of the unknown vector.[2] For

[2]For further comparison, the corresponding boxplots of the squared error are given in Appendix B on page 105.

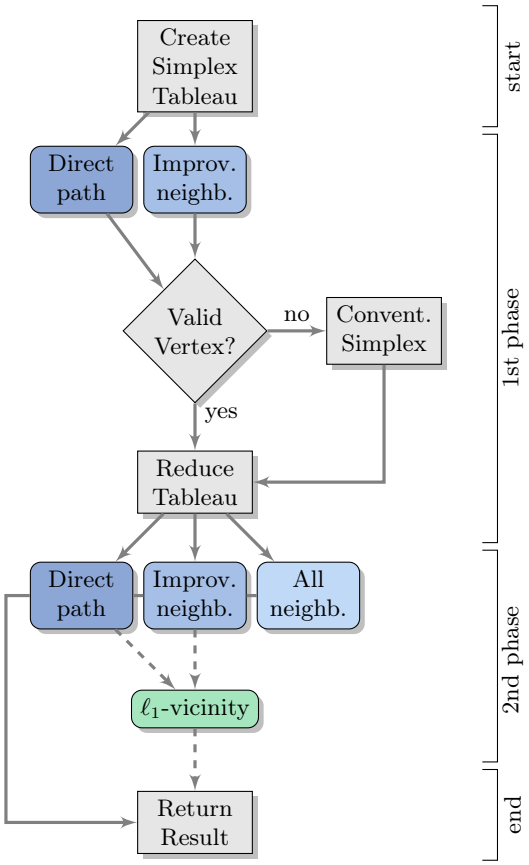


Figure 5.4.: Flowchart of the possible variants for the presented sparsity aware two-phase simplex method. Alternative path oriented search variants (cf. Section 5.2.1) are in blue color. The optional (dashed edges) examination of the ℓ_1-vicinity (cf. Section 5.2.2) is represented by the green block.

comparison, the performance of the original BP is added as well as for the greedy OMP (cf. the corresponding paragraph on page 26). For all sparsity aware simplex methods within these simulations, the search on the direct path (Variant 1, cf. Section 5.2.1) has been performed within the first phase. For the second phase, all three path oriented search variants have been examined together with the optional check of the ℓ_1-vicinity. As it can be seen in the figure, the sparsity aware simplex methods improve on the BP as expected. The first variant of examining the direct path alone comes only with small performance gains. The examination of the ℓ_1-vicinity comes always with a significant performance gain. With increasing number of examined vertices, the success rate improves as well, as it can be seen for path oriented search variants which check neighboring vertices for degeneracy (variants 2 and 3, cf. Section 5.2.1). Comparing the success rates of OMP with those of the other schemes, it can be observed that there are cases where the OMP fails earlier than the convex relaxation BP (sparsity $3 < \tau < 6$). Due to the sparsity awareness, the proposed variants are able to successfully reconstruct the sparse vector more often than OMP for sparsity levels up to $\tau < 10$.

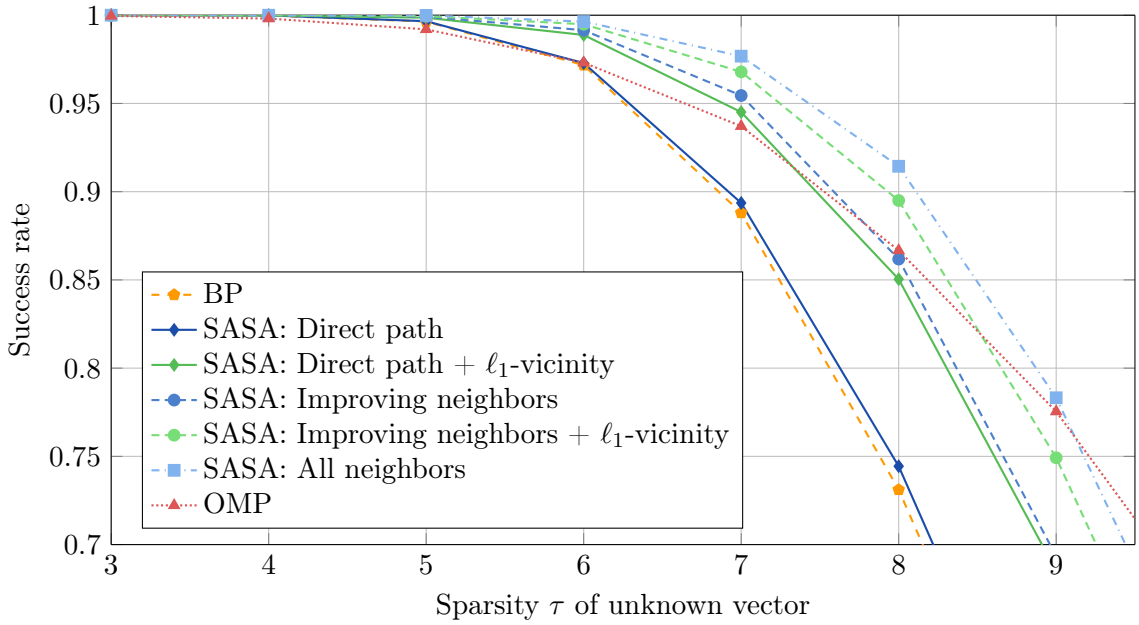


Figure 5.5.: Evaluation of sparsity aware simplex methods. Direct path search is used as SASA in each first phase. The SASA which is used in the second phase is given by the legend. BP (dashed orange) and OMP (dotted red) are given as reference. Simulation parameters: Appendix H.2.

5.4. Summary and Overview

The concept of SASAs has been proposed as contribution to CS in this chapter. Within Section 5.1, the well-known simplex algorithm and the two-phase method have been introduced, which allow to solve LPs with arbitrary linear inequality constraints. The simplex method can be used to apply BP to CS problems since the corresponding convex ℓ_1-optimization are transformed into an LP. As it is argued in Section 5.2, BP provides only a convex relaxation to the actual sparse recovery problem, which is known to have the same solution if the unknown vector is sparse enough. Consequently, if the original solution has not been sufficiently sparse, the optimization of the LP does not provide in the searched sparse solution. In order to approach this dilemma, degenerated vertices can be used which are unwanted in typical simplex schemes since they might result in stalling cycles. Often, anti-cycling strategies are applied which come with an increased complexity. In the proposed approach of sparsity awareness, degenerated vertices are actually searched since they provide the sparse solutions. Thus, anti-cycling strategies are not needed. Degenerated vertices can be searched by different ways: Path oriented variants have been provided which examine the vertices (and possibly their neighbors) on the way to the optimum for degeneracy. It is also possible to search the vicinity of the obtained ℓ_1-solution for degenerated vertices. The potential of each search variant has been evaluated by numerical simulations, whereby the individual complexity has been considered as well. Afterwards, it was shown in Section 5.3 that these search variants can be combined into sparsity aware two-phase simplex methods. Subsequently, the potential of sparsity awareness for CS has been evaluated by simulations.

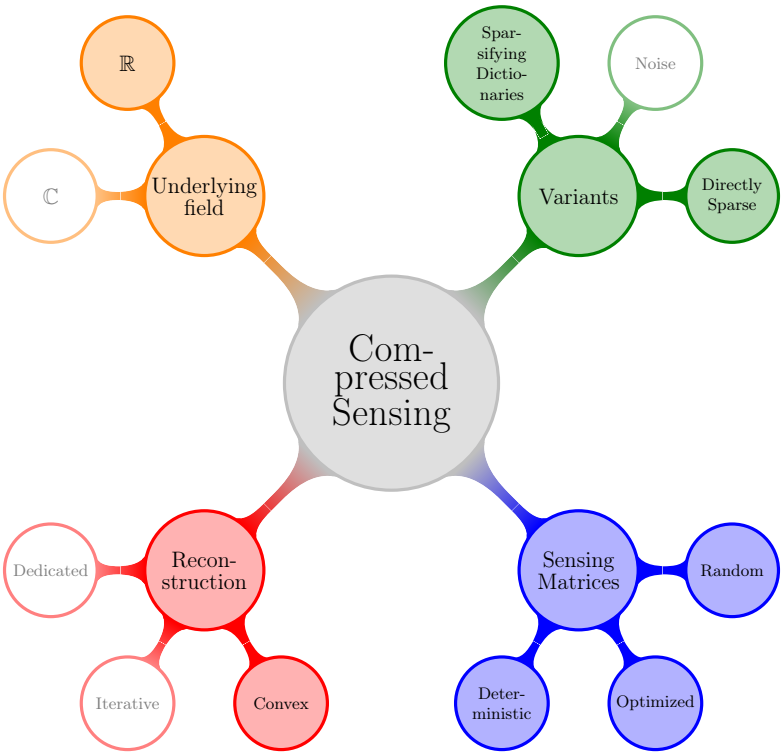


Figure 5.6.: Classification of this chapter's contents within the overall picture of Compressed Sensing (cf. Figure 4.1 on page 33).

The proposed approach of sparsity awareness resembles a proof of concept. Consequently, a basic variant of the simplex algorithm has been used for the given illustration, however, the fundamental concept of sparsity awareness can be principally adapted to more sophisticated simplex algorithms, e.g., the revised or the dual simplex algorithms [Lem54, Bea54, DT97].

In Figure 5.6, the efforts of the current chapter on sparsity aware simplex methods are put into the overall context of CS, cf. Figure 4.1. The methods provided within this chapter extend the conventional simplex method, therefore, they share the same possible applications within CS. For real-valued scenarios, the BP can be translated into an LP which can be solved by the (sparsity aware) simplex method. Within the numerical simulations, random matrices have been considered. However, the method does not depend on the used sensing matrix, as long as it is not ill-conditioned and the resulting system of linear equation is in general position. Elsewise, the uniqueness of a sufficiently sparse solution cannot be assured with high probability (cf. Section 3.2.2), and consequently, there are multiple degenerated vertices. Similarly, sparsifying dictionaries, although not explicitly discussed within this chapter, can be applied as long as the resulting sensing matrix is not ill-conditioned. Within this chapter, only directly sparse CS variants have been considered. Noise is not considered in this chapter, since the underlying BP itself would not be applied in noisy CS scenarios (cf. Section 4.1.1 on page 25). The provided reconstruction algorithms are based on the convex ℓ_1-relaxation, where the search for a degenerated vertex might be performed in a greedy way.

6

Coherence Optimization by Best Antipodal Spherical Codes

Within CS, the inter-column coherence of a sensing matrix is a popular property which is often used to provide performance guarantees for certain reconstruction algorithms or to assess the suitability of the matrix itself as previously introduced in Section 4.2.1 on page 27. Beyond the significance for CS, the coherence of vector sets is an important and limiting factor in many other applications, e.g., *Multiple-Input Multiple-Output* (MIMO) and *Code Division Multiple Access* (CDMA) wireless systems or non-orthogonal multi-pulse modulation [HMR+00, MV02, HJSP06, GD09, SJW09]. Consequently, there is considerable demand for vector sets with low coherence, and therefore, research in corresponding optimal constellations or optimization strategies is of general interest and not limited to the field of CS.

Coherence optimization is closely related to several other well known optimization problems, e.g., Grassmannian line packing, sphere packing, frame design and minimum distance optimization. Due to this circumstance, the overall field has already a considerable history [CHS96, LHJS03, FMT12]. Many approaches cover only the case of real vector spaces. Within this chapter, the more general complex case is mainly discussed and the special case of real vector spaces is explicitly derived. Since only a few very specific analytical (nearly) optimal solutions are known, there have been several different numerical approaches proposed in the last years [HMR+00, ARU01, XZG05, KCB07, DHJST08, SJW09, GD09, MD14]. Due to the very challenging optimization problem, there is still ongoing research. As shown in this chapter, *Best Antipodal Spherical Codes* (BASCs) result in vector sets of minimal coherence. Motivated by this observation, a corresponding concept for coherence minimization is proposed which is based on distance maximization of antipodal spherical codes.

In the following Section 6.1, the optimization problem is defined and corresponding theoretical bounds are given. Additionally, the problem of coherence optimization is connected to frame theory and Grassmannian line packing. Furthermore, a brief overview of existing optimization approaches is given. In Section 6.2 on page 55, the general concept of BASCs is introduced. Thereby, the equivalence of coherence and distance optimization is explicitly established and subsequently used for coherence optimization. The success of the proposed search approach is numerically evaluated and discussed. In Section 6.3 on page 68, the potential of the given BASC-based search approach within the field of CS is demonstrated for two CS variants.

6.1. Coherence of Vector Sets

As a uniqueness assuring property, the inter-column coherence has already been introduced in Section 4.2.1 on page 27 within the context of CS. In order to favor general applicability, the coherence is introduced for an arbitrary vector set $\mathcal{W}$ in the following. Such a set can generally be represented by a matrix $\boldsymbol{W} \in \mathbb{K}^{m\times n}$, where the n columns of dimension m correspond to the n vectors contained in $\mathcal{W}$. Thereby, $\mathbb{K}$ can be either $\mathbb{C}$ or $\mathbb{R}$ resulting in complex or real vector spaces. If not stated otherwise, $\mathbb{K} = \mathbb{C}$ is assumed in the following. Based on the introduced matrix notation, the coherence of $\boldsymbol{W}$ is defined as the maximal magnitude of all inner products between two different columns scaled by their corresponding Euclidean norms:

$$\mathrm{M}(\boldsymbol{W}) = \max_{i\neq j} \frac{|\langle \boldsymbol{w}_i, \boldsymbol{w}_j\rangle|}{\|\boldsymbol{w}_i\|\|\boldsymbol{w}_j\|}, \tag{6.1}$$

where $\boldsymbol{w}_i$ denotes the i-th column of $\boldsymbol{W}$. The normalization is often omitted in the context of matrices with columns of constant norm, which consequently results in another common definition of the coherence: $\mathrm{M}(\boldsymbol{W}) = \max_{i\neq j} |\langle \boldsymbol{w}_i, \boldsymbol{w}_j\rangle|$. By these definitions, the coherence depends only on the angle between the corresponding vectors and is not influenced by their individual norms.

As argued before, there are several applications demanding for vector sets with low coherence. The search for a minimum-coherence set $\mathcal{W}$ can be stated with the above mentioned matrix notation as optimization problem:

$$\min_{\boldsymbol{W}\in\mathbb{K}^{m\times n}} \mathrm{M}(\boldsymbol{W}), \tag{6.2}$$

where naturally only the non-trivial case of $m < n$ is of interest. Since the coherence is independent of the individual vector norms, it is sufficient to consider only normalized vector sets for the optimization.

The problem of coherence minimization is often considered in the literature and research. Therefore, the remainder of this section provides a brief overview. In Section 6.1.1, bounds on the achievable coherence are introduced and the connection to other research fields is established. Subsequently, existing approaches to coherence optimization are summarized in Section 6.1.2 on page 54 for later comparison.

6.1.1. Theory on Coherence Optimization

For important theoretical aspects of coherence optimization, an overview is given in the following. Therein, several bounds on the minimal achievable coherence are summarized for the non-trivial case of $m < n$. These bounds are later used to evaluate numerical optimization approaches. Subsequently, a connection between coherence minimization and related research fields is established.

Theoretical Bounds

As it is clear from its definition [cf. (6.1)], the coherence of a vector set is trivially upper bounded by 1 for the case of equal or antipodal vectors. However, lower bounds are not trivial at all, and thus, they are subject of several studies. Since the focus of this chapter is on coherence minimization, such lower bounds are naturally of immediate interest.

The most prominent lower bound on the coherence is the *Welch bound*:

$$\mathrm{M}(\boldsymbol{W}) \geq \sqrt{\frac{n-m}{m(n-1)}} \tag{6.3}$$

Actually, this bound was stated first for the real-valued case $\boldsymbol{W} \in \mathbb{R}^{m \times n}$ by Rankin [Ran56], and subsequently, for the complex-valued case by Welch [Wel74]. Thus, this bound is sometimes also called *Rankin* or *simplex bound* [CHS96].

Naturally, a vector set $\mathcal{W}$ is desired which achieves equality in (6.3). However, there is potential confusion in the literature: Two different criteria have been established which claim to determine whether a vector set satisfies the Welch bound with equality [Sar99]. In order to avoid any misleading ambiguity, both variants are provided in the following. Massey et al. [MM93] consider the *Root Mean Square* (RMS) magnitude of the inner product and denote vector sets as *Welch Bound Equality* (WBE) sequences for which

$$\sqrt{\frac{1}{n(n-1)} \sum_{k=1}^{n} \sum_{\substack{l=1 \\ l \neq k}}^{n} |\langle \boldsymbol{w}_k, \boldsymbol{w}_l \rangle|^2} = \sqrt{\frac{n-m}{m(n-1)}} \tag{6.4}$$

is fulfilled. This nomenclature introduced potential ambiguity, since equality in (6.3) is not necessarily implied by (6.4). In order to avoid further confusion with respect to the terms introduced in [MM93], Sarwate denoted vector sets, which achieve equality in (6.3), as *Maximum Welch Bound Equality* (MWBE) sequences [Sar99]. These vector sets form a subclass of previously described WBE sequences. In the case of MWBE sequences, the maximal inner product is considered for the classification as it is also used as optimization criterion within this chapter. Consequently, only MWBE sequences, which lead to vector sets with equality in (6.3), are considered in the following.

The Welch bound (6.3) cannot be achieved for too large vector sets: According to [DGS75], MWBE sequences can only exist if

$$\begin{aligned} n &\leq \frac{m(m+1)}{2} && \text{for } \boldsymbol{W} \in \mathbb{R}^{m \times n}, \\ n &\leq m^2 && \text{for } \boldsymbol{W} \in \mathbb{C}^{m \times n}. \end{aligned} \tag{6.5}$$

Thereby, it should be noted that these conditions are only necessary, and therefore, do not ensure the existence of MWBE sequences for the given range of vector sizes. In [CHS96], corresponding examples are shown, where equality in (6.3) cannot be achieved. Consequently, the Welch bound can only be met in certain combinations of number of vectors n and dimensionality m.

For large vector sets, which exhibit the limits of (6.5), the original Welch bound cannot be met with equality according to (6.5), consequently, such cases need to be considered by further bounds. For example, the *orthoplex bound* is given in [CHS96] for the real-valued case and has been subsequently extended to the complex-valued case in [Hen05, PTB11]:

$$\mathrm{M}(\boldsymbol{W}) \geq \sqrt{\frac{1}{m}} \tag{6.6}$$

As the Welch bound, the orthoplex bound (6.6) is also only achievable for a certain range of vectors n:

$$\begin{aligned} \frac{m(m+1)}{2} < n \leq (m-1)(m+2) \qquad & \text{for } \boldsymbol{W} \in \mathbb{R}^{m\times n} \\ m^2 < n \leq 2(m^2-1) \qquad & \text{for } \boldsymbol{W} \in \mathbb{C}^{m\times n} \end{aligned}$$

For the described case of large sets, another bound has been developed several decades before by Levenshtein et al. [LK78, Lev83, DY07]:

$$\begin{aligned} \mathrm{M}(\boldsymbol{W}) \geq \sqrt{\frac{3n-m^2-2m}{(m+2)(n-m)}} \qquad & \text{for } \boldsymbol{W} \in \mathbb{R}^{m\times n} \\ \mathrm{M}(\boldsymbol{W}) \geq \sqrt{\frac{2n-m^2-m}{(m+1)(n-m)}} \qquad & \text{for } \boldsymbol{W} \in \mathbb{C}^{m\times n} \end{aligned} \tag{6.7}$$

A further bound for this case has been derived in the 2000s by [MSEA03, XZG05] based on the maximal overlap radius of a beamformer codebook:

$$\mathrm{M}(\boldsymbol{W}) \geq 1 - 2n^{-\frac{1}{m-1}} \tag{6.8}$$

All lower bounds [Equations (6.3) and (6.6) to (6.8)] are relevant for the subsequent coherence minimization. By taking the maximum over all mentioned bounds within their corresponding regimes, the following composite lower bound can be derived for the complex-valued case:

$$\mathrm{M}(\boldsymbol{W}) \geq \begin{cases} \sqrt{\frac{n-m}{m(n-1)}} & \forall\ n \leq m^2 \\ \max\left\{\sqrt{\frac{2n-m^2-m}{(m+1)(n-m)}}, 1-2n^{-\frac{1}{m-1}}, \sqrt{\frac{1}{m}}\right\} & \forall\ m^2 < n \leq 2(m^2-1) \\ \max\left\{\sqrt{\frac{2n-m^2-m}{(m+1)(n-m)}}, 1-2n^{-\frac{1}{m-1}}\right\} & \forall\ 2(m^2-1) < n \end{cases} \tag{6.9}$$

This composite bound is subsequently used as reference to evaluate the performance of coherence minimization approaches.

Connections to Other Research Fields

The general problem of optimizing the coherence of vector sets can also be found in other research fields. In the following, prominent examples for such theoretical fields and their connection to coherence optimization are given.

Frame Theory The concept of *frame theory* has been introduced already in 1952 by Duffin et al. [DS52] and has been brought to new attention by Young and Daubechies et al. [You01, DGM86]. Since $m < n$, a *frame* can be considered as an overcomplete basis. Consequently, orthogonality between all vectors is not possible, and therefore, linear dependency cannot be avoided. Within this chapter, only finite-dimensional vector sets are of interest, and therefore, only *finite frames* are considered in the following. A vector set $\mathcal{W} = \{\boldsymbol{w}_i\}_1^n$ of n

vectors spanning $\mathbb{C}^m$ is denoted as frame if there exist two constants $0 < a \leq b < \infty$, such that for all $\boldsymbol{x} \in \mathbb{C}^m$

$$a\|\boldsymbol{x}\|^2 \leq \sum_{i=1}^{n} |\langle \boldsymbol{w}_i, \boldsymbol{x} \rangle|^2 \leq b\|\boldsymbol{x}\|^2,$$

where $a, b \in \mathbb{R}$ are the so called *frame bounds*. If $a = b$, the vector set $\mathcal{W}$ is denoted as *a-tight frame*. As a consequence, the rows of the corresponding matrix $\boldsymbol{W}$ are of equal-norm and orthogonal to each other. In case of unit columns ($\|\boldsymbol{w}_i\| = 1 \ \forall \ i \in [0, n]$), the vector set $\mathcal{W}$ is called a *unit norm frame*. Thus, a *Unit Norm Tight Frame* (UNTF) has necessarily a frame bound $a = n/m$ which is also known as frame redundancy. An *Equiangular Tight Frame* (ETF) has the additional property $|\langle \boldsymbol{w}_i, \boldsymbol{w}_j \rangle| = M \ \forall \ i \neq j$. By this definition, an ETF consists out of MWBE sequences. Detailed introductions to frame theory can be found in [KC07a, KC07b, CKP13]. In [SHJ03], frames minimizing the Welch bound (6.1) are defined as *Grassmannian frames*. Therefore, an ETF is an optimal Grassmannian frame by definition. For a given full-rank matrix $\boldsymbol{W}$, the closest a-tight frame $\boldsymbol{W}_{\mathrm{TF}}$ in Frobenius norm can be calculated by

$$\boldsymbol{W}_{\mathrm{TF}} = a(\boldsymbol{W}\boldsymbol{W}^{\mathrm{H}})^{-1/2}\boldsymbol{W} \tag{6.10}$$

according to [TDHJS05]. This relation can be used to obtain an a-tight frame which is close to a non-optimal Grassmannian frame which has been, for example, numerically obtained. In [BF03], the *frame potential*

$$\mathrm{FP}(\{\boldsymbol{w}_i\}_1^n) = \sum_{k=1}^{n}\sum_{l=1}^{n} |\langle \boldsymbol{w}_k, \boldsymbol{w}_l \rangle|^2$$

is introduced from a frame force, whereby this force is notably different from the forces subsequently defined in Section 6.2.2. Minimizing the frame potential results in a UNTF [BF03]. For these frames, $\mathrm{FP}(\{\boldsymbol{w}_i\}_1^n)$ equals (6.4) up to a constant factor, and thus, such a UNTF consists out of WBE sequences.

Grassmannian Line Packing The set of all l-dimensional subspaces of $\mathbb{C}^m$ (or $\mathbb{R}^m$ for the real-valued case) is denoted as *Grassmannian space* $\mathcal{G}(m, l)$ [CHS96]. The problem of finding the best packing of n l-dimensional subspaces in $\mathbb{C}^m$, with respect to some distance function, is commonly described as the Grassmannian subspace packing problem. Several distance functions are considered in the literature (e.g., the chordal or geodesic metric) [CHS96]. For the one-dimensional case of $l = 1$, which is also known as *Grassmannian line packing*, these different metrics lead to the same optimal solution if existing [DHJST08]. Such a perfect Grassmannian line packing corresponds to a Grassmannian frame (which motivated the name of these frames) [SHJ03].

Due to the presented relations, the search for ETFs, Grassmannian frames, or line packings corresponds to coherence minimization. Consequently, results in one of the research fields can be directly conveyed to the other fields. These relations emphasize the importance and generality of coherence optimization approaches. In the next section, existing approaches on coherence minimization are briefly summarized. Within Section 6.2, a numerical optimization approach is provided which is based on spherical codes.

6.1.2. Existing Approaches to Coherence Optimization

As described by (6.2) on page 50, the search for a vector set $\mathcal{W}$ with optimal coherence can be interpreted as optimization problem. The Welch bound cannot be obtained for all desired combinations of m and n, as it has been previously described in Section 6.1.1. Consequently, if such MWBE sequences do not exist (and therefore can not be achieved), vector sets with lowest possible coherence are naturally desired. Due to the importance for several research fields and applications, a variety of optimization algorithms and concrete constructions have been developed in order to solve (6.2) and to obtain vector sets with coherence as small as possible. Since the Welch and orthoplex bounds are only valid for a limited number of vectors [cf. Equations (6.5) and (6.6)], the composite lower bound (6.9) needs to be considered as reference in such cases.

In order to provide an overview and to assist reading the referred literature, direct solutions to the coherence optimization problem are briefly summarized in the following, as well as several common examples for analytical and numerical approaches. This summary is not intended to be exhaustive.

Analytical Approaches and Direct Solutions

For several specific dimensions m and numbers of vectors n, analytical approaches exist which are able to obtain MWBE sequences. See [CHS96] for a summary of methods and solutions for the real-valued case. Quite prominent examples are based on *conference matrices* $\boldsymbol{C}$ of size $l \times l$, which have zeros on their main diagonal and ± 1 on all other entries such that $\boldsymbol{C}^{\mathrm{T}}\boldsymbol{C} = (n-1)\boldsymbol{I}$ [vLS66, CHS96, HJSP03].

By extending an approach of [HMR+00], where rows of an IDFT matrices are selected in order to build $\boldsymbol{W}$, cyclic *difference sets* are used in [XZG05] to produce MWBE sequences. This idea is further extended to different types of difference sets in [Din06, DF07].

There are also approaches based on frame theory: For example, one approach utilizes so called *Steiner systems* in order to build sparse ETFs [FMT12]. As shown in [JMF14], a large class of these Steiner ETFs can be transformed into so called Kirkman ETFs for which all entries are of constant modulus.

Further types of analytical approaches are, for example, sequences based on *cosets* of certain codes (e.g. expurgated sets of Gold sequences) [MM93, Sar99] or the well-known method of *simplex signaling* [PS08].

For vector sets which cannot meet the Welch bound with equality according to (6.3), several analytical approaches exist which achieve a relatively low coherence. For example, the previously mentioned approaches based on difference sets can be extended to cyclotomic and almost difference sets [Din06, DF08, ZF12a, ZF12b, HW14]. This can be even further generalized by associated binary sequences [Yu12a, Yu12b, YFZ12].

For cases where the Welch bound cannot be achieved since the number of vectors n is too large [cf. (6.5)], *Mutually Unbiased Bases* (MUBs) are used in [WF89, DY07] to achieve the bound by Levenshtein (6.7). Such MUBs consist out of multiple m-dimensional bases for which the inner products between the elements of different bases have the same magnitude $1/\sqrt{m}$ [Sch60, WF89]. Consequently, the orthoplex bound (6.6) can also be achieved by MUBs. For example, the existence of MUBs with $m+1$ bases is shown in [WF89] for m being a prime power. In [DY07], this construction is further generalized and the corresponding

matrices are able to fulfill the bound by Levenshtein (6.7) and the orthoplex bound (6.6) with equality, which implies $n = m^2 + m$.

Further approaches for low-coherence vector sets are based on extended small Kasami codes or non-linear Kerdock codes [Sar99].

Numerical Approaches

As described before, optimal solutions are only known for certain dimensions m and numbers of vectors n if even existing. Therefore, numerical approaches are especially for cases of great interest, where optimal solutions are not known. In the last decades, many different algorithms have been published and the whole topic is still very active. Examples for such numerical approaches are not limited to:

- The authors of [HMR+00] propose row-pruning on DFT matrices based on random search strategies.
- Smooth approximations of the max operator, which is used in (6.2), are often applied such that an introduced free parameter can be utilized iteratively for a subsequent optimization [ARU01, GD09, MD14].
- Application of a generalized Lloyd algorithm in order to obtain a sphere vector quantizer which leads to low-coherence sets [XZG05].
- Within [KCB07], a non-linear map (exponential parametrization) is applied on space-time codes for coherent systems.
- Alternating projections are used in [DHJST08] to enforce alternately spectral and structural properties.
- In [SJW09], an expansion-compression algorithm is proposed, which alternately applies a max-min and a min-max optimization of the distance.
- Iterative decorrelation is applied by by a series of locally convex optimizations in [Rus13].
- The approach of [TKK14] combines shrinkage and matrix nearness [cf. (6.10)] with an optional averaging step.

Subsequently in Section 6.2, a numerical scheme is proposed which aims to obtain vector sets with minimal coherence by finding BASCs. Thereby, an approach for distance optimization is extended and applied to the problem of determining vector sets with low-coherence.

6.2. Best Antipodal Spherical Codes for Coherence Optimization

A spherical code $C_\mathrm{s}(m,n)$[1] corresponds to a set of n points placed on the surface of the m-dimensional unit sphere Ω_m which is centered at the origin of $\mathbb{K}^m$, where $\mathbb{K}$ can be either $\mathbb{C}$ or $\mathbb{R}$. This definition extends the real-valued variant from [CS99, EZ01, Slo] to the complex space. Any point of

$$C_\mathrm{s}(m,n) = \{\boldsymbol{s}_p\}_{p=1}^{n} \quad \text{with} \quad \boldsymbol{s}_p \in \mathbb{K}^m$$

[1] The suffix (m,n) may be skipped subsequently if it is of no further importance or clear from the context.

is commonly interpreted as codeword and determined by its position vector $\boldsymbol{s}_p$. Naturally, any set of n points can be equivalently described as $m \times n$ matrix comprising the corresponding position vectors as columns. Spherical codes, which maximize the minimal Euclidean distance $\mathrm{d}_{pq} = \|\boldsymbol{s}_p - \boldsymbol{s}_q\|$ with $p \neq q$ and $p, q \in [1, n]$ (or equivalently, minimize the maximal inner product of the corresponding vectors), are denoted as *Best Spherical Codes* (BSCs) $C_{\mathrm{bs}}(m, n)$. Usually, all rotations of a BSC are considered equivalent, consequently, a BSC is only characterized by its distance distribution.

In the following, antipodal spherical codes are defined with an additional equivalence property: Antipodal vectors are considered equivalent, which results in

$$\boldsymbol{s}_q \equiv -\boldsymbol{s}_q \in C_{\mathrm{ras}}(m, n) \quad \forall\ q \in [1, n] \tag{6.11}$$

for real-valued antipodal spherical codes $C_{\mathrm{ras}}(m, n)$ and in

$$\boldsymbol{s}_q \equiv \boldsymbol{s}_q \cdot \mathrm{e}^{\mathrm{j}\phi} \in C_{\mathrm{cas}}(m, n) \quad \forall\ \phi \in \mathbb{R},\ q \in [1, n] \tag{6.12}$$

for complex-valued antipodal spherical codes $C_{\mathrm{cas}}(m, n)$. As a consequence of these definitions, two equivalent vectors lie on the same (complex) line (cf. [LHJS03]). Every antipodal spherical code is obviously still a valid C_{s}. Naturally, a BASC, denoted by C_{bas}, maximizes the minimal Euclidean distance between all its vectors (considering the equivalence relation).

In the subsequent Section 6.2.1, the equivalence of coherence and distance optimization is shown for the case of antipodal spherical codes. In Section 6.2.2 on the facing page, an approach is proposed which searches for BASCs. By the aforementioned equivalence, this concept can be used to obtain low-coherence vector sets.

6.2.1. Equivalence of Coherence and Distance Optimization

The concept of utilizing BASCs as source for low-coherence vector sets is based on the equivalence of maximizing the minimal distance in an antipodal spherical code and minimizing the coherence of a vector set. It seems (cf. [LHJS03]) that this connection is commonly known also for the complex case[2], however, this connection is explicitly derived within a complex vector space in the following for the sake of completeness and illustration, since the subsequently proposed approach is based thereon.

The magnitude of the inner product is invariant to a phase-change of the corresponding vectors:

$$|\langle \boldsymbol{s}_p, \boldsymbol{s}_q \rangle| = \left| \left\langle \boldsymbol{s}_p, \boldsymbol{s}_q \cdot \mathrm{e}^{\mathrm{j}\phi} \right\rangle \right| \quad \forall\ \boldsymbol{s}_p, \boldsymbol{s}_q \in C_{\mathrm{s}},\ \phi \in \mathbb{R}$$

Consequently, it is sufficient to consider only complex antipodal spherical codes C_{cas} for the optimization:

$$\min_{C_{\mathrm{s}}} \max_{p \neq q} |\langle \boldsymbol{s}_p, \boldsymbol{s}_q \rangle| = \min_{C_{\mathrm{cas}}} \max_{p \neq q} |\langle \boldsymbol{s}_p, \boldsymbol{s}_q \rangle|$$

For these antipodal codes, the following equation can also be established due to the implied equivalence [cf. (6.12)]:

$$\mathrm{Re}\left(\langle \boldsymbol{s}_p, \boldsymbol{s}_q \rangle\right) \equiv \mathrm{Im}\left(\left\langle \boldsymbol{s}_p, \boldsymbol{s}_q \cdot \mathrm{e}^{\mathrm{j}\frac{\pi}{2}} \right\rangle\right) \equiv \mathrm{Im}\left(\langle \boldsymbol{s}_p, \boldsymbol{s}_q \rangle\right)$$

[2] For real vector spaces, the connection is trivial. However, a simple generalization is neither obvious nor given in [LHJS03] for the complex case.

From the squared absolute of the inner product $|\langle \boldsymbol{s}_p, \boldsymbol{s}_q \rangle|^2 = \mathrm{Re}^2\left(\langle \boldsymbol{s}_p, \boldsymbol{s}_q \rangle\right) + \mathrm{Im}^2\left(\langle \boldsymbol{s}_p, \boldsymbol{s}_q \rangle\right)$, it follows consequently:

$$|\langle \boldsymbol{s}_p, \boldsymbol{s}_q \rangle| \equiv \sqrt{2}\,\mathrm{Re}\left(\langle \boldsymbol{s}_p, \boldsymbol{s}_q \rangle\right) \tag{6.13}$$

Since $\boldsymbol{s}_p, \boldsymbol{s}_q \in C_\mathrm{s}$, only unit vectors are considered $\langle \boldsymbol{s}_p, \boldsymbol{s}_p \rangle = \langle \boldsymbol{s}_q, \boldsymbol{s}_q \rangle = 1$. This can be combined with the squared distance

$$\|\boldsymbol{s}_p - \boldsymbol{s}_q\|^2 = \langle \boldsymbol{s}_p, \boldsymbol{s}_p \rangle + \langle \boldsymbol{s}_q, \boldsymbol{s}_q \rangle - 2\,\mathrm{Re}\left(\langle \boldsymbol{s}_p, \boldsymbol{s}_q \rangle\right), \tag{6.14}$$

where it should be noted that the square function is monotonic for positive real values. The desired equivalence follows directly by inserting (6.13):

$$C_\mathrm{bcas} = \underset{C_\mathrm{cas}}{\mathrm{argmax}} \min_{p \neq q} \|\boldsymbol{s}_p - \boldsymbol{s}_q\| = \underset{C_\mathrm{s}}{\mathrm{argmin}} \max_{p \neq q} |\langle \boldsymbol{s}_p, \boldsymbol{s}_q \rangle|$$

Consequently, by determining a *Best Complex Antipodal Spherical Code* (BCASC) C_bcas, a spherical code is obtained which results in a complex matrix of minimal coherence.

As within this section, BASCs correspond to vector sets with minimal coherence. Based on this observation, numerical schemes for the search of BASCs can be used for coherence minimization. For such numerical schemes, it typically cannot be guaranteed that the global optimum is actually found. Consequently, it is of interest whether a near optimal solution for a maximal minimum distance optimization is also close to the optimum for coherence minimization: Due to the previously used definition in (6.14), a discrepancy in the distance contributes quadratically while the influence of the inner product is only linear. As a direct consequence, near optimal solutions of numerical approaches for BASCs will also result in close to optimal low-coherence vector sets.

6.2.2. Search for Best Antipodal Spherical Codes

Within this section, an approach is proposed which allows to obtain spherical codes close to BASCs for the cases of real and complex vector spaces. The given procedure is based on the method of [LSZ88] which numerically searches spherical codes close to BSCs in real vector spaces. In the following, the approach of [LSZ88] is summarized in order to provide a basis for the subsequent extension to complex vector spaces and to antipodal spherical codes.

Search for Best Spherical Codes

Often, physically motivated models are established for the optimization of spherical codes, as for example in [Lee57]. Thereby, the points of spherical codes are typically considered as n charged particles on the unit sphere. Consequently, these charges are subject to some field of repelling forces. Starting from an arbitrary initial position, these particles will move until the total potential energy of the system reaches a (local) minimum. Typically, there are multiple local minima. The particles will settle in any one of these which results in a stable or unstable equilibrium of mutual repelling forces. Depending on the nature of the assumed forces, there are principally multiple potential functions possible. In [Laz80], a generalized potential function $\mathrm{g}(C_\mathrm{s}(m,n))$ is introduced, for which a specific form is given in [LBK86]:

$$\mathrm{g}(C_\mathrm{s}(m,n)) = \sum_{p=1}^{n} \sum_{q=1}^{p-1} \|\boldsymbol{s}_p - \boldsymbol{s}_q\|^{-(\nu-2)}, \tag{6.15}$$

where $\nu \in \mathbb{N}$ with $\nu > 2$ is a free parameter. For this potential function, the global minimum of $\mathrm{g}(C_\mathrm{s}(m,n))$ is attained by a BSC in case of $\nu \to \infty$.

In order to minimize the generalized potential function under the unit radius constraint of the sphere, the method of Lagrange multipliers $\mathcal{L} = \{\lambda_p\}_{p=1}^{n}$ with $\lambda_p \in \mathbb{R}$ can be used. Thereby, the Lagrange function $\mathrm{G}(C_\mathrm{s}(N,M),\mathcal{L})$, which corresponds to the potential function (6.15) and the unit radius constraint, is given by

$$\mathrm{G}(C_\mathrm{s}(m,n),\mathcal{L}) = \mathrm{g}(C_\mathrm{s}(m,n)) + \sum_{p=1}^{n} \lambda_p \left(\|\boldsymbol{s}_p\|^2 - 1 \right). \tag{6.16}$$

The necessary conditions for a global minimum of the potential function (6.15) are consequently equal to

$$\frac{\partial \mathrm{G}(C_\mathrm{s}(m,n),\mathcal{L})}{\partial s_{p_u}} = 0 \quad \text{and} \quad \frac{\partial \mathrm{G}(C_\mathrm{s}(m,n),\mathcal{L})}{\partial \lambda_p} = 0 \quad \forall\, p \in [1,n]\,,\ u \in [1,m]\,,$$

where s_{p_u} is the u-th element of $\boldsymbol{s}_p$. As already derived in [LSZ88], these conditions can be expressed by the equilibrium:

$$\left\{ \boldsymbol{s}_p = \frac{\displaystyle\sum_{q \neq p} \left[(\boldsymbol{s}_p - \boldsymbol{s}_q) / \|\boldsymbol{s}_p - \boldsymbol{s}_q\|^\nu \right]}{\left\| \displaystyle\sum_{q \neq p} \left[(\boldsymbol{s}_p - \boldsymbol{s}_q) / \|\boldsymbol{s}_p - \boldsymbol{s}_q\|^\nu \right] \right\|} \right\}_{p=1}^{n} \tag{6.17}$$

In order to simplify expressions, the underlined denotation of unit vectors $\underline{\boldsymbol{s}} = \boldsymbol{s}/\|\boldsymbol{s}\|$ is used hereafter. Thereby, the equilibrium (6.17) can be represented by:

$$\left\{ \underline{\boldsymbol{s}}_p = \underline{\sum_{q \neq p} \frac{\underline{\boldsymbol{s}}_p - \underline{\boldsymbol{s}}_q}{\|\underline{\boldsymbol{s}}_p - \underline{\boldsymbol{s}}_q\|^\nu}} = \underline{\sum_{q \neq p} \boldsymbol{\delta}_{pq}} \right\}_{p=1}^{n} \tag{6.18}$$

The right side within (6.18) can be interpreted as a superposition of all effective forces which act on the corresponding code words of the spherical code: $\boldsymbol{f}_p = \sum_{q \neq p} \boldsymbol{\delta}_{pq}\ \forall\, p \in [1,n]$. Based on these forces, a mapping $\boldsymbol{P}$ can be introduced:

$$\boldsymbol{P}(C_\mathrm{s}(m,n)) = \left\{ \underline{\underline{\boldsymbol{s}}_p + \alpha \underline{\boldsymbol{f}}_p} \right\}_{p=1}^{n}, \tag{6.19}$$

with $\alpha \in \mathbb{R}$ being a "damping factor". For a small enough values of α, the iterative application of the mapping

$$C_\mathrm{s}(m,n)^{(k+1)} = \boldsymbol{P}(C_\mathrm{s}(m,n)^{(k)}), \quad k = 0, 1, \ldots \tag{6.20}$$

converges to one fixed point.

For ν large enough, it is already numerically inferred in [LSZ88] that these fixed points correspond to spherical codes whose minimal distances are close to those of BSCs. Consequently by finding a fixed point for the iterative process (6.20) with a sufficiently large ν, the corresponding spherical code should be close to the best one.

Within this section, the original approach of [LSZ88] for real vector spaces has been presented as basis for subsequent extensions. Subsequently, this approach is formally extended to complex vector spaces.

Search for Best Complex Spherical Codes

Subsequently, it is shown that a minimum of the generalized potential function $\mathrm{g}(C_\mathrm{s}(m,n))$ can be expressed by the equilibrium (6.18) also in the case of complex vector spaces. Based thereon, the approach of [LSZ88] is formally extended in the following for the search of *Best Complex Spherical Codes* (BCSCs).

The Lagrange function (6.16) is also valid for the case of complex vector spaces and is still real-valued. The same holds for the Lagrange multipliers $\mathcal{L}$ and the constraint functions $\left\{\|\boldsymbol{s}_p\|^2 - 1 = 0\right\}_{p=1}^{n}$. However, the necessary conditions for a global minimum of the potential function (6.15) are slightly different in a complex vector space:

$$\frac{\partial \mathrm{G}(C_\mathrm{s}(m,n),\mathcal{L})}{\partial s_{p_u}^{\mathrm{R}}} = 0, \quad \frac{\partial \mathrm{G}(C_\mathrm{s}(m,n),\mathcal{L})}{\partial s_{p_u}^{\mathrm{I}}} = 0 \quad \text{and} \quad \frac{\partial \mathrm{G}(C_\mathrm{s}(m,n),\mathcal{L})}{\partial \lambda_p} = 0, \tag{6.21}$$

for all $p = [1,n]$ and $u = [1,m]$.

The norm of a vector difference is subsequently denoted by $\delta_{pq} = \|\boldsymbol{s}_p - \boldsymbol{s}_q\|$ and the abbreviated functions $f(x) = x^{-(\nu-2)}$ and $\varphi(\boldsymbol{s}_p) = \|\boldsymbol{s}_p\|^2 - 1$ are introduced in order to simplify expressions. With the help of these abbreviations, the generalized potential function is provided by [cf. (6.15)]

$$\mathrm{g}(C_\mathrm{s}(m,n)) = \sum_{p=1}^{n}\sum_{q=1}^{p-1} f(\delta_{pq}),$$

which leads to an abbreviated Lagrange function:

$$\mathrm{G}(C_\mathrm{s}(m,n),\mathcal{L}) = \sum_{p=1}^{n}\left[\lambda_p\varphi(\boldsymbol{s}_p) + \sum_{q=1}^{p-1} f(\delta_{pq})\right]$$

With the help of the following derivations

$$\frac{\partial\varphi(\boldsymbol{s}_p)}{\partial s_{p_u}^{\mathrm{R}}} = 2s_{p_u}^{\mathrm{R}}, \quad \frac{\partial\delta_{pq}}{\partial s_{p_u}^{\mathrm{R}}} = \frac{s_{p_u}^{\mathrm{R}} - s_{q_u}^{\mathrm{R}}}{\delta_{pq}}, \quad \text{and} \quad \frac{\partial f(\delta_{pq})}{\partial\delta_{pq}} = \frac{-(\nu-2)}{\delta_{pq}^{\nu-1}},$$

the necessary condition for a minimum of the potential function, which considers the real-part $s_{p_u}^{\mathrm{R}}$ [cf. (6.21)], can be reformulated. Thereby, it should be noted that for the double sum $\sum_{p=1}^{n}\sum_{q<p}$, only summands need to be considered, which depend on p, since all other summands are canceled by the derivation. Due to $\delta_{pq} = \delta_{qp}$, these summands correspond to those of $\sum_{q\neq p}$:

$$\begin{aligned}
\frac{\partial \mathrm{G}(C_\mathrm{s}(m,n),\mathcal{L})}{\partial s_{p_u}^{\mathrm{R}}} =& \frac{\partial\lambda_p\cdot\varphi(\boldsymbol{s}_p)}{\partial s_{p_u}^{\mathrm{R}}} + \sum_{q\neq p}\frac{\partial f(\delta_{pq})}{\partial s_{p_u}^{\mathrm{R}}} = 0\\
=& 2s_{p_u}^{\mathrm{R}}\lambda_p + \sum_{q\neq p}\frac{\partial f(\delta_{pq})}{\partial\delta_{pq}}\cdot\frac{s_{p_u}^{\mathrm{R}} - s_{q_u}^{\mathrm{R}}}{\delta_{pq}}\\
=& 2s_{p_u}^{\mathrm{R}}\lambda_p - (\nu-2)\sum_{q\neq p}\frac{s_{p_u}^{\mathrm{R}} - s_{q_u}^{\mathrm{R}}}{\delta_{pq}^{\nu}}\\
\Rightarrow\ s_{p_u}^{\mathrm{R}} =& \frac{\nu-2}{2\lambda_p}\sum_{q\neq p}\frac{s_{p_u}^{\mathrm{R}} - s_{q_u}^{\mathrm{R}}}{\delta_{pq}^{\nu}}
\end{aligned}$$

The necessary condition for the imaginary-part $s_{p_u}^{\mathrm{I}}$ [cf. (6.21)] can be equivalently reformulated with the imaginary instead of the real part. The remaining necessary condition for λ_p can also be reformulated with the help of the previously introduced abbreviations and derivations:

$$\begin{aligned}
\frac{\partial \mathrm{G}(C_{\mathrm{s}}(m,n),\mathcal{L})}{\partial \lambda_p} =& \frac{\partial \lambda_p \cdot \varphi(\boldsymbol{s}_p)}{\partial \lambda_p} = \varphi(\boldsymbol{s}_p) = \|\boldsymbol{s}_p\|^2 - 1 = 0 \\
\Rightarrow 1 =& \sum_{u=1}^{m} {s_{p_u}^{\mathrm{R}}}^2 + {s_{p_u}^{\mathrm{I}}}^2 \\
=& \sum_{u=1}^{m} \left(\frac{\nu-2}{2\lambda_p} \sum_{q\neq p} \frac{s_{p_u}^{\mathrm{R}} - s_{q_u}^{\mathrm{R}}}{\delta_{pq}^{\nu}} \right)^2 + \left(\frac{\nu-2}{2\lambda_p} \sum_{q\neq p} \frac{s_{p_u}^{\mathrm{I}} - s_{q_u}^{\mathrm{I}}}{\delta_{pq}^{\nu}} \right)^2 \\
=& \frac{(\nu-2)^2}{4\lambda_p^2} \sum_{u=1}^{m} \left(\sum_{q\neq p} \frac{(s_{p_u}^{\mathrm{R}} - s_{q_u}^{\mathrm{R}})}{\delta_{pq}^{\nu}} \right)^2 + \left(\sum_{q\neq p} \frac{(s_{p_u}^{\mathrm{I}} - s_{q_u}^{\mathrm{I}})}{\delta_{pq}^{\nu}} \right)^2 \\
=& \frac{(\nu-2)^2}{4\lambda_p^2} \left\| \sum_{q\neq p} \frac{\boldsymbol{s}_p - \boldsymbol{s}_q}{\delta_{pq}^{\nu}} \right\|^2 \\
\Rightarrow \frac{2\lambda_p}{\nu-2} =& \left\| \sum_{q\neq p} \frac{\boldsymbol{s}_p - \boldsymbol{s}_q}{\delta_{pq}^{\nu}} \right\|
\end{aligned}$$

By combining the above reformulations of the necessary conditions, the equilibrium of a codeword element can be given:

$$\begin{aligned}
s_{p_u} =& s_{p_u}^{\mathrm{R}} + \mathrm{j} \cdot s_{p_u}^{\mathrm{I}} \\
=& \frac{\nu-2}{2\lambda_p} \sum_{q\neq p} \frac{(s_{p_u}^{\mathrm{R}} - s_{q_u}^{\mathrm{R}})}{\delta_{pq}^{\nu}} + \mathrm{j} \cdot \frac{(s_{p_u}^{\mathrm{I}} - s_{q_u}^{\mathrm{I}})}{\delta_{pq}^{\nu}} \\
=& \frac{\sum\limits_{q\neq p} (s_{p_u} - s_{q_u})/\delta_{pq}^{\nu}}{\left\| \sum\limits_{q\neq p} (\boldsymbol{s}_p - \boldsymbol{s}_q)/\delta_{pq}^{\nu} \right\|}
\end{aligned}$$

These results can be combined to vectors as for the equilibrium of (6.18). Consequently, a minimum of the generalized potential function $\mathrm{g}(C_{\mathrm{s}}(m,n))$ can be expressed by the equilibrium (6.18) also in the case of complex vector spaces.

The above derivation allows to utilize the iterative process, which has been previously described for real vector spaces [cf. Equations (6.18) to (6.20)], also for the case of a complex vector space. Based on this result, the search for BCASCs is described in the subsequent section, which can be used for coherence minimization.

Search for Best (Complex) Antipodal Spherical Codes

Within Section 6.2.1, the equivalence of coherence and distance optimization has been established for antipodal spherical codes. In order to apply the previously described approaches for the search of BSCs and BCSCs to antipodal spherical codes, additional restrictions have to be considered during the calculation of the accumulated forces.

Calculation of Accumulated Forces The previously described antipodal equivalence (cf. Equations (6.11) and (6.12) on page 56) needs to be resembled such that the absolute value in the definition of the coherence (6.1) is accounted for. This can be achieved during the calculation of $\underline{\boldsymbol{f}}_p$ for the used mapping (6.19). In real vector spaces, this is accomplished by

$$\left\{ \underline{\boldsymbol{f}}_p = \underline{\sum_{q \neq p} \frac{\underline{\boldsymbol{s}}_p - \underline{\boldsymbol{s}}_q}{\|\underline{\boldsymbol{s}}_p - \underline{\boldsymbol{s}}_q\|^\nu} + \frac{\underline{\boldsymbol{s}}_p + \underline{\boldsymbol{s}}_q}{\|\underline{\boldsymbol{s}}_p + \underline{\boldsymbol{s}}_q\|^\nu}} \right\}_{p=1}^{n}, \tag{6.22}$$

where the antipodals can be thought of as additional vectors. In contrast to the real-valued case, a complex factor of $\exp[\mathrm{j}\phi]$ has to be considered for the complex-valued case [cf. (6.12)]. Previously, additional equivalent points have been used for the antipodals in the real vector space. However, an infinite number of such additional points would be needed since the phase ϕ in (6.12) is continuous. For a first approximation, a finite number of K distinct points can be evaluated, which is generated by $\exp[\mathrm{j}2\pi k/K] \; \forall \; k \in [1, K]$:

$$\left\{ \underline{\boldsymbol{f}}_p = \underline{\sum_{k=1}^{K} \sum_{q \neq p} \frac{\underline{\boldsymbol{s}}_p - \underline{\boldsymbol{s}}_q \mathrm{e}^{\mathrm{j}2\pi \frac{k}{K}}}{\left\|\underline{\boldsymbol{s}}_p - \underline{\boldsymbol{s}}_q \mathrm{e}^{\mathrm{j}2\pi \frac{k}{K}}\right\|^\nu}} \right\}_{p=1}^{n} \tag{6.23}$$

In order to have a reasonable numerical approximation, the number of additional equivalent points K must be sufficiently large. In case of $K \to \infty$, the inserted sum over k in (6.23) is transformed into an integral:

$$\left\{ \underline{\boldsymbol{f}}_p = \underline{\int_{\kappa=0}^{2\pi} \sum_{q \neq p} \frac{\underline{\boldsymbol{s}}_p - \underline{\boldsymbol{s}}_q \mathrm{e}^{\mathrm{j}\kappa}}{\left\|\underline{\boldsymbol{s}}_p - \underline{\boldsymbol{s}}_q \mathrm{e}^{\mathrm{j}\kappa}\right\|^\nu} \mathrm{d}\kappa} \right\}_{p=1}^{n} \tag{6.24}$$

However, because of the norm in the denominator, this integral is hard to solve analytically (if this is even possible) and numerical integration must be used in consequence.[3]

Best (complex) spherical codes can be searched with the help of the previously described mapping (6.19) in combination with the iterative process (6.20) and superimposed forces [cf. Equations (6.22) to (6.24)]. Depending on whether the real or complex vector space is considered, the described method is denoted as BASC or BCASC search approach. Thereby, it is emphasized that these methods are not guaranteed to obtain optimal solutions. An algorithmic description of the proposed search approach is given in the next section.

[3] Details on the numerical integration, which has been used within this thesis, are given in Appendix H.3.

Algorithm 6.1: Iterative BCASC search approach

Input : $\left\{\underline{\boldsymbol{s}}_p\right\}_{p=1}^{n}$ /* initial spherical code of n codewords of in m dimensions */

Output: C_s /* optimized spherical code */

1 $\alpha_\mathrm{init} \leftarrow 0.9$, $\epsilon \leftarrow 10^{-4}$ /* exemplary numerical parameters */
2 $\nu \leftarrow 2$, $\nu_\mathrm{max} \leftarrow 2^{10}$
3 $i_\mathrm{max} \leftarrow 10^5$, $\alpha \leftarrow \alpha_\mathrm{init}$
4 **while** $\nu < \nu_\mathrm{max}$ **do**
5 FixedPoint $\leftarrow$ **false** /* initialize indicator */
6 $i \leftarrow 0$ /* initialize iteration counter */
7 **while** $i < i_\mathrm{max}$ **and** FixedPoint $=$ **false do**
8 **for** $p \leftarrow 1$ **to** n **do** /* for each vector */
9 $\boldsymbol{f}_p \leftarrow \int_{\kappa=0}^{2\pi} \sum_{q \neq p} \frac{\underline{\boldsymbol{s}}_p - \underline{\boldsymbol{s}}_q \mathrm{e}^{\mathrm{j}\kappa}}{\left\|\underline{\boldsymbol{s}}_p - \underline{\boldsymbol{s}}_q \mathrm{e}^{\mathrm{j}\kappa}\right\|^\nu} d\kappa$ /* calculate superimposed forces */
10 $\{\boldsymbol{s}_p\}_{p=1}^{n} \leftarrow \left\{\underline{\boldsymbol{s}}_p + \alpha \underline{\boldsymbol{f}}_p\right\}_{p=1}^{n}$ /* apply accumulated forces */
11 **if** $\left\|\underline{\boldsymbol{f}}_p - \underline{\boldsymbol{s}}_p\right\| < \epsilon \ \forall\ p \in [1, n]$ **then** /* check for fixed point */
12 FixedPoint $\leftarrow$ **true** /* stop loop and proceed */
13 $i \leftarrow i + 1$
14 $\nu \leftarrow 2\nu$ /* adjust free parameter */
15 $\alpha \leftarrow \frac{\alpha_\mathrm{init}}{\nu - 1}$ /* adjust damping factor */
16 **return** $C_\mathrm{s} \leftarrow \left\{\underline{\boldsymbol{s}}_p\right\}_{p=1}^{n}$ /* return obtained spherical code */

Algorithmic Description The iterative application of the earlier described mapping (6.19) is proposed in [LBK86, LSZ88] as minimal distance optimization algorithm. Based thereon, an algorithmic description for the described BCASC search approach is given in Algorithm 6.1. Therein, empirically determined exemplary numerical parameters are provided as well. Starting with a small initial value for the free parameter ν, a fixed point is determined by iteratively applying the aforementioned mapping. Thereby, a fixed point is assumed if the norm on the difference between the normalized accumulated force $\underline{\boldsymbol{f}}_p$ and current vector $\underline{\boldsymbol{s}}_p$ is less than some threshold ϵ. This is reasonable, since in case the fixed point is found, the effective tangential forces vanish and the resulting accumulated force $\underline{\boldsymbol{f}}_p$ points in the same direction as $\underline{\boldsymbol{s}}_p$. If no fixed point is reached within i_max iterations, the algorithm proceeds as in the case of an obtained fixed point in order to avoid stalling. As soon as a fixed point is obtained (or if the maximum number of iterations is reached), the free parameter ν is increased and the damping factor α is reduced. The reduction of α follows a numerically inferred rule-of-thumb which has been obtained by [LBK86]. The overall procedure is repeated until a sufficiently large free-parameter ν_max has been reached. As bed in [LBK86, LSZ88], the procedure of increasing ν and decreasing α promotes convergence.

The presented algorithmic description of a BCASC search can be transformed into further search approaches by exchanging the formula used to calculate the accumulated forces (Line 8 in Algorithm 6.1): For example, by inserting (6.22), BASCs can be searched.

Evaluation with Respect to Coherence Optimization The previously presented BCASC search approach for coherence optimization is evaluated in the following. Since the proposed approach is based on random seeds and typically results in some local optimum, ten random seeds have been used for optimization and the vector set with the lowest coherence has been selected subsequently. As reference for comparison, the composite lower bound (6.9) is considered which includes also the Welch bound. Furthermore, the approach of [MD14], for which the source code can be found at [Med], is also considered for comparison. Thereby, the set with minimal coherence out of ten runs is considered as well. The range of coherence values, which had been obtained within the ten runs, is given as vertical bars in the subsequent figures in order to provide a measure for the obtained range of coherences. Further simulation details are given in Appendix H.3.

For $m = 3$ dimensions, a comparison of the obtained coherence values is given in Figure 6.1. As it can be seen in the figure, the BCASC search approach provides coherence values which

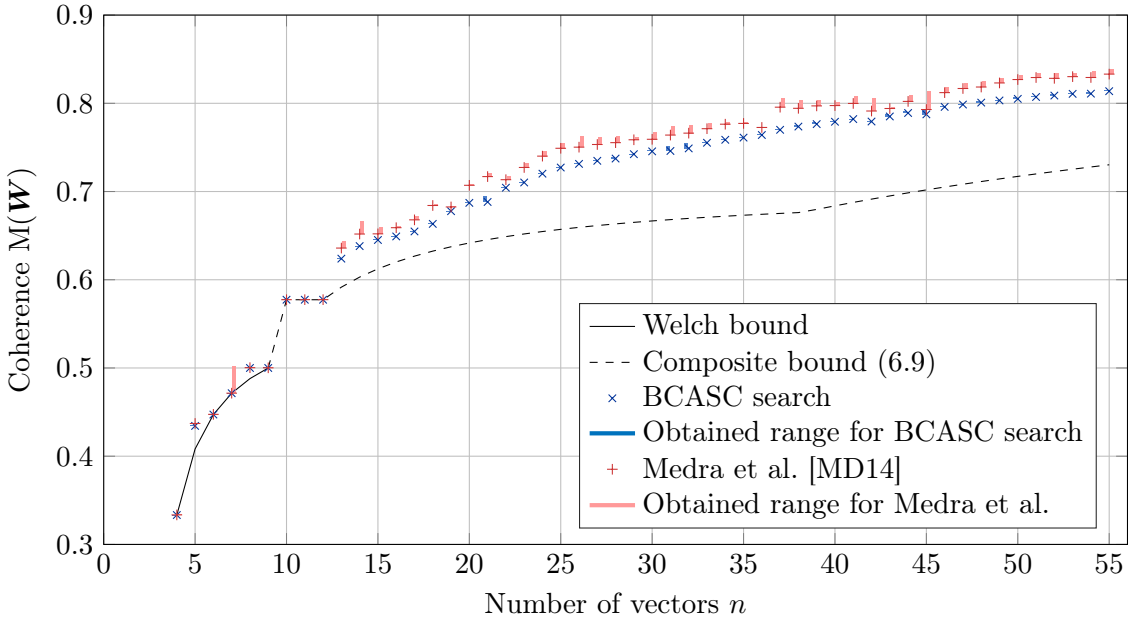
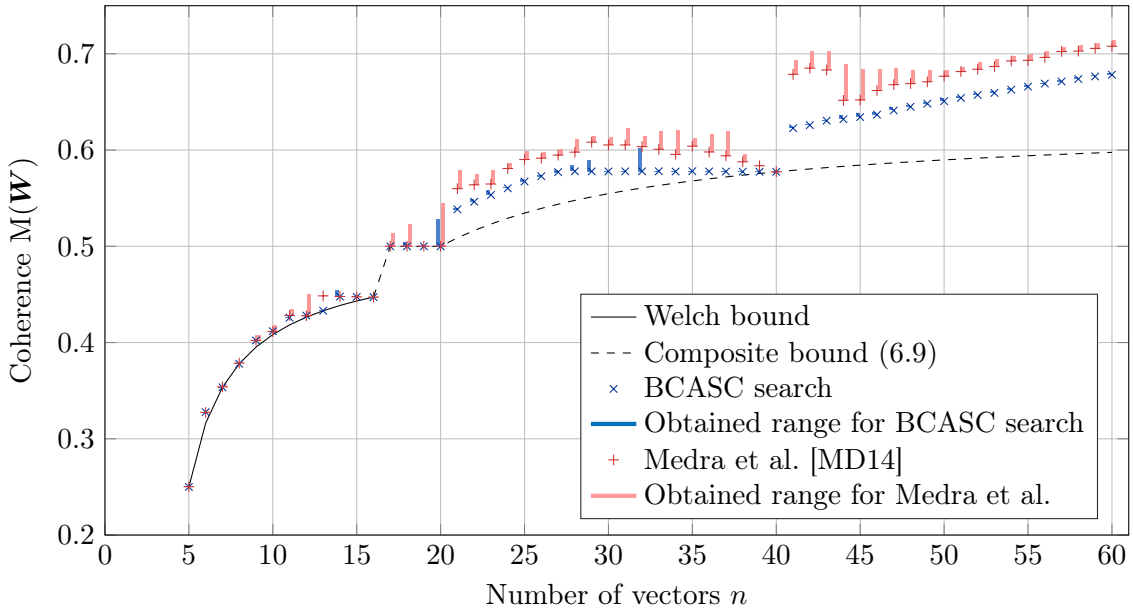
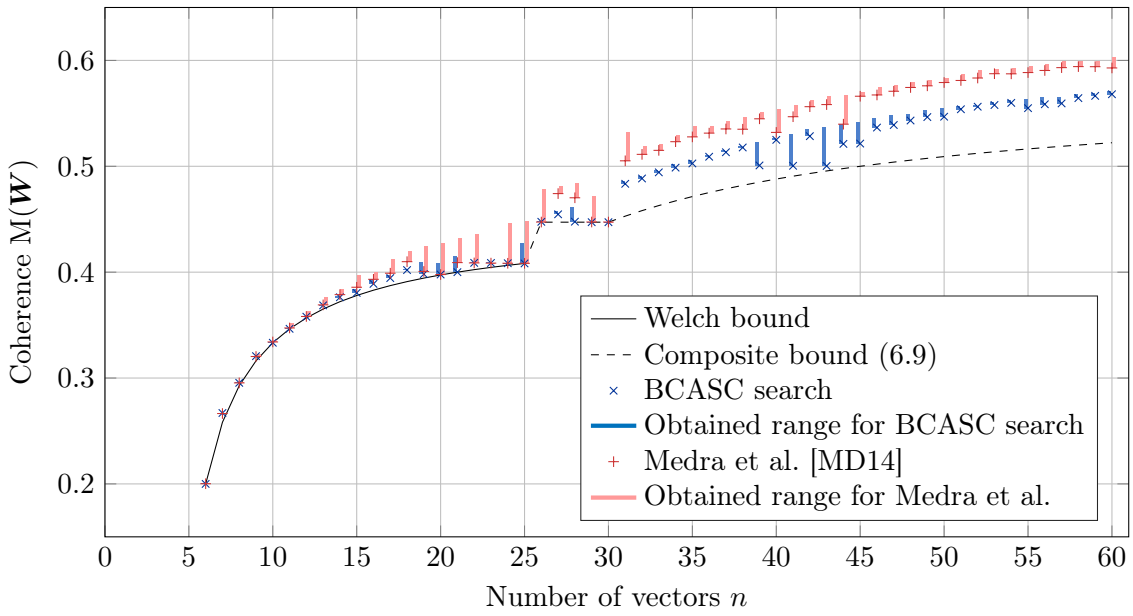


Figure 6.1.: Best coherence out of ten runs for varying number of vectors n in $m = 3$ dimensions. Vertical bars indicate the range of obtained coherence values. The lower bound is drawn solid if the Welch bound (6.5) is fulfilled.

meet the Welch bound with equality or are very close to it as long as the Welch bound can be met according to (6.5) ($n \leq 9 = n^2$). For $n \leq 12$, the BCASC search method and the approach by [MD14] obtain vector sets with almost identical coherence. Especially the quasi-constant coherence level, which is obtained by both methods for the range $10 \leq n \leq 12$, is remarkable. It corresponds to the orthoplex bound (6.6) and can be also observed in other dimensions, as subsequently shown, where this effect is further discussed, as soon as results for higher dimensions are presented. For larger vector sets, the proposed BCASC obtains generally smaller coherence levels. As it can be seen from the vertical bars in the figure, the coherence values of the obtained vector sets are quite stable. Consequently, most of the found (local) optima result in similar coherence values. Starting from $n \geq 13$, the orthoplex bound



(a) $m = 4$



(b) $m = 5$

Figure 6.2.: Best coherence out of ten runs for varying number of vectors n in m dimensions. Vertical bars indicate the range of obtained coherence values. The lower bound is drawn solid if the Welch bound (6.5) is fulfilled.

is replaced by (6.7) as active lower bound in (6.9). For even larger vector sets, the bound of (6.8) is dominating for $n \geq 39$. As in [XZG05], the lower bound (6.8) gets tighter for larger vector sets.

Further illustrations for dimensions $m = 4$ and 5 are given in Figure 6.2. The coherence values of the vector sets, which have been obtained by both methods, are almost equivalent for $n \leq 20$ and 30 respectively. Thereby, the achieved coherence values are more stable within the ten runs in case of the BCASC search approach, as indicated by the vertical bars. There is a level of constant coherence observable for $n > m^2$ as it has been observable for $m = 3$. After this plateau, the BCASC search approach always provides vector sets with smaller coherence than the method of [MD14]. As previously mentioned, this described plateau corresponds to the orthoplex bound (6.6). Since it is limited by (6.7), its length is equal to the individual dimensionality m for the mentioned cases. Such an almost constant level can also be observed for $m = 2$ (cf. [XZG05, Fig. 3]). The $n = m^2 + m$ vectors of each obtained set for $m = 3, 4$ and 5 can be put into a matrix $\boldsymbol{W}$ such that the corresponding Gram matrix $\boldsymbol{G} = \boldsymbol{W}^{\mathrm{H}}\boldsymbol{W}$ has a block diagonal structure. Within this structure, there are $m + 1$ identity matrices of dimension m on the diagonal while all other entries are of constant modulus. Consequently, the vector sets of each diagonal block correspond to an orthonormal basis for an m-dimensional subspace. Thus, the obtained solutions correspond to Grassmannian subspace packings $\mathcal{G}(m, m)$ of $m + 1$ subspaces (cf. corresponding paragraph on page 53). These vector sets correspond also to the previously mentioned constructions of MUBs given in [WF89, DY07]. For higher dimensions, cf. Appendix C on page 107, such a plateau-like level for $m^2 < n < m^2 + m$ has not been found by the described methods. Other plateaus can generally be found implicitly: In case, a vector set, which has been obtained by optimization, has a smaller coherence but larger cardinality n compared to some other set, arbitrary vectors can be removed while the low coherence remains the same. Consequently, sets with lower cardinality but same coherence can be derived therefrom, which outperform the previously known sets with respect to coherence. For example, there is a plateau for $35 \leq n \leq 43$ implicitly given by the vector set of $n = 43$ in Figure 6.2b. Plateaus of almost constant coherence usually precede the optimal constellation of $n = m^2$. Increasing dimensionality favors the vector sets obtained by BCASC search approach also for small number of vectors ($n < m^2$), as it can be seen for $m = 6, \cdots, 10$ in Appendix C.

In [XZG05, Table II], the performance of numerical coherence optimization approaches is compared. Based thereon, the updated Table 6.1 is extended by the results corresponding to the proposed BCASC search approach. With exception of the setting with $n = 16$ vectors

Table 6.1.: Comparison of numerical search algorithms as in [XZG05]

m	n	BCASC search	Medra et al. [MD14]	Love [Lov]	Xia et al. [XZG05]	Composite bound (6.9)
2	8	0.7950	0.7997	0.8415	0.8216	0.7500
3	16	0.6491	0.6590	0.8079	0.6766	0.6202
4	16	**0.4472**	0.4473	0.7525	0.4514	0.4472
4	64	0.6869	0.7151	0.7973	0.7447	0.6000

of $m = 4$ dimensions, the Welch bound cannot be obtained since $n > m^2$ [cf. (6.5)]. For this case, the BCASC search approach reaches the Welch bound (bold). For the other cases, vector sets with lowest coherence are provided by the BCASC search approach.

Similarly to the previous table, the results of [DHJST08] are compared with respect to the

Table 6.2.: Comparison of numerical search algorithms as in [MD14]

m	n	BCASC search	Medra et al. [MD14]	Dhilon et al. [DHJST08]	Composite bound (6.9)
4	5	**0.2500**	0.2502	0.2500	0.2500
4	6	0.3277	0.3274	0.3275	0.3162
4	7	**0.3536**	0.3540	0.3536	0.3536
4	8	**0.3780**	0.3787	0.3782	0.3780
4	9	0.4022	0.4021	0.4034	0.3953
4	10	0.4118	0.4113	0.4114	0.4082
4	16	**0.4472**	0.4473	0.4473	0.4472
4	20	**0.5000**	0.5001	0.5335	0.5000
5	6	**0.2000**	0.2002	0.2001	0.2000
5	7	0.2670	0.2665	0.2669	0.2582
5	8	0.2955	0.2954	0.2955	0.2928
5	9	0.3207	0.3203	0.3216	0.3162
5	10	**0.3333**	0.3341	0.3336	0.3333
5	16	0.3889	0.3932	0.3959	0.3830

coherence in Table 6.2 as in [MD14, Tab. II]. It can be seen from the table that the BCASC approach reaches the composite bound (6.9) most often (bold). The method of [MD14] obtained slightly better results in cases where the bound could not be reached, however, as discussed earlier, this behavior changes significantly in favor of the BCASC search approach for larger numbers of vectors n. In general, the algorithm of [DHJST08] comes with the worst performance within this comparison. With exception of the setting with $n = 20$ vectors in $m = 4$ dimensions, there are no significant negative outliers for [DHJST08].

Naturally, it depends on the needed level of coherence and the available computational resources which numerical optimization algorithm should be used. The BCASC search approach is computational costly especially for large n and m. The next section treats implementation aspects and approximations which might reduce the needed computation time.

Implementation Aspects and Approximations The BCASC search approach comes with a considerable computational effort. Due to its iterative nature, parallelization is only possible during the calculation of the accumulated forces $\{\boldsymbol{f}_p\}_{p=1}^n$, which results in n parallel computational threads. However, the resulting overhead is considerable compared to the short running time of the individual threads.

The magnitude of the damping factor α offers a more promising potential speedup. As described earlier, a sufficiently small value of α is necessary for convergence [cf. (6.19)]. However, small values of α result also in slow convergence. This situation can be coped with adaptively determining the values of α on a per force basis: In case the direction of the

force $\underline{\boldsymbol{f}}_p$, which acts on $\underline{\boldsymbol{s}}_p$, has not changed from one iteration to another, the corresponding value of α_p is increased by a factor until a predefined maximum is reached. This approach comes with an additional advantage: Even smaller values for α can be used in the beginning, which leads to preciser solutions.

For the BCASC search approach, numerically solving the integral in (6.24) in each iterative step of the algorithm is potentially the computationally most expensive part. Consequently, there might be interest in less complex approximations. As given previously in (6.23), the integral can be approximated by summing over K points. The coherence of vector sets obtained by this simple approximation is considered in the following by increasing the number of summands K. The best coherence out of ten runs is plotted in Figure 6.3 for the constellations given in Table 6.1. By dotted lines, the result of an optimization with an elaborate numerical

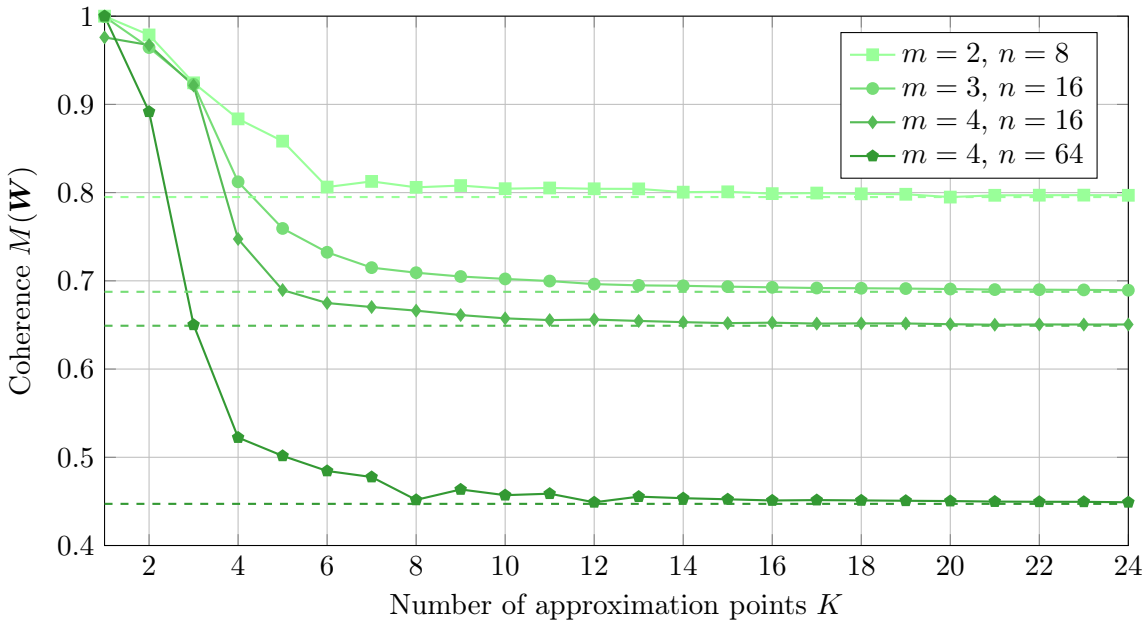


Figure 6.3.: Obtained coherence over the number of approximation points K. Dotted lines indicate the results of an elaborate numerical integration.

integration is indicated as reference. As expected, the coherence converges with increasing K. Thereby, the convergence is not monotonic (cf. $K = 8$ in Figure 6.3 for $m = 4$ and $n = 64$). This effect is caused by the uniform point distribution on the unit circle implied by K. The gaps between the K evaluation points influence the optimization, and consequently, favor vector sets with increased or reduced coherence. For $K = 22$, the difference between the numerical integration and the K-point approximation is surveyed by Table 6.3 in more detail. The effect of the K-point approximation with respect to the obtained coherence and corresponding running time (in seconds) can be observed within this table. As mentioned before, the BCASC search approach is able to reach the Welch bound for the case of $n = 16$ vectors in $m = 4$ dimensions, while the approximation, which is based on $K = 22$ points, is not able reach the Welch bound and takes even more time for the optimization. For the other cases however, the K-point approximation is generally, as expected, significantly faster. This advantage increases severely as the dimensionality m or the number of vectors n increase.

Table 6.3.: Comparison of running time and coherence between elaborate numerical integration and K-point approximation. Detailed parameters: Appendix H.3

		Numerical integration		Summation	
N	M	Coherence	Time [s]	Coherence	Time [s]
2	8	0.7950	699.03	0.7971	419.47
3	16	0.6491	2 903.86	0.6506	1 132.97
4	16	**0.4472**	27.26	0.4496	89.28
4	64	0.6869	64 720.95	0.6892	3 627.88

Again, this is caused by the point distribution on the unit circle implied by K: Especially in settings, where the actual solution is easily found (all ten runs achieved the Welch bound, cf. Figure 6.2a for $m = 4$ and $n = 16$), the gaps between the K points mislead the optimization. Generally, the coherence values obtained by the K-point approximation are slightly worse than for the case with numerical integration. With exception of the Welch bound achieving case $n = 16$ vectors in $m = 4$ dimensions (bold), the obtained coherence is still lower than for the other numerical approaches given in Table 6.1. In consequence, the presented K-point approximation might be a suitable alternative especially for settings with many vectors where thy existing bounds on the coherence are typically not achieved.

In case of the K-point approximation, the complexity for one iteration can be calculated: Since n codewords interact with each other and for K distinct evaluation points, whereby the norm for each of the m vector elements is calculated, the complexity of one iteration for a fixed ν scales asymptotically with $\mathcal{O}(m^2n^2K)$.

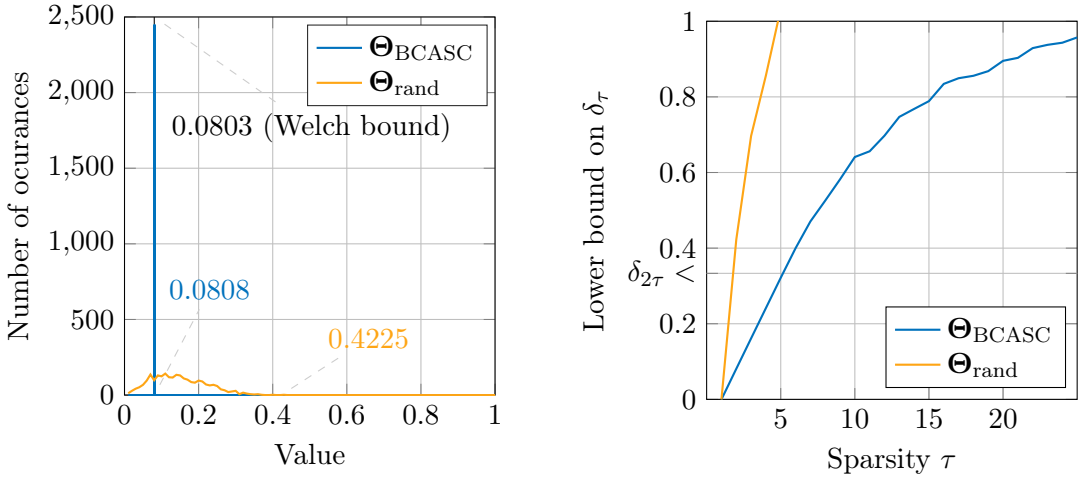
6.3. Applications in Compressed Sensing

The potential applications of the introduced coherence minimization approach are not limited to the previously mentioned examples of MIMO, CDMA, optical communication systems [HMR+00, HJSP06, GD09, SJW09, WF15], or CS. In the following, the described search for best (complex) antipodal spherical codes is used exemplary for two problems arising in the context of CS (cf. Section 4.4). The most straight forward utilization for the optimization of the sensing matrix is introduced in Section 6.3.1. The BASC search approach is used to adapt the measurement matrix with respect to a given dictionary in Section 6.3.2 on page 70.

6.3.1. Sensing Matrix Optimization

As aforementioned, the sensing matrix is naturally a crucial part of CS systems, and consequently, several properties have been proposed to determine the suitability of a given matrix, cf. Section 4.2 on page 27. Since the coherence, cf. Section 4.2.1, is efficiently verifiable, it is often considered as criterion for the optimization of sensing matrices. This circumstance provides the motivation for the B(C)ASC search based coherence optimization technique described in Section 6.2. Therefore, it is only straightforward to utilize the proposed search approach for sensing matrix optimization in the following.

The n codewords of a spherical code $C_\mathrm{s}(m, n)$ in an m-dimensional complex space are used as columns for a sensing matrix $\boldsymbol{\Theta}_\mathrm{BCASC} \in \mathbb{C}^{m\times n}$ in the exemplary case of $n = 38$ and $m = 50$. The distribution of $|\langle\boldsymbol{\theta}_i,\boldsymbol{\theta}_j\rangle|/\|\boldsymbol{\theta}_i\|\|\boldsymbol{\theta}_j\|$ $\forall\ i \neq j$ and $i, j \in [1, n]$ can also be used to evaluate the similarity of the columns. In Figure 6.4a, such a distribution is shown together with the Welch bound for the case of the optimized matrix. For reference, a corresponding distribution is given



(a) Distribution of $\frac{|\langle\boldsymbol{\theta}_i,\boldsymbol{\theta}_j\rangle|}{\|\boldsymbol{\theta}_i\|\|\boldsymbol{\theta}_j\|}$

(b) Lower bound on RIC δ_τ

Figure 6.4.: Evaluation of the given sensing matrix optimization with respect to the uniqueness assuring properties (cf. Section 4.2). Detailed parameters: Appendix H.3.

also for the case of a random matrix $\boldsymbol{\Theta}_\mathrm{rand}$. As bed in Section 4.2.2, the direct evaluation of $\boldsymbol{\Theta}$ for the RIP is not tractable, but a lower bound on the RIC δ_τ can be given by Monte-Carlo experiments. In Figure 6.4b, corresponding bounds are given for the described matrices, whereby 10^7 permutations per sparsity have been used for the Monte-Carlo experiments, cf. Section 4.2.2. The proposed search approach obtained a sensing matrix, where the absolute inner products are very focused resulting in a coherence $\mathrm{M}(\boldsymbol{\Theta}_\mathrm{BCASC}) = 0.0808$ which is very close to the Welch bound ($\mathrm{M}(\boldsymbol{W}) \geq 0.0803$). In comparison, the distribution of the random matrix is wide spread and results in a coherence of $\mathrm{M}(\boldsymbol{\Theta}_\mathrm{rand}) = 0.4225$. Also the lower bound on the RIC suggests the advantages of the optimized sensing matrix: According to the RIP (4.12), BP is able to recover τ-sparse signals up to $\tau \leq 5/2$ in case of the optimized matrix, where the random matrix is only able to recover 1-sparse signals.

The reconstruction performance of a low-coherence sensing matrix is evaluated in Figure 6.5 by combined boxplots for a noiseless CS scheme with respect to the achieved squared error $\|\boldsymbol{\chi} - \widehat{\boldsymbol{\chi}}\|^2$. As it can be seen in the figure, the optimized sensing matrix clearly outperforms the random reference. Already with a sparsity of $\tau = 13$, there are reconstruction failures in case of the random matrix, while the optimized sensing matrix is able to reconstruct successfully for $\tau < 21$. As the median indicates, the majority of sparse vectors are reconstructed up to $\tau = 25$ for the case of the random matrix, where the BCASC based variant is able to reconstruct the majority of vectors until $\tau = 30$. This result underlines the pessimistic character of RIP guarantees, where unique decoding with BP is only guaranteed up to $\tau = 1$ and 2 at best for the case of given random and BCASC matrices.

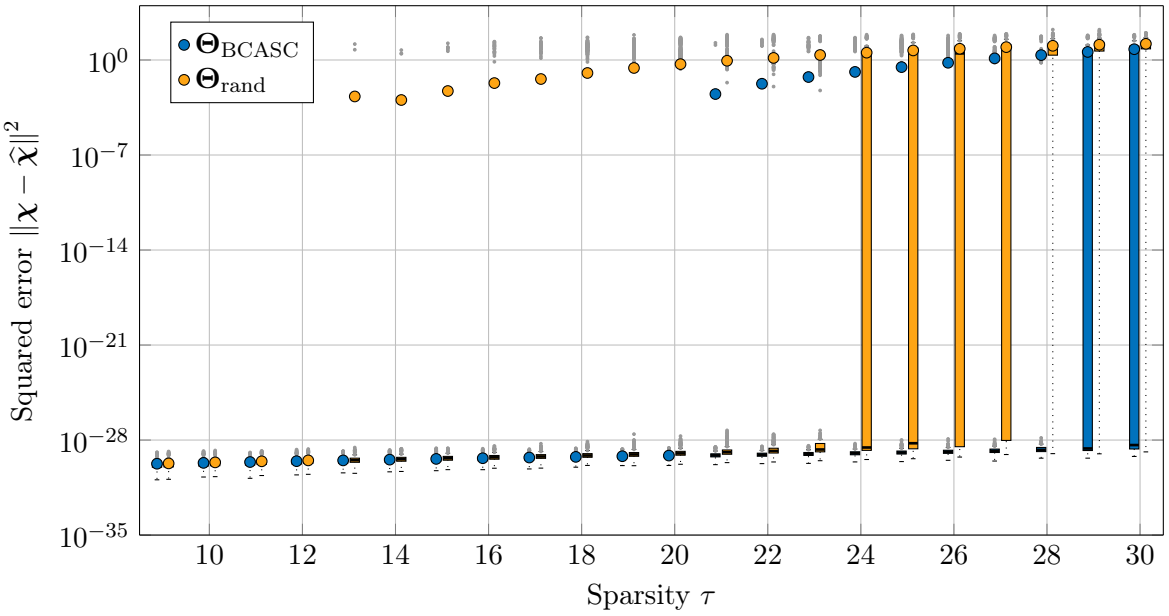


Figure 6.5.: Boxplots illustrating the squared error in CS schemes with an OMP reconstruction: An optimized low-coherence sensing matrix (blue) versus a random matrix (yellow). Simulation parameters: Appendix H.3.

The provided CS scenario is noiseless in order to illustrate the influence of the coherence on the reconstruction. The same optimized matrix is used as reference within a noisy CS scheme in Figures 7.5 and 7.8 on page 88 and on page 95.

Within this exemplary evaluation, the complex-valued case of BCASC search based sensing matrices has been discussed. Additionally, a real-valued low-coherence sensing matrix is used as bounding reference within Section 6.3.2.

6.3.2. Measurement Matrix Adaptation

Beside the straight forward optimization of sensing matrices, the introduced coherence minimization approach can also be used in the context of CS schemes with a given sparsifying dictionary $\boldsymbol{\Psi}$, cf. Section 4.4.2 on page 32. Within such scenarios, $\boldsymbol{\Psi}$ is assumed to be fixed. Consequently, only the measurement matrix $\boldsymbol{\Phi}$ remains as free parameter for an optimization. As it has been the motivation for the sensing matrix optimization in the previous section, most approaches aim to determine $\boldsymbol{\Phi}$ such that the coherence of the resulting sensing matrix $\mathrm{M}(\boldsymbol{\Theta}) = \mathrm{M}(\boldsymbol{\Phi}\boldsymbol{\Psi})$ is minimized, e.g., [Ela07, DCS09, AFMS10, ZFLR14]. As described in Section 4.4.2, it is argued in [CR07, CW08] that the rows of $\boldsymbol{\Phi}$ should be incoherent to the columns of $\boldsymbol{\Psi}$ resulting in minimal $\mathrm{M}(\boldsymbol{\Phi}, \boldsymbol{\Psi})$, cf. (4.14) on page 32. Since it is only reasonable that the measurements themselves are also incoherent, the intra row coherence of $\boldsymbol{\Phi}$ should be optimized as well which excludes trivial equal measurement solutions. Within this section, it is numerically investigated which optimization criterion is likely to result in more suitable sensing matrices.

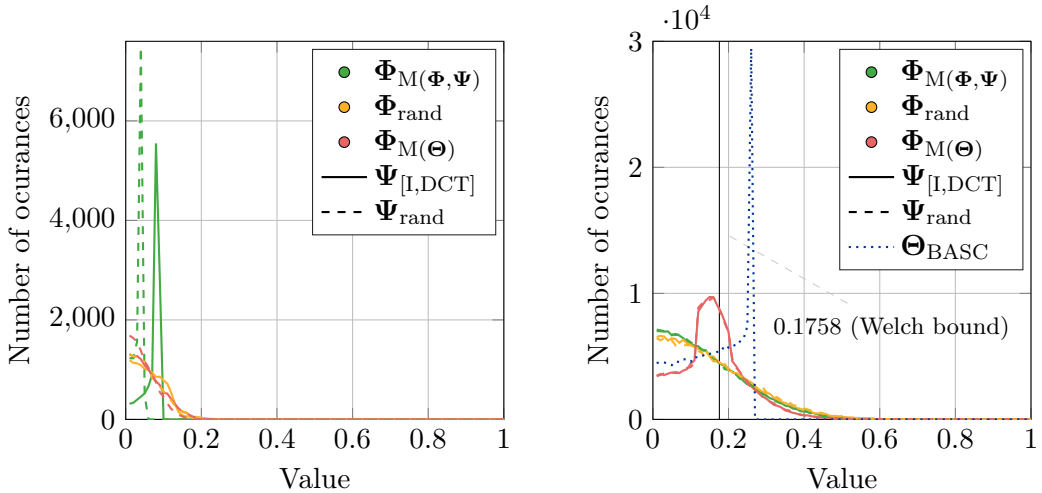
The proposed coherence optimization approach can be adapted such that $\mathrm{M}(\boldsymbol{\Phi}, \boldsymbol{\Psi})$ is optimized for a given dictionary $\boldsymbol{\Psi}$. Thereby, the rows $\boldsymbol{\phi}_i$ of $\boldsymbol{\Phi}$ correspond to codewords of

a spherical code C_s. The dictionary is considered by adding its columns $\boldsymbol{\psi}_i$ as fixed codewords into C_s. Consequently, the superimposed forces are calculated over all codewords but affect only those corresponding to rows of $\boldsymbol{\Phi}$. The remainder of the previously described algorithm is not changed. An algorithmic description for the overall approach is provided in Appendix E on page 113 for a complex-valued scenario. The description can be adapted to real vector spaces, as for the general search approach, by exchanging (6.24) on page 61 with (6.22) in Line 8 of Appendix E. A measurement matrix, which is obtained by the given search approach, is subsequently denoted by $\boldsymbol{\Phi}_{\mathrm{M}(\boldsymbol{\Phi},\boldsymbol{\Psi})}$.

The existing contributions to measurement matrix adaptation are typically limited to real vector spaces. In order to provide a basis for comparisons, this case is also considered in the following evaluation. Since it is the most prominent example, the M($\boldsymbol{\Theta}$)-optimization approach of [Ela07] is used in the following as reference[4], whereby the obtained measurement matrix is denoted by $\boldsymbol{\Phi}_{\mathrm{M}(\boldsymbol{\Theta})}$. Naturally, a BASC based sensing matrix can be considered as non-achievable bound on the reconstruction performance of this approach, since it is not limited by the fixed dictionary.

In the following, the performance of the two optimization approaches are further investigated for $\boldsymbol{\Phi} \in \mathbb{R}^{m \times l}$ with $\boldsymbol{\Psi} \in \mathbb{R}^{l \times n}$. The performance of random measurement matrices is considered as well for reference. As given dictionaries, a concatenation of the identity matrix and its *Discrete Cosine Transform* (DCT) $\boldsymbol{\Psi}_{[\mathrm{I,DCT}]}$ has been considered as well as column normalized random matrix $\boldsymbol{\Psi}_\mathrm{rand}$ of corresponding size thats elements are drawn from a standard normal distribution.

In order to compare the optimization approaches and to evaluate their success, the distributions of $|\langle \boldsymbol{\phi}_i, \boldsymbol{\psi}_j \rangle| / \|\boldsymbol{\phi}_i\| \|\boldsymbol{\psi}_j\|$ $\forall\, i \in [1, m]$ and $j \in [1, n]$ and $|\langle \boldsymbol{\theta}_i, \boldsymbol{\theta}_j \rangle| / \|\boldsymbol{\theta}_i\| \|\boldsymbol{\theta}_j\|$ $\forall\, i \neq j$ and $i, j \in [1, n]$ are given in Figure 6.6 for $m = 30$, $l = 200$ and $n = 400$. The described optimization approach has been able to adapt to the given dictionaries as it can be seen in Figure 6.6a. It



(a) Distribution of $\frac{|\langle \boldsymbol{\phi}_i, \boldsymbol{\psi}_j \rangle|}{\|\boldsymbol{\phi}_i\| \|\boldsymbol{\psi}_j\|}$ (b) Distribution of $\frac{|\langle \boldsymbol{\theta}_i, \boldsymbol{\theta}_j \rangle|}{\|\boldsymbol{\theta}_i\| \|\boldsymbol{\theta}_j\|}$

Figure 6.6.: Evaluation of the optimization approaches for $m = 30$, $l = 200$ and $n = 400$.

[4]Since the comparison is of fundamental nature and not highly competitive, it is reasonable to chose a widely known instead of some cutting-edge M($\boldsymbol{\Theta}$)-optimization variant.

might be surprising that $\mathbf{\Phi}_{\mathrm{M}(\mathbf{\Phi},\mathbf{\Psi})}$ is more incoherent to the dictionary $\mathbf{\Psi}_{\mathrm{rand}}$ than to $\mathbf{\Psi}_{[\mathrm{I,DCT}]}$. However, there is more space for an optimization of $\mathbf{\Phi}$ since the elements of a random matrix are more coherent than those of $\mathbf{\Psi}_{[\mathrm{I,DCT}]}$. The approach of [Ela07] does not significantly influence the coherence between measurement matrix and dictionary. Similarly, the optimization of $\mathrm{M}(\mathbf{\Phi},\mathbf{\Psi})$ has almost no influence on $\mathrm{M}(\mathbf{\Phi\Psi})$, as it can be seen in Figure 6.6b where the optimization result of [Ela07] can be observed. For reference, the corresponding distribution of a non-restricted low-coherence matrix $\mathbf{\Theta}_{\mathrm{BASC}}$ is given as well. As briefly mentioned before, the approach of [Ela07] leaves still space for further improvement, as it can be also seen from the figure.

The reconstruction performance of both optimization approaches is evaluated by plotting the success rate over the sparsity of the unknown vector $\boldsymbol{\chi}$ in Figure 6.7 for a noiseless CS scheme in the case of $\mathbf{\Psi}_{[\mathrm{I,DCT}]}$.[5] Both optimization approaches lead to an increased

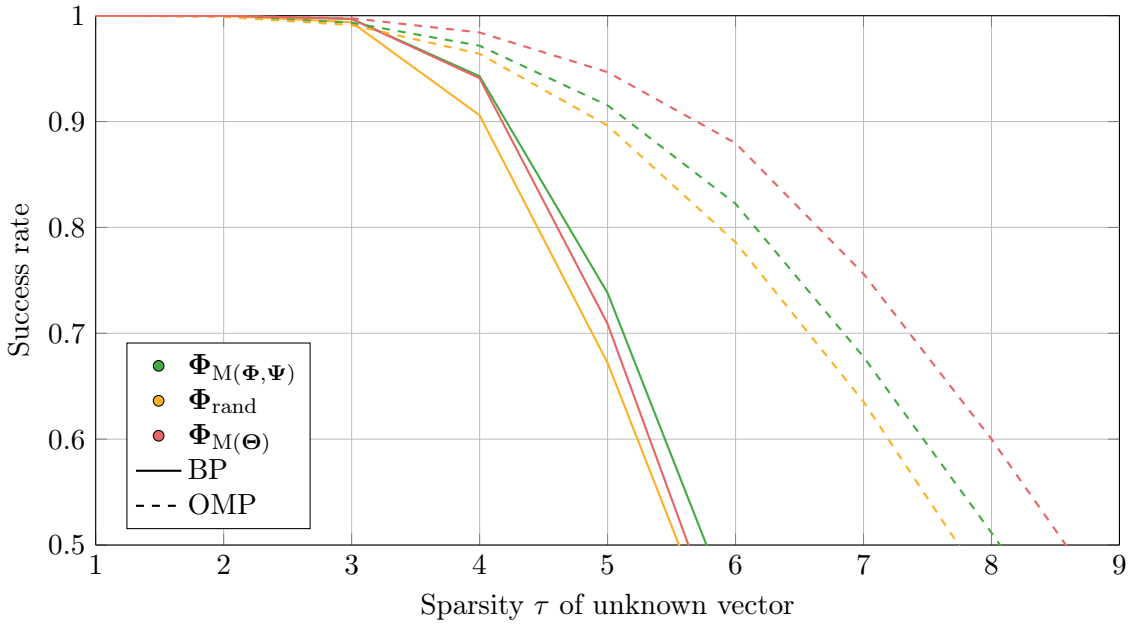


Figure 6.7.: Evaluation of the measurement matrix optimization approaches by their success rate for $\mathbf{\Psi}_{[\mathrm{I,DCT}]}$ with $m = 30$, $l = 200$ and $n = 400$. Simulation parameters: Appendix H.3.

success rate compared to a random measurement matrix $\mathbf{\Phi}_{\mathrm{rand}}$, where the achieved gains are significantly larger in case of OMP. For the given example, the optimization of $\mathbf{\Phi}_{\mathrm{M}(\mathbf{\Phi},\mathbf{\Psi})}$ comes with a slightly better success rate as the approach of [Ela07] in the case of BP. However, this advantage is only marginal especially considering that the optimization approach of [Ela07] leaves space for improvement, cf. Figure 6.6b. In contrast to this, $\mathbf{\Phi}_{\mathrm{M}(\mathbf{\Theta})}$ is clearly superior in case of column coherence based reconstruction algorithms like the OMP, as it can be also seen in the figure, since those naturally gain especially in case of an optimization with respect to $\mathrm{M}(\mathbf{\Theta})$. For other examples of dictionaries, e.g., $\mathbf{\Psi}_{\mathrm{rand}}$, also with different dimensions, similar results have been obtained. Inferring from the provided numerical findings, the coherence

[5] For further comparison, the corresponding boxplots of the squared error are given in Appendix D on page 111.

of the sensing matrix $\mathrm{M}(\boldsymbol{\Theta})$ is the more influential criterion especially for corresponding algorithms, e.g., OMP. In case of other reconstruction approaches, an optimization of $\mathrm{M}(\boldsymbol{\Phi}, \boldsymbol{\Psi})$ might still be considered, whereas its advantage is marginal at best.

6.4. Summary and Overview

The BASC search approach is proposed in this chapter for coherence optimization. The relevance of this contribution to the field of CS has been demonstrated for two independent CS variants. The chapter started with a general introduction to the coherence of vector sets in Section 6.1. In order to provide a theoretical foundation for coherence optimization, multiple relevant bounds have been summarized for complex and real vector spaces. Additionally, connections to other research fields, namely frame theory and Grassmannian line packing, have been established. Subsequently, an overview on existing concepts for coherence optimization has been provided, where analytical and numerical approaches have been considered. Within Section 6.2, the concept of antipodal spherical codes has been introduced. Based on the equivalence of coherence and distance optimization for these antipodal codes, a search strategy for BASC has been proposed by extending an existing method for real-valued BSC to the case of complex vector spaces and subsequently to antipodal spherical codes. The obtained BCASC search approach has been thoroughly compared to other optimization methods. Finally, implementational aspects have been discussed as well as approximations which can be used to reduce computational costs. In Section 6.3, applications of the obtained coherence optimization approach are discussed. Due to the original motivation, the direct application for sensing matrix optimization is most striking. The suitability of the obtained matrices is estimated with respect to the earlier introduced uniqueness assuring properties and the evaluated by numerical simulations. As another example for a potential application in CS, the search method has been extended in order to accommodate a proposed optimization criterion for measurement matrix adaption with respect to a given dictionary. The obtained measurement matrix is finally examined and compared to the result of a common approach which is based on a different optimization criterion. Thereby, some light could be shed on an ongoing confusion of two different optimization criteria which are both based on coherence.

The proposed approach for coherence minimization is an extension to the distance optimization method of [LSZ88], which has been provided for real vector spaces. Since coherence is a limiting factor in many applications, its optimization is of natural interest and importance. Due to its fundamental character and its connections to other research fields, it is obvious that further applications are not limited to the field of CS. For example in [WF15], the proposed approach is used as reference. Compared to existing numerical approaches, the provided method obtained improved coherence values for a wide range of vector sets. In the case of complex vector spaces, a computationally costly numerical integration has to be performed due to the lack of an analytical solution. This potential drawback can be countered by utilizing a simpler but faster numerical summation which yields often only slightly worse coherence results. An analytical solution to the described integral might be subject of future investigations. As by the example of fixed codewords or a fixed dictionary, the proposed scheme can be adapted in order to suit additional side conditions.

In Figure 6.8, the efforts of the current chapter on coherence optimization by BASC are put into the overall context of CS, cf. Figure 4.1. The concept of BASC has been proposed

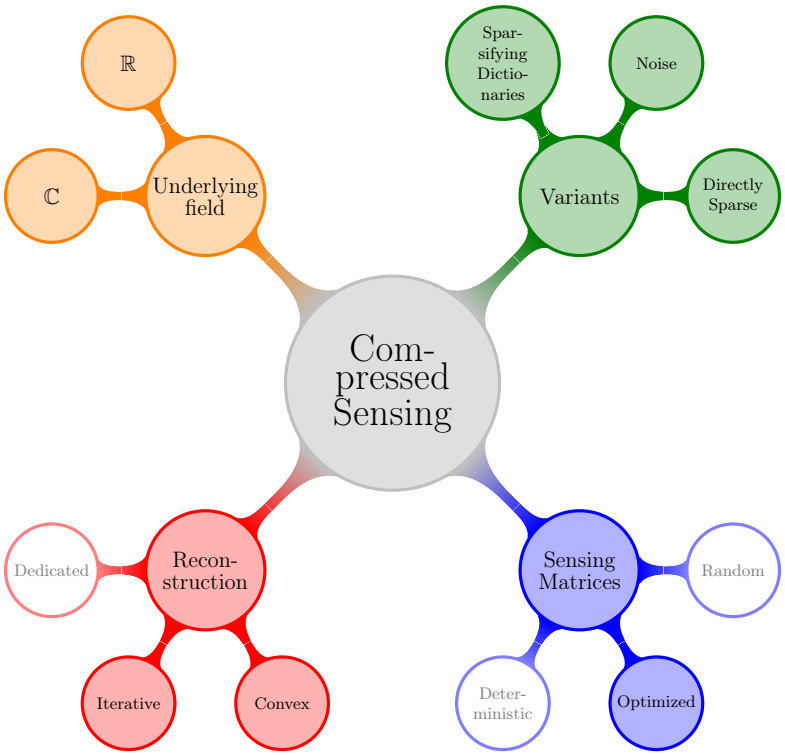


Figure 6.8.: Classification of this chapter's contents within the overall picture of Compressed Sensing (cf. Figure 4.1 on page 33).

for real and complex vector spaces within this chapter. Since the sensing matrices arise from numerical optimization, there is no deterministic structure, and consequently, there are no structures which could be used by dedicated algorithms. However, all general reconstruction algorithms can be applied, where coherence based variants, e.g., OMP, are especially suited in case of coherence optimized sensing matrices. Therefore, potential applications are not limited to noiseless CS scenarios. The proposed coherence minimization technique can be extended to suit an optimization criterion by [CR07, CW08] for the adaptation of measurement matrices with respect to a given sparsifying dictionary. Thereby, the coherence of the resulting sensing matrix proofed to be the more adequate criterion in most cases.

7

Compressed Sensing Based on Complex-Valued Reed–Solomon Codes

The motivation of the previous chapter has been to obtain sensing matrices with advantageous properties which allow for improved reconstruction performance with well-established recovery algorithms. As an alternative approach, deterministic CS schemes can be designed such that special structures of the sensing matrices can be exploited by dedicated algorithms which might offer faster computation or an increased reconstruction performance [DeV07, AMM12]. Most of the deterministic constructions, which have been proposed in the last decade, are based on channel coding approaches [DeV07, PH08, HCS08, AHSC09, CHJ10a, AMM12]. With the same motivation, the focus will be on deterministic CS schemes based on CRS codes in the remaining chapter.

In the 1980s, Marshall [Mar81, Mar84] and Wolf [Wol83b] were the first to investigate RS codes over the complex numbers. Within their work, they denoted the resulting CRS codes as *analog codes*, *DFT codes* or *real-number codes* [Wol83a, Wol83b, Mar84]. Several researchers worked on CRS codes and corresponding decoding techniques in the following decades [MS85, Kum85, Hen89, MHET99, Red00, RG04, TH08, AT08, HHZ11, VL14, Red14]. Potential applications like *Orthogonal Frequency-Division Multiplexing* (OFDM) schemes further increased research interest [Hen00, HH05, ADAA08, HHH12].

The CS scheme is compared to RS in [CW08, FR13], where typically its robustness against noise is criticized. However, this is caused by not spending further attention on the necessary adaption of the decoding algorithms. The application of CRS codes in CS is successfully introduced in an explicit form by [PH08].

In the following section, CRS codes and their properties are introduced. Additionally, the connection to CS is established. In Section 7.2 on page 79, the concept of robust decoding is introduced and applied to CRS codes. Thereby, the potential of power decoding in noisy scenarios is evaluated for different error location and evaluation algorithms. Afterwards, the verification and iterative improvement on the actual error locations is discussed and the performance of a corresponding CS scheme is evaluated. Subsequently in Section 7.3 on page 89, the concept of *Continuity Assisted Decoding* (CAD), which allows to decode beyond the radius for power decoding, is established and connected to the Padé-approximation. The potential use of the additional reliability information for CRS codes and CS schemes based thereon is presented as well in this section.

7.1. Complex-Valued Reed–Solomon Codes

In the following, the notation of RS codes from Section 2.2 on page 8 is reused for the introduction of CRS codes in order to underline the close relationship, therefore, CRS codes are referred to for the remainder of this chapter if not stated otherwise. In Section 7.1.1, the definition of CRS codes is given and several properties are discussed. Subsequently, the connection of CRS codes to CS is given in Section 7.1.2 on the next page.

7.1.1. Definition and Properties

The definition of CRS codes corresponds to the one of RS codes over $\mathbb{F}_q$ (cf. Chapter 2 on page 5), where the DFT is used. Compared to the previously introduced DFT [cf. (2.3)], a slightly different variant is considered for the complex case in the following:

$$A_i = \frac{1}{\sqrt{n}}\boldsymbol{a}(\alpha^i), \quad a_i = \frac{1}{\sqrt{n}}\boldsymbol{A}(\alpha^{-i}) \quad \forall\, i \in [0, n-1] \tag{7.1}$$

Note the two scalar factors of $1/\sqrt{n}$, which correspond to the factor $1/n$ which often occurs in only one of the transforms. However, the given notation of (7.1) is beneficial since the DFT and IDFT matrices are the complex conjugate of each other, which simplifies later expressions.

With this definition of the DFT, a $\mathcal{CRS}$ code of length n and dimension k is defined as a set of codewords $\boldsymbol{a}(z) \in \mathbb{C}[z]$ with $\deg \boldsymbol{a}(z) = n-1$:

$$\mathcal{CRS} = \left\{ \boldsymbol{a}(z) \mid a_i = \frac{1}{\sqrt{n}}\boldsymbol{A}(\alpha^{-i}),\ \forall\, i \in [0, n-1],\ \deg \boldsymbol{A}(z) < k \right\},$$

where $\boldsymbol{A}(z) \in \mathbb{C}[z]$ and $d = n - k + 1$ is the minimum Hamming distance (hence, CRS codes are MDS, cf. (2.7) on page 8) and α is an element of order n which is commonly chosen to be an n-th root of unity $\alpha = \mathrm{e}^{-\mathrm{j}2\pi/n}$. The corresponding parity-check matrix $\boldsymbol{H} \in \mathbb{C}^{n\times(n-k)}$ contains the factor $1/\sqrt{n}$ as well:

$$\boldsymbol{H} = \frac{1}{\sqrt{n}} \begin{pmatrix} \alpha^0 & \alpha^0 & \alpha^0 & \cdots & \alpha^0 \\ \alpha^k & \alpha^{(k+1)} & \alpha^{(k+2)} & \cdots & \alpha^{n-1} \\ \vdots & \vdots & & & \vdots \\ \alpha^{(n-1)k} & \alpha^{(n-1)(k+1)} & \alpha^{2(k+2)} & \cdots & \alpha^{(n-1)(n-1)} \end{pmatrix} \tag{7.2}$$

As for conventional RS codes, a CRS codeword can be interpreted as a column vector $\boldsymbol{c} \in \mathcal{CRS}$ which can also be written as a polynomial $\boldsymbol{c}(z) = \sum_{i=0}^{n-1} c_i z^i$, where the components of $\boldsymbol{c}$ are denoted by $c_i \in \mathbb{C}$. From $\boldsymbol{c} \in \mathcal{CRS}$ and the definition of $\boldsymbol{H}$ as in (7.2), the relation $\boldsymbol{H}^{\mathrm{T}}\boldsymbol{c} = \boldsymbol{0}$ can be obtained.

In contrast to finite fields, there is no field built upon α^i, although α is an element of order n, which generates a multiplicative group by its powers, since the set $\{0, \alpha^i\ \forall\, i \in [0, n-1]\}$ is obviously not closed under addition (cf. (2.1) on page 6). Additionally, all polynomial coefficients are not restricted to the aforementioned multiplicative group and can attain any complex number ($c_i \in \mathbb{C}$). Therefore, the results of all addition-based operations are generally not restricted to $\{0, \alpha^i\ \forall\, i \in [0, n-1]\}$.

The received word $\boldsymbol{r} \in \mathbb{C}^n$ is the sum of the codeword and a sparse error $\boldsymbol{e} \in \mathbb{C}^n$: $\boldsymbol{r} = \boldsymbol{c} + \boldsymbol{e}$. The indices of the non-zero coefficients in $\boldsymbol{e}$ correspond to the error positions, which are denoted by $\mathcal{U} = \{i \mid c_i \neq r_i\}$. The number of errors in $\boldsymbol{r}$ is consequently equal to $\tau = \#\mathcal{U}$. The syndrome is defined as for conventional RS codes with $\boldsymbol{s} = \boldsymbol{H}^{\mathrm{T}}\boldsymbol{r} = \boldsymbol{H}^{\mathrm{T}}\boldsymbol{e}$.

In the context of CRS codes, distributed background noise is typically considered which may arise depending on the application and implementation. This additional noise $\boldsymbol{\eta} \in \mathbb{C}^n$ affects the received word $\boldsymbol{r} = \boldsymbol{c} + \boldsymbol{e} + \boldsymbol{\eta}$ and needs, therefore, to be considered during the decoding process as it is demonstrated subsequently in Section 7.2 on page 79.

Since all coefficients of CRS codes are complex, Marshall argues that the CRS codes are analogous to complex-valued versions of BCH codes and denotes these codes as *complex-number BCH codes* [Mar84]. Real-numbered BCH codes can be obtained by ensuring that $\boldsymbol{H}$ contains for each column also the complex-conjugate as it is already described in [Mar81]. Since these real-numbered BCH codes can be considered as subfield-subcodes of CRS codes, it would be more suited to describe just those real-number codes as BCH codes in this context.

For conventional codes, it is known that unique decoding is possible until half the minimum distance: $\tau \leq \lfloor (d-1)/2 \rfloor$, where τ represents the number of errors [Bos99]. In case of CRS codes however, Wolf showed in [Wol83b] that unique decoding is almost surely possible (with probability 1) until $\tau \leq d - 2$ (cf. Section 3.2.2 on page 21). However, there is currently no fast algorithm available which achieves this bound.

As for the above described CRS codes, the concepts of IRS codes and virtual extension, presented at the end of Section 2.2.1 on page 10, can be equivalently transfered to the complex-valued scenario. For the sake of brevity, a mostly redundant introduction to *Interleaved Complex-valued Reed–Solomon* (ICRS) codes is omitted here as well as the complex-valued version of virtual extension. For references and definitions, the reader may refer to the corresponding section on page 10.

7.1.2. Connection to Compressed Sensing

The previously described CRS codes can be used in a CS scheme, as it is proposed in [PH08], where a variant of the *Coppersmith–Sudan* decoding algorithm [CS03] is used: The sensing matrix $\boldsymbol{\Theta}$ used in this CS scheme corresponds to the transposed parity check matrix $\boldsymbol{H}^{\mathrm{T}}$ of a CRS code which is based on a DFT matrix with $m = n - k$. The (noise-free) measurement vector $\boldsymbol{\beta} = \boldsymbol{H}^{\mathrm{T}}\boldsymbol{\chi}$ can be translated with help of the complex conjugate $\boldsymbol{H}^*$ into a received word [PH08]

$$\boldsymbol{r} = \boldsymbol{H}^*\boldsymbol{\beta} = \boldsymbol{c} + \boldsymbol{e}, \tag{7.3}$$

which can be also represented as the addition of an arbitrary codeword $\boldsymbol{c} \in \mathcal{CRS}$ and some error vector $\boldsymbol{e}$. As for any other received word, the syndrome can be computed by applying the parity check matrix $\boldsymbol{H}$:

$$\boldsymbol{s} = \boldsymbol{H}^{\mathrm{T}}\boldsymbol{r} = \boldsymbol{H}^{\mathrm{T}}\boldsymbol{e} = \boldsymbol{H}^{\mathrm{T}}\boldsymbol{H}^*\boldsymbol{\beta}. \tag{7.4}$$

Since $\boldsymbol{H}^*$ corresponds to a partial IDFT matrix, $\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H}^*$ results in the identity matrix, and therefore, the measurement vector $\boldsymbol{\beta}$ equals the syndrome $\boldsymbol{s}$ and, consequently, the searched sparse vector coincides with the error $\boldsymbol{\chi} = \boldsymbol{e}$. Hence, CRS decoding algorithms can be utilized in a CS scheme to find $\boldsymbol{\chi}$. The actual value of the implicitly assumed codeword $\boldsymbol{c}$ is not of further importance for the CS scheme and is therefore not calculated during the decoding.

In power decoding, the syndrome is extended by component-wise powering the received polynomial, cf. (2.16) on page 12. Due to the aforementioned relation, an explicit calculation of $\boldsymbol{s}^{(1)}$ can be avoided since the measurement vector equals the syndrome. However, the extended syndromes $\boldsymbol{s}^{(l)}$ with $1 < l \leq l_{\max}$ need to be calculated conventionally nevertheless.

Within the introduction of CRS codes in Section 7.1.1, potential background noise has been described. As argued in Section 4.4.1 on page 31, noise is also often considered in the context of CS. However, it should be noted that the noise is typically added in different places of the overall scheme. For typical applications of CRS codes, noise affects the received word $\boldsymbol{r}$, while for CS schemes, the (noise-free) measurement vector $\boldsymbol{\beta}$ or the sparse vector $\boldsymbol{\chi}$ are typically affected. These different variants result in equivalent noise vectors which can be interpreted as filtered versions of each other. This fact should be carefully considered when comparing analyses based on different models. In order to assist comparability, it is assumed for the remainder of this thesis that zero-mean AWGN $\boldsymbol{\eta} \in \mathbb{C}^{n-k}$ is affecting the vector $\boldsymbol{\beta}$ resulting in $\boldsymbol{\mu} = \boldsymbol{\Theta}\boldsymbol{\chi} + \boldsymbol{\eta}$, cf. Section 4.4.1 on page 31.

In Figure 7.1, the potential application of CRS power decoding algorithms in a CS-scheme is summarized as block-diagram. The green parts indicate the possible application of syndrome

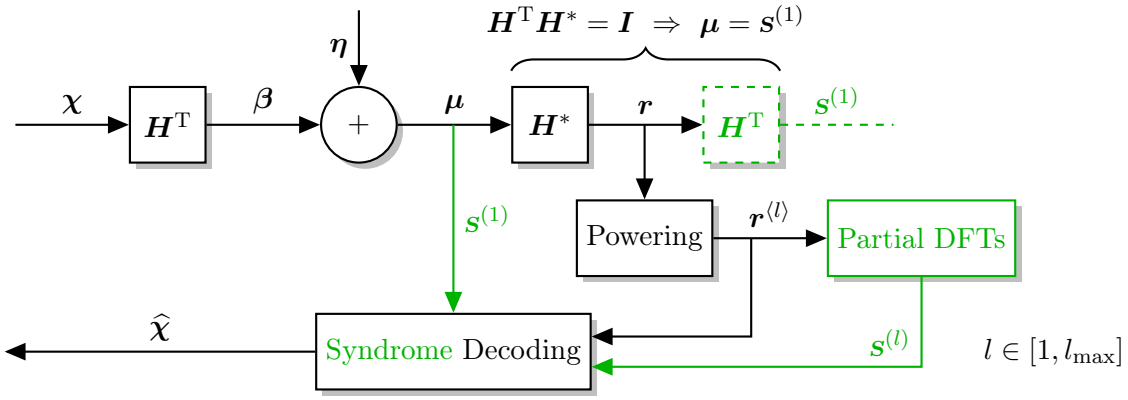


Figure 7.1.: Block diagram for the application of CRS power decoding schemes to noisy CS. Syndrome decoding is in green, where avoidable calculations are dashed.

decoding, where the dashed lines indicates that calculating $\boldsymbol{s}^{(1)}$ explicitly is not necessary.

In order to have a column normalized sensing matrix, a scaling factor needs to be introduced: $\boldsymbol{\Theta} = \sqrt{n/m}\boldsymbol{H}^{\mathrm{T}}$. As a consequence, this scaling factor needs also to be considered for the reconstruction in (7.3) and (7.4). With exception of a slight potential loss of precision for floating point implementations observable only in noise-free scenarios, this scaling factor does not have further consequences for the decoding. In noisy cases however, it should be noted that the value of such a scaling factor influences the performance of the system with respect to some fixed noise power. Therefore, column normalized sensing matrices are considered for the remainder of this thesis in order to provide reasonable and fair comparisons.

CRS codes can be uniquely decoded up to half the minimum distance as described before. This bound corresponds to the condition for SLEs derived by the null space (cf. Section 3.2.2 on page 21). Subsequently, several approaches are introduced to overcome this bound. In the context of CS, bounds similar to previously discussed almost sure decodability of CRS codes up to $d - 2$ errors are obtained by [VB98, Wak06, FR13].

The common separation of error location and evaluation in channel coding can also be found in CS approaches which are focussed on support recovery, e.g., [CHJ10b]. Thereby, determining the error positions is considered as the actual problem for which several algorithms have been proposed, cf. Section 4.1 on page 24. The subsequent error evaluation is typically obtained as a least-squares solution. However, there are also joint approaches for CS as well as for decoding CRS codes, cf. BP or [Red00].

7.2. Robust Decoding of CRS Codes for Compressed Sensing

Since the principles of linear algebra do not differ between finite and infinite fields, all conventional RS decoding schemes can be used for CRS codes as well. However, all combinatorial arguments used in finite fields cannot be applied to infinite fields [Mar84]. Furthermore, the numerical stability of a potential algorithm becomes an additional criterion. In consequence, algorithms for RS codes might be intractable or unstable when applied to CRS codes. For example, the Guruswami–Sudan algorithm contains a factorization step which makes it unattractive for CRS codes [PH08]. The stability or robustness of CRS decoding algorithms has been discussed and investigated in several publications since the early days of CRS codes, e.g., [Kum85, Hen89, MHET99]. Thereby, non-sparse background noise is typically assumed which may arise depending on the application and implementation. For example, such noise can be caused by the finite-precision of the hard and software implementation, by quantization of the vector components, or even by the measurement processes itself.

For the remainder of this work a floating-point representation [IEE08] is considered for the real and imaginary parts of the complex-values which arise in the context of CRS codes. Due to the finite precision, rounding errors cannot be avoided and the corresponding decreased accuracy might accumulate or even amplify during subsequent calculations. Additionally, there might be over and underflows where the calculated result leaves the representable number range. In case of cancellation, significant bits of the number representation are lost which can occur for example by subtracting two close numbers. Consequently, the result of such a subtraction is less reliable and potentially inaccurate. As a direct result, it should therefore be noted that the accuracy of a given algorithm in floating-point arithmetic depends also on its implementation. For example, addition and multiplication are neither guaranteed to be associative nor distributive in floating-point arithmetic. While these operations are commutative for two terms or factors, the order of the operations can be influential in case of more terms or factors. For more details about the consequences of floating-point operations on algorithms, refer to [Hig96].

Due to the finite precision, the performance evaluation is more complicated for CRS codes in comparison to the unique-decoding of RS codes over finite fields. Since the decoding result of RS codes is unique, the performance is evaluated with respect to the complexity of the decoding algorithm. However, for the presented scenario of CRS codes, the error location and evaluation algorithms given in Section 2.2.2 on page 11 and their individual implementations differ in the obtained accuracy level and stability when operated over the field of complex numbers in a noiseless setting. Therefore, they need to be evaluated with respect to the quality of the decoding result under distorting influences. Depending on the application, the stability of an algorithm might even be the more important property. Furthermore, additional distributed noise might overlay such precision losses and be the dominating effect

depending on its magnitude. Therefore, the influence of the floating-point representation might be of reduced relevance in noisy settings. Instead, the stability of the algorithms and their implementations is even more important as soon as noise is considered.

In order to cope with the additional distributed noise, it is straight forward to add thresholds to existing RS decoding algorithms, e.g., [Hen89, Red00]. The actual values of these thresholds depend, besides on the finite precision, also on the additional noise which might not be known a-priori. Therefore, the noise needs to be estimated in such cases. Depending on the application, the used thresholds might also account for decreased precision, e.g., due to many operations. Most approaches determine the actual values for the thresholds empirically, however, there are also attempts to give adaptive thresholds or specialized algorithms [RG04, TH08, HHZ11, Red14].

Since most approaches to CRS decoding are based on well-established techniques from channel coding over finite fields, the zero characteristic and the missing combinatorial arguments in infinite fields are often presented as drawback. However, there are also advantageous aspects of the infinite fields, as shown in the remainder of this chapter. In Section 7.2.1, basic error location and evaluation algorithms and their adaptations to CRS codes are discussed. Thereby, the influence of distributed noise is examined. Subsequently in Section 7.2.2 on page 86, methods are presented which perform an additional verification step followed by an iterative improvement of the error locations.

In the following, the case of conventional (almost sure) unique decoding is considered in Sections 7.2.1 and 7.2.2. Therefore, the main interest is on numbers of errors which are limited to the power decoding radius $\tau_\mathrm{m} = \tau_\mathrm{max}^{(l_\mathrm{max})}$, as described by (2.11) on page 11. The results provided in this section have been partly obtained in the course of a supervised master's thesis [Riz14] and provide the basis for a subsequent extension given in Section 7.3 which allows to decode even beyond the power decoding radius.

7.2.1. Basic Error Location and Evaluation in Noisy Environments

In the following, the error location and evaluation algorithms given in Section 2.2.2 on page 11 are modified for the use over the complex numbers in a power decoding manner. Their performance is subsequently evaluated at the end of this section.

Peterson Type Algorithm

As described before for the decoding of RS codes on page 13, the Peterson algorithm determines the number of errors by exploiting $\tau = \operatorname{rank} \boldsymbol{S}$. The common approach for RS codes considers only the minimum number of equations from (2.17) on page 12 by reducing the assumed number of errors ν (starting from τ_m) until $\det \boldsymbol{M}_\nu \neq 0$. However, the stability of CRS decoding algorithms is generally improved by considering all available equations (and therefore the complete syndrome). Thus, the full matrix $\boldsymbol{M}_{\tau_\mathrm{m}} = \boldsymbol{S}$ should be used for CRS codes. Since the rank of a matrix equals the number of its non-zero singular values, this can be achieved by applying the *Singular Value Decomposition* (SVD) on $\boldsymbol{S} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\mathrm{H}$, where $\boldsymbol{U}$ and $\boldsymbol{V}$ are unitary square matrices and $\boldsymbol{\Sigma}$ is a real-valued matrix with singular values $\Sigma_{i,i}$ of $\boldsymbol{S}$ on the diagonal. In order to determine this number, a threshold ϵ or some other decision criteria can be used as previously discussed:

$$\tau = \#\{\ i\ :\ \Sigma_{i,i} > \epsilon\ \} \quad \forall\, i \in [1, \tau_\mathrm{m}]$$

From this formula, it is obvious that the determined number of errors is upper bounded by τ_{m}. For a noisy environment, ϵ needs to be adjusted correspondingly. As argued before, there are also alternatives to empirically determining ϵ [RG04, TH08, HHZ11].

In a subsequent step, the error locator $\boldsymbol{\Lambda}$ is determined over (2.17). For RS codes over finite fields, it is sufficient to reduce this system to a square matrix which can be efficiently inverted and solved. However, for the presented scenario of CRS codes, the overdetermined system will typically not have a unique solution due to noise and finite precision. Therefore, an ℓ_2-minimization over the full system in (2.17) can be used to determine $\boldsymbol{\Lambda}$, instead of computing the inverse of the reduced non-singular matrix $\boldsymbol{M}_\tau$. This least squares solution can be obtained by a Moore–Penrose pseudoinverse [Moo20, Bje51, Pen55], cf. Section 3.1.1 on page 18. Due to the increased number of equations, this approach provides more stable results and minimizes the remaining noise energy.

Berlekamp–Massey Type Algorithms

The SS algorithm [SS06] given in Algorithm A.1 and discussed on page 13, which extends the well-known BM algorithm to power decoding, can also be generalized for CRS codes. For the conventional case, the discrepancy calculated by (2.20) is compared to zero. Following the previous discussion, a threshold must be used instead for the application to CRS decoding. Furthermore, the accuracy of $\boldsymbol{s}^{(l)}$ for $l \geq 2$ is reduced since the noise effects contained in $\boldsymbol{s}^{(1)}$ are accumulated by powering component-wise. These accuracy differences need to be considered for the corresponding values of ϵ during the evaluation of $\Delta_t^{(l)}$ by different empirically determined values or by adaptive algorithms.

For the conventional description over finite fields, the synthesis algorithm checks all sequences with full length whether they can be generated by an LFSR found with $\Delta_t^{(l)} = 0$. Due to the finite characteristic of the field, this check is necessary since each field element can occur with a finite probability, and therefore, $\Delta_t^{(l)} = 0$ could have been obtained by coincidence. Because of the finite precision and the potential noise, these checks are problematic for the decoding of CRS codes, since even small values with $\Delta_t^{(l)} < \epsilon$ might accumulate subsequently and lead to an overestimation of the number of errors. However, since the complex field is of characteristic zero, the probability of obtaining a short connection polynomial generating only a partial sequence is zero, and therefore, these checks are not necessary. As consequence, the varying length multi-sequence LFSR synthesis algorithm can be stopped as soon as $\Delta_t^{(l)} = 0$ is reached. Hence, the zero characteristic has an advantageous aspect for the generation of minimum-length LFSR.

The length of the obtained error locator polynomial $\boldsymbol{\Lambda}(z)$ determines the number of detected errors and the IDFT $\boldsymbol{\lambda}(z) = \mathscr{F}^{-1}\left[\boldsymbol{\Lambda}(z)\right]$ can be used to find the error positions. In an ideal and noise-free scenario, the coefficients λ_i, which correspond to the error positions ($i \in \mathcal{U}$), of the time-domain polynomial $\boldsymbol{\lambda}(z)$ are zero according to the definitions of DFT and $\boldsymbol{\Lambda}(z)$. For typical implementations of CRS codes, however, these coefficients are more likely of rather small magnitude. Therefore, the selection of the error locations can be based on the magnitudes of λ_i for $i \in [1, n-k]$, cf. [Red00, ADAA08, TH08], such that the $\widehat{\tau}$ coefficients with smallest magnitudes provide the estimated error locations $\widehat{\mathcal{U}}$.

Extended Euclidean Type Algorithms

As introduced in Section 2.2.2 on page 14, the EE algorithm and its generalized version for power decoding, the MS algorithm, operate both over polynomials. Within these algorithms, the degree of a polynomial $\boldsymbol{p}(z) \in \mathbb{C}[z]$ needs to be determined which can be achieved by

$$\deg \boldsymbol{p}(z) = \arg\max_{i \in \mathbb{N}_0} |p_i| > \epsilon.$$

Thereby, it should be noted that the magnitude of the coefficients can differ considerably. Therefore, all polynomials in a row should be normalized with respect to the largest coefficient of these polynomials after each iteration in order to prevent overflows. Such a scaling is not harmful to the algorithm, cf. Footnote 3 on page 15. As soon as the algorithms terminated successfully, the error positions $\widehat{\mathcal{U}}$ can be obtained from the error locator polynomial $\boldsymbol{\Lambda}(z)$ as described previously for BM type algorithms.

As it can be seen from the equations in (2.22), the MS algorithm is especially susceptible to precision loss due to the iteratively repeated row reductions. (The same observation holds for the row subtractions performed in the EE algorithm.) Since the row reductions are applied in order to reduce the degree of the leading term, the corresponding highest coefficient can be set to an exact zero, cf. the description on page 15. Similarly, it proved to be beneficial for the overall stability that coefficients with values less than ϵ are also set to zero. Consequently, it might happen that $\boldsymbol{M}[z]$ contains a row with zeros. Since $\deg 0$ is commonly not defined, such rows cannot be considered in the algorithm and are ignored.

Numerical Evaluation of Error Location Algorithms

In the following, the performance and noise sensitivity of the discussed error location algorithms is evaluated. For the simulations of a noisy CS scheme, a CRS code of length $n = 50$ and dimension $k = 12$ is considered, where complex-valued zero-mean AWGN with a standard deviation of $\sigma_{\boldsymbol{\eta}} = 10^{-5}$ is affecting the measurement $\boldsymbol{\beta}$ as described in Sections 4.4.1 and 7.1.2 on page 31 and on page 78. Further details on the used simulation parameters are summarized in Appendix H.4 on page 121. Since error location algorithms are evaluated, the nomenclature of CRS codes instead of CS is subsequently used during the evaluation.

In a first step, the discussed algorithms are evaluated for the reliability of their estimated number of errors $\widehat{\tau}$ in Figure 7.2, where $\widehat{\tau}$ is determined by the length of $\boldsymbol{\Lambda}(z)$ in the case of BM and EE type algorithms. Within this figure, boxplots are used to illustrate the statistical behavior, where the distributions of the obtained values for $\widehat{\tau}$ are visualized over the actual value τ. The solid line represents the ideal result for the case of successful power decoding in a noise-less scenario, where, all three algorithms provide equally reliable results close to the best achievable case. However, for the described noisy scenario, the number of errors obtained by the SVD based approach of the given Peterson algorithm is more reliable than those of the other two algorithms for a wide range of errors, e.g., for $\tau \leq 18$ in Figure 7.2. For more errors, the Peterson type approach tends to underestimate τ, while the other two algorithms generally tend to overestimation, where the BM type approach is usually closer to the correct number. The EE type algorithm significantly overestimates the number of errors, and therefore, it proves to be especially susceptible to noise. These described tendencies of over and underestimation for the presented algorithms are exploited in the subsequent

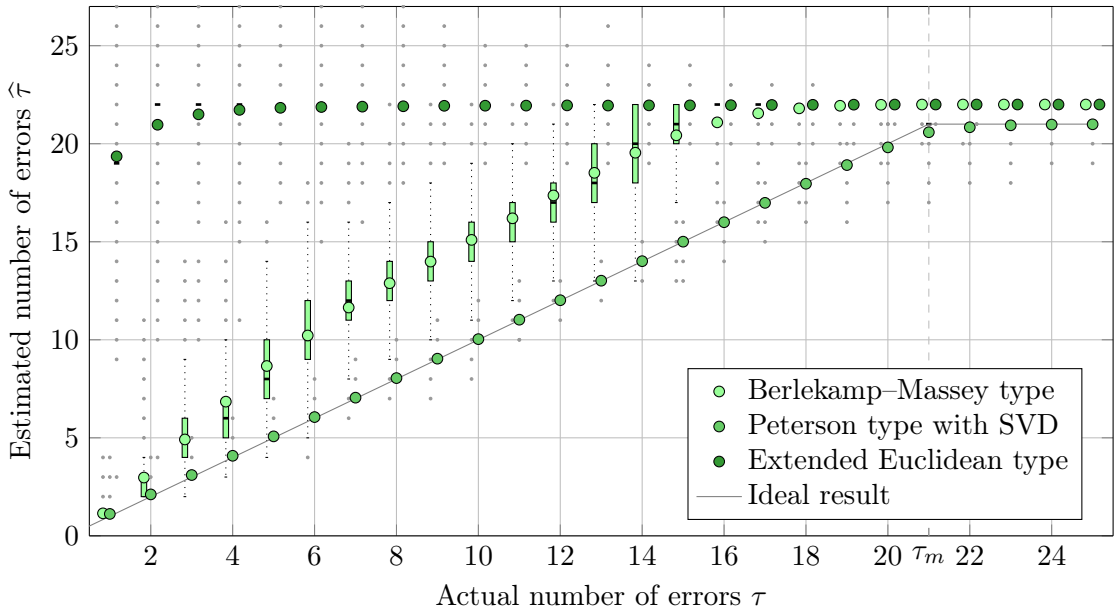


Figure 7.2.: Boxplots illustrating the reliability of the presented power decoding error location algorithms in estimating the number of errors $\widehat{\tau}$. Simulation parameters: Appendix H.4.

Section 7.2.2. Results with $\tau_m < \widehat{\tau}$ correspond to failures in BM and EE type approaches due to the additional noise or too many occurred errors.

Since a correct number of errors does not necessarily indicate the recovery of the correct support, the corresponding estimates of the described error location algorithms have to be evaluated as well. For the described simulation setting, the average support recovery is given in Figure 7.3, where the rate of the recovery is determined by the ratio

$$\frac{\#\widehat{\mathcal{U}}_{\text{corr}}}{\#\widehat{\mathcal{U}}} \quad \text{with } \widehat{\mathcal{U}}_{\text{corr}} = \widehat{\mathcal{U}} \cap \mathcal{U}$$

being the correct part of the estimated support set $\widehat{\mathcal{U}}$. On a first sight, the average support recovery corresponds to the results for the number of errors as it can be seen in the figure: The Peterson type algorithm provides the error locations most reliably, and for a larger number of errors, the BM and EE type approaches achieve comparable reliability while the EE performs typically worse due to its immense overestimation of the number of errors. However, on a second sight, this result might be surprising since the under and overestimation in $\widehat{\tau}$ could be interpreted as an indication for a failure of the corresponding algorithm, and consequently, the estimate of $\widehat{\boldsymbol{\Lambda}}(z)$ should be almost meaningless. As it is clear from Figure 7.3, this is not the case: The obtained estimates for the error locator polynomial $\widehat{\boldsymbol{\Lambda}}(z)$ are not very different from the actual $\boldsymbol{\Lambda}(z)$ and the dominating problem is in determining the correct number of errors τ. Additionally, it might be surprising that the average support recovery is relatively high ($> 75\%$) for all three algorithms in case of a larger number of errors ($\tau \geq \tau_{\text{m}}$), where a total failure of the algorithms would have been expected. The effect of the good-natured estimation $\widehat{\boldsymbol{\Lambda}}(z)$ is subsequently utilized and discussed in Sections 7.2.2 and 7.3.

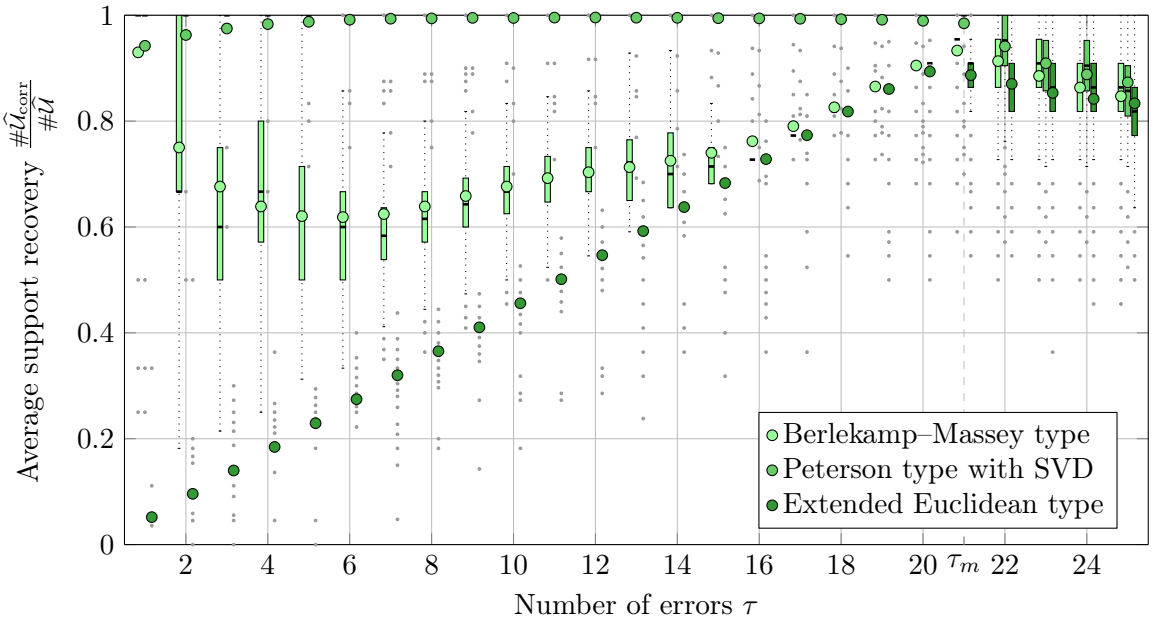


Figure 7.3.: Boxplots illustrating the average support recovery provided by the presented power decoding error location algorithms. Simulation parameters: Appendix H.4.

In the following, the previously provided error evaluation algorithms for RS codes, the GZ and the Forney algorithm, are applied to CRS codes and evaluated for the described error location algorithms.

Gorenstein–Zierler and Forney Algorithms

The GZ algorithm, which is described in Section 2.2.2 on page 16, determines the error values for a given set of error positions by solving an overdetermined system of linear equations which has a unique solution, cf. (2.25). With the same argumentation as for the Peterson algorithm, where an overdetermined system has to be solved as well, the conventional approach of solving a reduced square system is not recommendable. Therefore, the least squares solution of the system represented by (2.25) is considered here, in order to incorporate all available equations.

As previously argued, such an ℓ_2-minimization based error evaluation is also applied in the context of CS reconstruction: Approaches which focus on support recovery, where the algorithms determine the positions of the non-zero components and obtain subsequently the corresponding least squares solution, e.g., [CHJ10b]. The ℓ_2-minimization should not be confused with the convex relaxations like the BPDN, cf. (4.7) on page 25. These approaches treat the noise also by an ℓ_2-norm, however, the support is determined jointly within the optimization.

For comparison purposes, the efficient Forney algorithm, which is briefly described on page 16, is considered for CRS codes as well. Its main equation (2.26) is also valid for the complex case, and therefore, no further adaptation for CRS codes is necessary.

Numerical Evaluation of Error Evaluation Algorithms

For the given error evaluation algorithms, GZ and Forney, the performance with respect to the achieved squared error $\|\boldsymbol{\chi} - \widehat{\boldsymbol{\chi}}\|^2$ is illustrated with combined boxplots in Figure 7.4, where the outliers are canceled from the plot in order to improve overall readability. The simulation

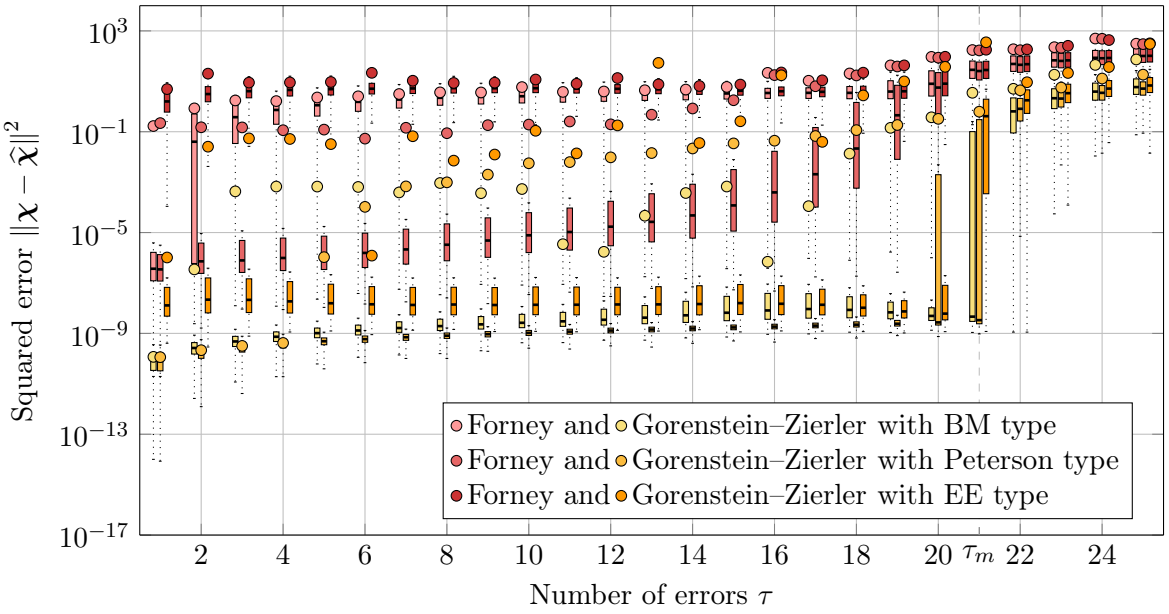


Figure 7.4.: Boxplots illustrating the squared error for the presented power decoding error evaluation algorithms: Forney (red) and GZ (orange), in combination with the presented power decoding error location algorithms (brightness). Outliers are not explicitly shown. Simulation parameters: Appendix H.4.

is based on the previously obtained results for the error location algorithms. As it can be seen in the figure, the efficient Forney algorithm is more susceptible to noise than the GZ algorithm independent of the used error location algorithm. This behavior is caused by the additional use of $\boldsymbol{\Lambda}(z)$ in (2.26), since the corresponding estimate $\widehat{\boldsymbol{\Lambda}}(z)$ is of course also affected by the noise and determined by the preceding error location algorithm. Consequently, it is not beneficial to rely also on $\widehat{\boldsymbol{\Lambda}}(z)$ for the error evaluation. Since the Peterson type algorithm provided the best average support recovery, it is natural that the GZ algorithm shows the best performance for this error location algorithm. It should be noted that the *Mean Squared Error* (MSE) is a pessimistic performance measure since a single bad result might be of severe influence. Since outliers are of reduced influence in the median, it might be a more reasonable measure depending on the intended application. For example, the mean of the GZ algorithm based on the error locator obtained by the BM type algorithm is below 10^{-7} for $\tau < 2$, while the median is below this limit for $\tau \leq 21 = \tau_{\mathrm{m}}$.

The severe differences in reconstruction performance between GZ and Forney emphasize the importance of the additional robustness aspect for CRS codes. For conventional channel codes over finite fields, robustness is not an issue and efficiency with respect to complexity is of major interest. However, potential precision problems and additional noise enforce the consideration of robustness as another criterion for the evaluation of decoding algorithms.

Based on these simulations, only the GZ algorithm is considered in the following, as it results in a least squares solution for a given set of error locations. Subsequently, the focus will be on the iterative refinement of estimated error locations for the next section.

7.2.2. Verification and Iterative Improvement on Error Locations

The adapted standard algorithms given in the previous Section 7.2 arose from the (almost sure) unique decoding of RS codes over finite fields. For such conventional RS codes, the corresponding algorithms are sufficient and no further action is needed since the decoding is proven to be (almost surely) unique for $\tau \leq \tau_\mathrm{m}$. The traditional separation into error location and evaluation algorithms is also reasonable for unique hard-decoding schemes. The mentioned adaptations of the basic error location and evaluation help to cope with the additional noise commonly assumed in CRS systems and allow the application of RS decoding schemes to noisy CRS schemes. However, due to ambiguities introduced by the noise, the situation is more challenging for CRS decoding since the result of an error location algorithm might be erroneous and its failure could be detected during error evaluation. Therefore, the strict separation into error location and evaluation is not ideal for CRS decoding and joint approaches for error location and evaluation are more promising.[1] In the following, additional steps are discussed which resemble such a joint error location and evaluation.

In order to increase the reliability of the obtained results, an additional verification step can be introduced after the error location algorithm. In [Red00] for example, the decoding result is checked by calculating a codeword estimate $\widehat{\boldsymbol{c}} = \boldsymbol{r} - \widehat{\boldsymbol{\chi}}$. Subsequently, the parity check matrix can be used to evaluate the validity of the estimation with the help of the syndrome $\widehat{\boldsymbol{s}} = \boldsymbol{H}^\mathrm{T}\widehat{\boldsymbol{c}}$, which is supposed to be zero for a valid codeword. For this validation, it proved to be beneficial to perform a component-wise check on $\widehat{\boldsymbol{s}}$. Consequently, an error estimation $\widehat{\boldsymbol{\chi}}$ is assumed to be correct if

$$|\widehat{S}_i| < \epsilon \quad \forall \quad i \in [1, N-K]\,. \tag{7.5}$$

Of course, the validity check (7.5) can only indicate whether decoding has been unsuccessful. In such a case, the selected error locations might be wrong and a different set could result in a successful decoding. The coefficients λ_i of the time-domain polynomial $\boldsymbol{\lambda}(z)$ can be utilized to find the correct error positions. In [Red00], an empty set of positions is iteratively increased by those corresponding to the coefficients λ_i with smallest magnitude until (7.5) is fulfilled or τ_m positions have been selected. Another example for a similar approach can be found in [ADAA08] with a validity check slightly different to (7.5): The $\Delta\nu$ coefficients of smallest magnitude are selected, with $\Delta, \nu \in \mathbb{N}$ and $\Delta > 1$, where ν is iteratively increased from 1 to τ_m until the validity check is passed. From the resulting set of locations, all possible combinations of ν coefficients are evaluated by the validity check. Thereby, it should be noted that there might be more than τ_m coefficients considered, however, the final decoding result has at most τ_m error positions.

Within his supervised master's thesis [Riz14], our student proposed an *Iterative Erasure and Evaluation* (IEE) scheme which proved to be suited for noisy scenarios. Afterwards, a simplified description of this scheme has been published within [MRZB15]. Since it is

[1] Since (additional) information from the error evaluation is used, such joint approaches can be considered as soft-decoding schemes.

subsequently extended in Section 7.3, this IEE scheme is briefly described and further analyzed in the following. Based on initial estimates for the error location polynomial $\widehat{\boldsymbol{\Lambda}}(z)$ and the number of errors $\widehat{\tau}_{\text{initial}}$ which are provided by some error location algorithm, the scheme aims to provide an error estimate $\widehat{\boldsymbol{\chi}}$ for $\tau \leq \tau_{\text{m}}$. The approach is partly inspired by GMD decoding, where unreliable coefficients are iteratively erased and the resulting word is decoded each time, cf. Section 2.2.2 on page 12. As for the two previous examples, [Red00] and [ADAA08], the positions corresponding to the smallest magnitudes $|\lambda_i|$ are considered as least reliable and the obtained error estimate is verified by a validity check as in (7.5). The IEE scheme consists out of two loops determining the error positions which are subsequently validated. If this check is failed, the assumed set of erasures is considered to be incomplete and the scheme aims to extend the set accordingly. The outer loop starts with $\widetilde{\tau}_{\text{outer}} = \widehat{\tau}_{\text{initial}}$. The set of erasures $\mathcal{E}$, with $\#\mathcal{E} = \widetilde{\tau}_{\text{outer}}$, is subsequently increased by the position corresponding to the next smallest $|\lambda_i|$ and the validity check is performed for each set until $\widetilde{\tau}_{\text{outer}} = n - k$ (if the algorithm is not terminated earlier). In case of a non-valid error estimate, the outer loop continues immediately and the set $\mathcal{E}$ is increased by the next element. A valid solution with $\widetilde{\tau}_{\text{outer}} \leq \tau_{\text{m}}$ is directly returned and the algorithm terminates. However, in the case of $\widetilde{\tau}_{\text{outer}} > \tau_{\text{m}}$, an inner loop is used to reduce the number of erasures in order to be within the capacity of (almost sure) unique decoding. The number of erasures $\widetilde{\tau}_{\text{inner}}$ for the inner loop is initialized with the number of the outer loop $\widetilde{\tau}_{\text{outer}}$. The recent error vector $\widetilde{\boldsymbol{\chi}}$ can be used to determine which position is removed from $\mathcal{E}$. Therefore, it is assumed that the position resulting in the element with the smallest non-zero magnitude of the obtained error $\widetilde{\boldsymbol{\chi}}$ is due to the noise and not an actual error position. Consequently, such a position is canceled from $\mathcal{E}$ within the inner loop until $\#\mathcal{E} = \widetilde{\tau}_{\text{inner}} = \tau_{\text{m}}$, whereby $\widetilde{\boldsymbol{\chi}}$ is recalculated for each reduced set $\mathcal{E}$. If the thereby obtained vector $\widetilde{\boldsymbol{\chi}}$ does not pass the validity check, the inner loop is left and the outer loop continues with increasing $\widetilde{\tau}_{\text{outer}}$ and choosing the erasures $\mathcal{E}$ as the $\widetilde{\tau}_{\text{outer}}$ the smallest magnitudes $|\lambda_i|$. An algorithmic description of the IEE scheme is given in Appendix F on page 115.

Since the SVD based Peterson type algorithm proved to be most reliable in determining the number of errors, this approach is chosen to provide $\widehat{\tau}_{\text{initial}}$. The described IEE approach does not rely on $\widehat{\boldsymbol{\Lambda}}(z)$ having minimal degree but on the small magnitudes of λ_i for $i \in \mathcal{U}$. In the previous section, it is argued that the estimation of $\boldsymbol{\Lambda}(z)$ is good-natured for an overestimation of the number of errors in case of the presented error location algorithms. Therefore, it is possible to use error locator polynomials of degree τ_{m} which corresponds to the maximal number of errors τ_{m}. The overestimation can be achieved for the Peterson type algorithm by obtaining the least squares solution to (2.17) on page 12 with $\tau = \tau_{\text{m}}$. For the BM and EE type algorithms, the used threshold ϵ needs to be reduced in order to obtain an overestimated error locator polynomial $\widehat{\boldsymbol{\Lambda}}(z)$. The obtained polynomials possess still the desired small magnitudes of λ_i. Due to the overestimated degree, these polynomials have additional roots which are not limited to powers of α (not even to the unit circle), and therefore, these roots are unlikely to result in $\lambda_i = 0$ with $i \notin \mathcal{U}$. Nevertheless, they might potentially result in additional coefficients with small magnitude $|\lambda_i|$. Especially in noisy scenarios, these additional coefficients might interfere with those of actual error positions. The simulation results regarding the performance with respect to the average support recovery are given in Appendix G on page 117 and show the improved performance, where the approach based on the Peterson type algorithm performs slightly better than the other two approaches. Therefore, this variant is used subsequently for the initial values required by the IEE scheme.

In order to evaluate the presented IEE approach in the context of a noisy CS scheme, the squared error $\|\boldsymbol{\chi} - \widehat{\boldsymbol{\chi}}\|^2$ has been determined as before. The result is given as boxplots in Figure 7.5. For comparison, a GZ error evaluator combined with an error overestimating

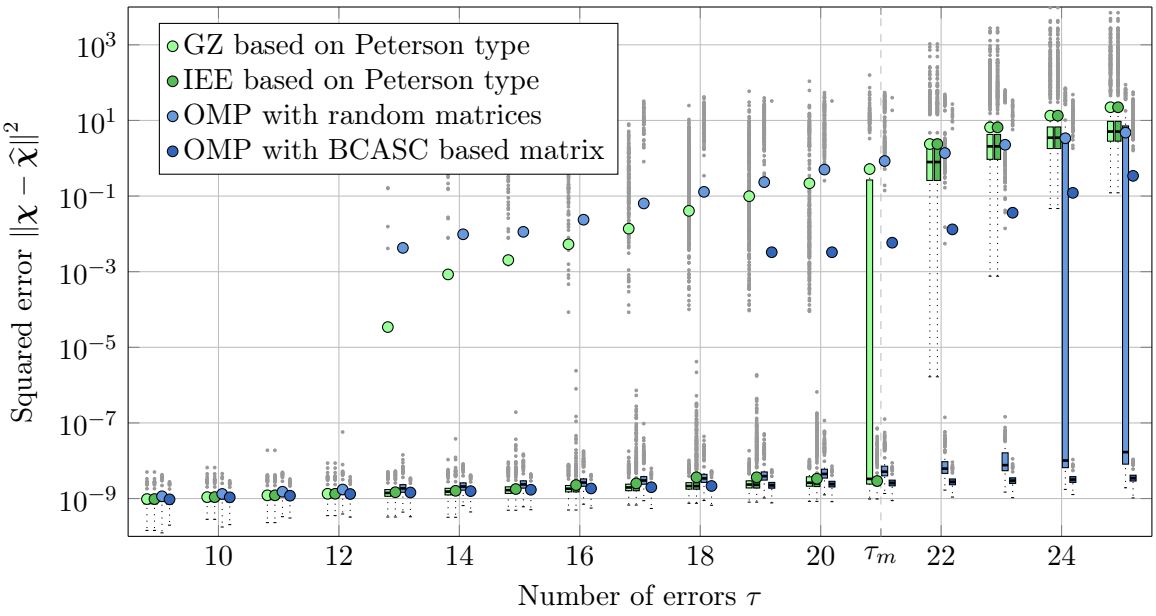


Figure 7.5.: Boxplots illustrating the squared error in CS schemes: Deterministic CRS based scheme (green) with different dedicated algorithms (brightness) and a general OMP algorithm (blue) in combination with different sensing matrices (brightness). Simulation parameters: Appendix H.4.

Peterson type location algorithm is considered as well. Since the possibility to choose the sensing matrix is implicitly assumed for deterministic CS schemes, an optimized sensing matrix $\boldsymbol{\Theta}$ is also investigated. The low-coherence matrix $\boldsymbol{\Theta}$, which is used therefore, is based on the previously described BCASC approach (cf. Chapter 6) with a close to optimum coherence of $0.809 > 0.8028$, cf. (6.3) on page 51. For further comparison, the combined result of 10 000 column-normalized random matrices is examined as well, where the real and imaginary components of each matrix element is drawn from a zero-mean Gaussian random source. For the reconstruction, the OMP is used for the matrices which are not based on CRS codes. As it can be seen from the figure, the IEE approach can decode reliably up to τ_m, while the non-refined GZ error evaluator starts to fail for some scenarios with $\tau > 12$. However, as the median indicates, GZ based decoding is successful most of the time for $\tau \leq \tau_\mathrm{m}$. The conventional CS scheme based on random matrices starts to fails with $\tau > 12$ as the GZ decoding based approach. Naturally, the optimized low-coherence matrix provides better results, where first unsuccessful recoveries occur for $\tau > 18$. However, both non CRS based CS schemes are able to provide correct reconstructions in most cases for $\tau > \tau_\mathrm{m}$ (cf. the median in Figure 7.5). This is not possible for the presented approaches based on CRS decoding. However, in the following section, an approach is introduced which aims to overcome this restriction.

7.3. Continuity Assisted Decoding beyond the Power Decoding Radius

The schemes presented so far are by design only capable of decoding until τ_m errors since they follow the concept of (almost sure) unique decoding. As mentioned before, it is principally possible to decode beyond τ_m in the complex field. With the perspective of channel coding over finite fields, the missing or disadvantageous features of fields over complex numbers are usually spotted and discussed most often. However, the continuity of the Euclidean norm resembles an advantageous feature which is not existing in this way for finite fields. The corresponding distance can be used to measure the closeness between two elements within the complex field while there is no equivalent measure for finite fields. Since the magnitude of a complex number equals its Euclidean distance to zero, this measure provides closeness-based reliability information whether a somehow distorted number might be actually equal to zero. As described in Section 7.1.1 on page 76, the set $\left\{0, \alpha^i \ \forall\ i \in [0, n-1]\right\}$ is not closed under addition. Consequently, the evaluation of a polynomial at some power of $\alpha = \mathrm{e}^{-\mathrm{j}2\pi/n}$ results in a general complex number: $\boldsymbol{\Lambda}(\alpha^i) \in \mathbb{C}$ with $|\boldsymbol{\Lambda}(\alpha^i)| \in \mathbb{R}_{\geq 0}$ for all $i \in [0, n-1]$. Since noise and precision losses introduce small perturbations, the value of $|\lambda_i|$ is already considered in Section 7.2.2 as reliability information for i being an error position. A similar closeness-based reasoning is used in the following to decode beyond the power decoding radius.

As commonly known, the error location can be interpreted as a Padé approximation. In Section 7.3.1, additional closeness-based reliability-like information for CRS codes is obtained with the help of a low-degree Padé approximation in the case of $\tau > \tau_\mathrm{m}$. Subsequently, this information is utilized for the decoding of CRS codes within Section 7.3.2 on page 93.

7.3.1. Low-Degree Padé Approximation Provides Reliability-Like Information

The technique of Padé approximation was developed in the late 19th century by Henri Padé and has been published within his dissertation [Pad92]. However, the general concept has been described before by [Jac46, Fro81]. A brief description based on [BGM96] is given in the following. Refer to standard literature for a detailed introduction and analysis of the Padé approximation, e.g., to the aforementioned book [BGM96].

For a given function $\boldsymbol{f}(z)$ with a power series $\sum_{i=0}^{\infty} f_i z^i$, the Padé approximation $[u/w]_{\boldsymbol{f}}(z)$ is a rational function

$$[u/w]_{\boldsymbol{f}}(z) = \frac{\boldsymbol{a}(z)}{\boldsymbol{b}(z)} = \frac{a_0 + a_1 z + \cdots + a_u z^u}{b_0 + b_1 z + \cdots + b_w z^w},$$

where b_0 is commonly normalized to one[2]. The MacLaurin expansion of $[u/w]_{\boldsymbol{f}}(z)$ coincides with power series of $\boldsymbol{f}(z)$ up to a degree of $u + w$. Consequently, the approximation can be specified equivalently by

$$\boldsymbol{a}(z) - \boldsymbol{f}(z)\boldsymbol{b}(z) \in \mathcal{O}(z^{u+w+1}) \quad \text{as} \quad z \to 0. \tag{7.6}$$

[2]The special case of $b_0 = 0$ is not considered here. Refer to standard literature, e.g., [BGM96], for this case.

Based on (7.6), the approximation can be determined by solving the system of equations:

$$\begin{pmatrix} f_{u-w+1} & f_{u-w+2} & \cdots & f_u \\ f_{u-w+2} & f_{u-w+3} & \cdots & f_{u+1} \\ \vdots & \vdots & & \vdots \\ f_u & f_{u+1} & \cdots & f_{u+w-1} \end{pmatrix} \cdot \begin{pmatrix} b_w \\ b_{w-1} \\ \vdots \\ b_1 \end{pmatrix} = \begin{pmatrix} -f_{u+1} \\ -f_{u+2} \\ \vdots \\ -f_{u+w} \end{pmatrix} \tag{7.7}$$

Subsequently, the coefficients of $\boldsymbol{a}(z)$ can be determined by (7.6):

$$a_i = f_i + \sum_{k=1}^{\min(u,w)} b_k f_{u-k} \quad \forall\ i \in [0, u]$$

Solving the key equation (2.15) on page 12 can be considered as Padé approximation of the syndrome $\boldsymbol{S}(z)$:

$$[(\tau - 1)/\tau]_{\boldsymbol{S}}(z) = \frac{\boldsymbol{\Omega}(z)}{\boldsymbol{\Lambda}(z)}.$$

The equivalence of different error location algorithms to each other and to the Padé approximation is commonly known for RS codes, e.g., [Fit95] (compare also (7.7) to the description of the Peterson algorithm in Section 2.2.2 on page 13). For (almost sure) unique decoding, there must be at least as many independent equations as there are unknowns in order to determine the error locator polynomial by a Padé approximation. In case of $\tau > \tau_{\mathrm{m}}$, the corresponding system of equations cannot be solved. However, it is still possible to determine

$$[(\tau_{\mathrm{m}} - 1)/\tau_{\mathrm{m}}]_{\boldsymbol{S}}(z) = \frac{\widehat{\boldsymbol{\Omega}}(z)}{\widehat{\boldsymbol{\Lambda}}(z)}$$

from the given syndrome. Since the MacLaurin expansion of $[(\tau_{\mathrm{m}} - 1)/\tau_{\mathrm{m}}]_{\boldsymbol{S}}(z)$ equals the syndrome up to a degree of $2\tau_{\mathrm{m}}$, both functions are related to each other. Therefore, the obtained polynomial $\widehat{\boldsymbol{\Lambda}}(z)$ can be interpreted as low-degree approximation of the actual error locator $\boldsymbol{\Lambda}(z)$. Generally, the roots of both polynomials, $\widehat{\boldsymbol{\Lambda}}(z)$ and $\boldsymbol{\Lambda}(z)$, do not coincide. By definition, the roots of $\boldsymbol{\Lambda}(z)$ are located at all τ error locations α^{-u_i} with $u_i \in \mathcal{U}$ for $i \in [1, \tau]$ (cf. (2.12) on page 11). By inspecting the magnitudes of $\widehat{\boldsymbol{\Lambda}}(\alpha^{-i})$ for $i \in [0, n-1]$, the approximation $\widehat{\boldsymbol{\Lambda}}(z)$ can be used to extract reliability-like information: The position corresponding to the smallest magnitude is most likely in error. Consequently, the order of the magnitudes provides a list of least reliable positions which can be used by subsequent erasure decoding algorithms as soft information, and thus, decoding beyond τ_{m} becomes possible. An example for such an approach is given in Section 7.3.2.

As discussed before, erasure decoding can correct up to $\tau = n - k$ errors if all erasures are actual errors ($\mathcal{U} = \mathcal{E}$, cf. Section 2.2.2 on page 12). It is known that a τ-sparse error vector is unique with high probability as long as $\tau < n - k$ (cf. Section 3.2.2 on page 21). There is no contradiction for $\tau = n - k$: The finitely many τ-sparse error vectors are identified by their error positions. It is consequently possible to decode successfully for $\tau = n - k$ as long as the actual error positions are provided.

On a first sight, the previous procedure of selecting the smallest magnitudes $|\lambda_i|$ in noisy scenarios (cf. Section 7.2.2), is very similar to the current approach of approximating the

error locator polynomial and subsequently sorting $|\widehat{\boldsymbol{\Lambda}}(\alpha^{-i})| = |\widehat{\lambda}_i|$. However, it should be noted that the initial assumption is different. Due to noise, the actual roots of $\boldsymbol{\Lambda}(z)$ have values of small magnitude for $\tau \leq \tau_{\mathrm{m}}$. For the currently presented concept, the low degree approximation $\widehat{\boldsymbol{\Lambda}}(z)$ has different roots, but provides small magnitudes at α^{-u_i} with $u_i \in \mathcal{U}$ for $i \in [1, \tau]$. Of course, both effects are superimposed in the case of noise and $\tau > \tau_{\mathrm{m}}$.

The approximated error locator polynomial $\widehat{\boldsymbol{\Lambda}}(z)$, which equals the denominator of a low-degree Padé approximation $[(\tau_{\mathrm{m}} - 1)/\tau_{\mathrm{m}}]_{\boldsymbol{S}}(z)$, can be determined by the error location algorithms stated before. Based on the previously defined noisy scenario (cf. Appendix H.4), the magnitudes of the described low-degree approximations of the error locator polynomial $|\widehat{\boldsymbol{\Lambda}}(z)|$ are plotted with dashed lines over the unit circle ($z = \mathrm{e}^{-\mathrm{j}\kappa}$ with $0 \leq \kappa \leq 2\pi$) for the case of $22 = \tau > \tau_{\mathrm{m}} = 21$ in Figure 7.6. For these approximations, the BM, Peterson and EE type al-

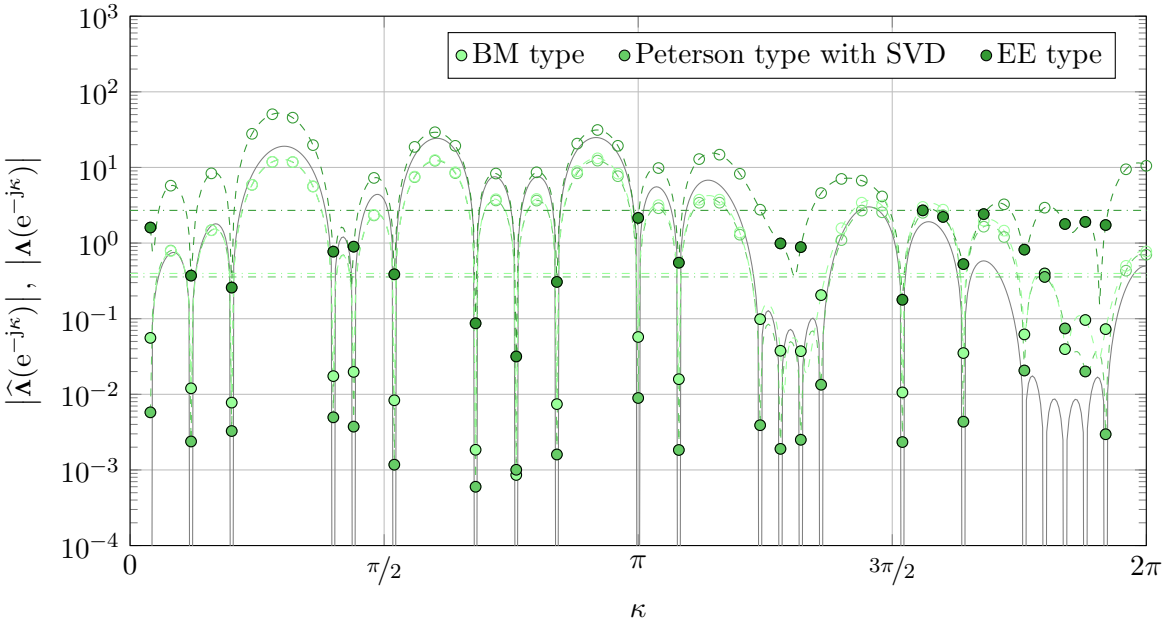


Figure 7.6.: Magnitudes of error locator polynomials on the unit circle for $22 = \tau > \tau_{\mathrm{m}} = 21$. Plots for approximations $\widehat{\boldsymbol{\Lambda}}(z)$ are dashed. Ideal reference $\boldsymbol{\Lambda}(z)$ is gray. Values corresponding to $\widehat{\lambda}_i = \widehat{\boldsymbol{\Lambda}}(\alpha^{-i})$ are marked by circles. Filled marks belong to the τ smallest magnitudes for each approximation. Dash-dotted horizontal lines indicate corresponding boundary levels. Simulation parameters: Appendix H.4.

gorithms for power decoding have been used. The marks within Figure 7.6 indicate the values for $\widehat{\lambda}_i = \widehat{\boldsymbol{\Lambda}}(\alpha^{-i})$, where the filled marks belong to the τ smallest $|\widehat{\lambda}_i|$ for each approximation. The boundary level for which $|\widehat{\lambda}_i|$ belongs to the τ smallest is indicated by a dash-dotted horizontal line. The curve for the correct error locator polynomial is given as reference in gray. As it can be seen from this figure, the approximations which are based on the BM and Peterson type error location algorithms obtain the support with their τ smallest values of $|\widehat{\lambda}_i|$ for this exemplary case with $\tau > \tau_{\mathrm{m}}$. For the EE type algorithm, 19 out of 22 positions correspond to the actual error locations. However, the complete support is contained within the 27 smallest values of $|\widehat{\lambda}_i|$. Therefore, the number of smallest values, which are necessary in order to contain all error positions, is subsequently investigated.

In the following, $\mathcal{P}$ denotes the set of positions corresponding to the smallest values of $|\widehat{\lambda}_i|$ such that all error positions are contained: $\mathcal{U} \in \mathcal{P}$. In order to investigate how many of the smallest values $|\widehat{\lambda}_i|$ are necessary to contain all error positions $\mathcal{U}$, the corresponding statistic based on our previous simulations is given in Figure 7.7. For the non-gray area in

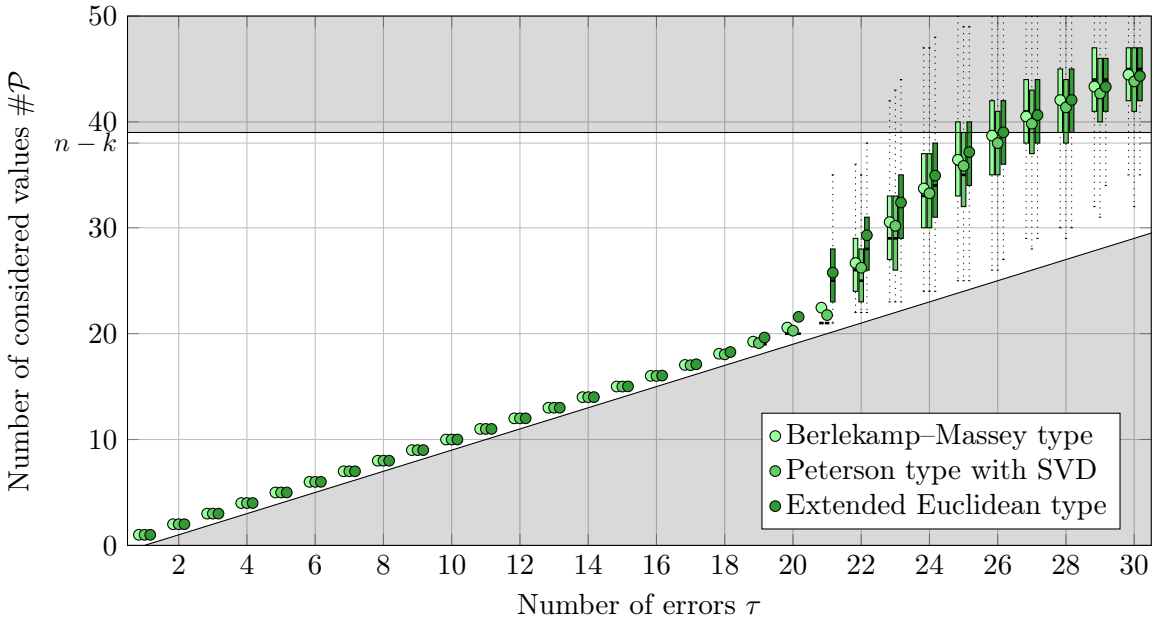


Figure 7.7.: Boxplots illustrating the number of the smallest values $|\lambda_i|$ necessary to contain all error positions. Non-gray area allows application of erasure decoding. Simulation parameters: Appendix H.4.

this figure, erasure decoding can be applied, since $\tau \leq \#\mathcal{P} \leq n-k$. The inequality case $\tau < \#\mathcal{P} \leq n-k$ corresponds to additional erasures, which pose no further problems during erasure decoding. As it can be seen from the figure, decoding is possible in the majority of all cases for a sufficiently small number of errors. However, the assumed soft information is not able to identify the actual error positions for larger numbers of errors including the previously discussed boundary case of $\tau = n-k$.

Table 7.1.: Probability of $\#\mathcal{P} \leq n-k$ in percent. Simulation parameters: Appendix H.4.

Algorithm \ τ	20	21	22	23	24	25	26	27
BM type	100.00	99.99	97.49	90.88	80.40	66.09	49.51	35.20
Peterson type	100.00	100.00	98.16	92.45	83.65	70.35	55.07	40.07
EE type	100.00	98.53	94.80	86.89	75.60	61.38	47.21	33.89

The probability of $\#\mathcal{P} \leq n-k$ is also of interest, and consequently, it is given in Table 7.1 for $19 \leq \tau \leq 25$. As it can be seen from the table and the figure, the Peterson type algorithm is especially successful in providing reliability-like information. The BM and EE type algorithms

also allow to decode beyond τ_{m}, however, their estimated error positions are less reliable. Since the Peterson type algorithm provides the most reliable results, it is considered in the following as source for reliability-like information for subsequent soft decoding.

These results provide the basis for a recently accepted DFG project, in which a colleague investigates the further potential of this continuity based reliability-like information for the decoding of CRS codes and its application in CS. In the following section, we provide an extension of the previously introduced IEE approach in order to decode beyond the power decoding radius.

7.3.2. Exploiting Continuity for CRS-Based Compressed Sensing

As argued in the previous section, the low-degree Padé approximation on the syndrome provides reliability-like information for cases of $\tau > \tau_{\mathrm{m}}$. Based thereon, decoding algorithms can be derived which are based on erasure decoding. The previously introduced IEE approach is, by design, only capable of decoding up to τ_{m}, as almost all algorithms for CRS codes, e.g., [Red00, ADAA08]. In the following, the CAD approach is provided which is capable of decoding beyond τ_{m}. This new algorithm and can be interpreted as a modified and extended version of the IEE concept.

The algorithm described in [ADAA08] as well as the previously given IEE approach consider potentially more than τ_{m} of the smallest magnitudes $|\lambda_i|$. However, both algorithms subsequently limit themselves to τ_{m} correctable errors by determining a set of τ_{m} error positions. The CAD approach does not come with such a limitation. Starting from the initial estimation on the number of errors τ_{initial} and the error locator $\boldsymbol{\Lambda}(z)$, both obtained by the previously described Peterson type error location algorithm, the set of potential error positions is iteratively enlarged based on the smallest magnitudes of $|\lambda_i|$. As soon as a vector $\widetilde{\boldsymbol{\chi}}$ is found for which the syndrome fulfills the validity check (7.5), the set of error positions is not further enlarged. The smallest coefficient of $\widetilde{\boldsymbol{\chi}}$ is canceled from the error position set and the erasure evaluation is repeated. This reduction is repeated until (7.5) is not longer fulfilled. The last $\widetilde{\boldsymbol{\chi}}$ fulfilling the check is returned as the estimated error vector $\widehat{\boldsymbol{\chi}}$. In case the validity check is never passed, the GZ algorithm is applied to the initially estimated error positions. An algorithmic description of the CAD scheme is given in Algorithm 7.1, where $\mathtt{GZ}(\boldsymbol{r}, \mathcal{E})$ denotes the GZ algorithm for a received word $\boldsymbol{r}$ and a set of erasures/error positions $\mathcal{E}$.

There is an important difference between IEE and CAD: For the IEE approach, the set of error positions is reduced until τ_{m} is reached and a single subsequent validity check (7.5) is performed, while for CAD, the check itself is used as stopping criterion for each reduction step. This allows CAD to obtain successful results for more than τ_{m} errors, while the IEE approach always terminates with a result of at most τ_{m} errors. However, this comes also with a potential drawback: The IEE approach might end up with a set of τ_{m} error positions which do not result in a fulfilled validity check (7.5). In such a case, the set of smallest magnitudes $|\lambda_i|$ is increased and the complete reduction is performed again. By this, it is possible for the IEE approach to consider larger position sets which is not possible for CAD, since therein, the position set is not further increased once the validity check (7.5) is fulfilled. Consequently, there are cases for which CAD fails while the IEE succeeds, however, as advantage, CAD is principally able to decode beyond τ_{m}. For a direct comparison, refer to the algorithmic description of the IEE scheme, which is given in Algorithm F.1 on page 115.

The presented CAD approach is evaluated for the previously described noisy CS scheme.

Algorithm 7.1: Continuity Assisted Decoding (CAD) scheme

Input : $\boldsymbol{r}$ /* received word */
$\boldsymbol{\Lambda}(z)$ /* estimate of the error locator polynomial */
τ_{initial} /* estimate on the number of errors */
ϵ /* threshold for verification */
Output: $\widehat{\boldsymbol{\chi}}$ /* estimate of the error vector */

$\boldsymbol{\lambda}(z) \leftarrow \mathscr{F}^{-1}\left[\boldsymbol{\Lambda}(z)\right]$, $\widetilde{\tau}_{\text{outer}} \leftarrow \tau_{\text{initial}}$, $success \leftarrow false$ /* initialization */
$\mathcal{E} \leftarrow$ locations corresponding to the τ_{initial} coefficients with smallest magnitudes $|\lambda_i|$ /* determine default erasures */
$\widehat{\boldsymbol{\chi}} \leftarrow \mathtt{GZ}(\boldsymbol{r}, \mathcal{E})$ /* default result */
while $\widetilde{\tau}_{\text{outer}} \leq n-k$ **and** $success = false$ **do** /* outer loop */
  $\mathcal{E} \leftarrow$ locations corresponding to the $\widetilde{\tau}_{\text{outer}}$ coefficients with smallest magnitudes $|\lambda_i|$ /* determine erasures */
  $\widetilde{\boldsymbol{\chi}} \leftarrow \mathtt{GZ}(\boldsymbol{r}, \mathcal{E})$ /* apply Gorenstein–Zierler */
  $\widetilde{\boldsymbol{s}} \leftarrow \boldsymbol{H}^{\mathrm{T}}\left(\boldsymbol{r} - \widetilde{\boldsymbol{\chi}}\right)$ /* calculate temporary syndrome */
  if $|\widetilde{S}_i| < \epsilon \quad \forall \quad i \in [1, N-K]$ **then** /* validity check */
    $\widetilde{\tau}_{\text{inner}} \leftarrow \widetilde{\tau}_{\text{outer}}$ /* initialization */
    while $|\widetilde{S}_i| < \epsilon \quad \forall \quad i \in [1, N-K]$ **do** /* inner loop */
      $\mathcal{E} \leftarrow$ positions of the $\widetilde{\tau}_{\text{inner}}$ elements with largest magnitude $|\widetilde{\chi}_i|$ /* determine erasures */
      $\widetilde{\boldsymbol{\chi}} \leftarrow \mathtt{GZ}(\boldsymbol{r}, \mathcal{E})$ /* apply Gorenstein–Zierler */
      $\widetilde{\boldsymbol{s}} \leftarrow \boldsymbol{H}^{\mathrm{T}}\left(\boldsymbol{r} - \widetilde{\boldsymbol{\chi}}\right)$ /* calculate temporary syndrome */
      if $|\widetilde{S}_i| < \epsilon \quad \forall \quad i \in [1, N-K]$ **then** /* validity check */
        $\widehat{\boldsymbol{\chi}} \leftarrow \widetilde{\boldsymbol{\chi}}$ /* new candidate */
        $success \leftarrow true$
      $\widetilde{\tau}_{\text{inner}} \leftarrow \widetilde{\tau}_{\text{inner}} - 1$ /* reduce number of erasures */
  $\widetilde{\tau}_{\text{outer}} \leftarrow \widetilde{\tau}_{\text{outer}} + 1$ /* increase number of erasures */
return $\widehat{\boldsymbol{\chi}}$ /* return result */

The result is compared to the performance of the IEE approach by boxplots in Figure 7.8. As for the evaluation of the IEE scheme, an optimized low-coherence matrix has been considered as well for comparison, whereby an OMP has been used, cf. Figure 7.5. The aforementioned cases of too small sets of error positions, for which CAD fails while IEE succeeds, occur within the simulation only for $20 \leq \tau \leq 21$. However, by utilizing the reliability-like information from the low-degree Padé approximation $\widehat{\boldsymbol{\Lambda}}(z)$, the proposed CAD approach is able to recover the sparse vector successfully in most cases for $\tau < 26$. This result fits to the earlier observations of Figure 7.7 and Table 7.1. As a drawback of the proposed CAD approach, an erroneous sparse vector is returned in case of a decoding failure which results in very large squared errors ($> 10^5$). If a-priori knowledge on the expected range of values for the sparse vector exists, these cases could be identified and taken care of by corresponding additional thresholds. By comparing the results of CAD with those of OMP for the case of an optimized low-coherence matrix, it can be observed that OMP provides earlier erroneous results ($\tau = 19$), however,

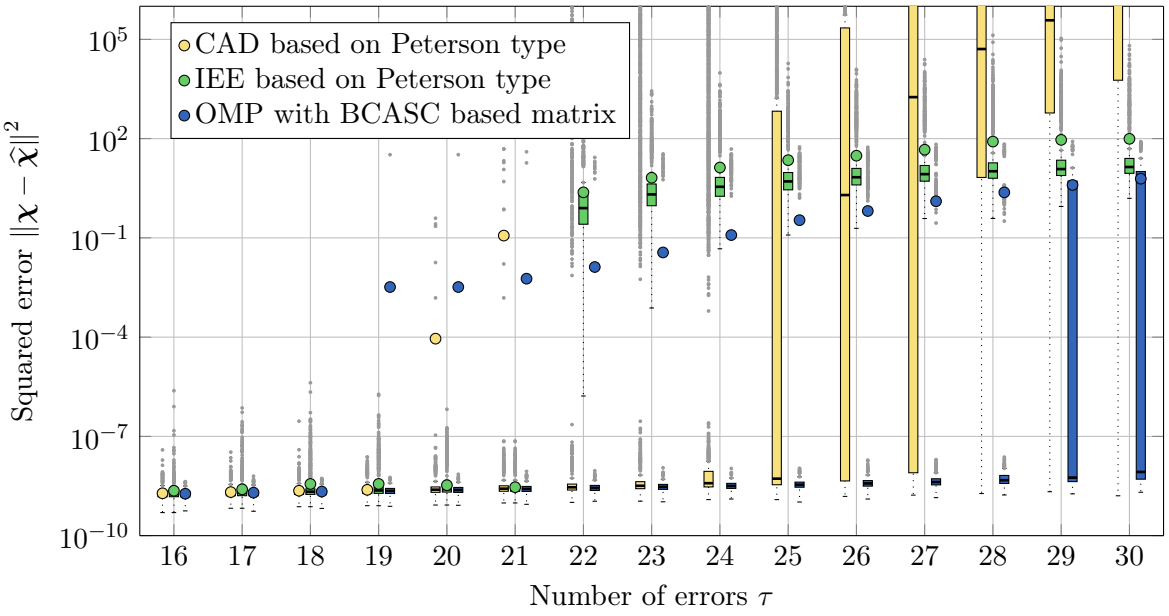


Figure 7.8.: Boxplots illustrating the squared error in CS schemes: Deterministic CRS based scheme with different dedicated algorithms CAD (yellow) and IEE (green) and a general OMP algorithm (blue) in combination with an optimized low-coherence sensing matrix. Simulation parameters: Appendix H.4.

for a larger number of errors $\tau \geq 25$, OMP succeeds more often.

Based on a low-degree Padé approximation, the possibility of decoding beyond τ_{m} is illustrated by CAD. Future improvements should focus on increasing the reliability of the decoding results for $\tau \geq 20$ and on the reduction of the squared errors in case of decoding failures.

The choice between CRS based sensing matrices (e.g., with CAD) and optimized low-coherence matrices (e.g., with OMP) corresponds to a computational trade-off: For CRS bases systems, the reconstruction is of higher complexity, while for low-coherence schemes, the matrix optimization is typically computational demanding.

7.4. Summary and Overview

Within this contribution, power decoding of CRS codes has been introduced. Furthermore, CAD has been proposed, which allows to decode beyond the power decoding radius by utilizing a low-degree Padé approximation. In a first step, CRS codes have been introduced within Section 7.1. A discussion of the commonalities and differences to RS codes followed. Subsequently, the connection to deterministic CS has been established, which provides the motivation to investigate and improve on CRS decoding schemes. After pointing out the consequences of floating-point implementations with finite-precision, the power decoding algorithms for RS codes, which are introduced in Section 2.2.2 on page 11 (namely the Peterson, BM, and EE type algorithms), have been adapted to CRS codes in Section 7.2. Due to finite-precession and noise, the common separation in error location and evaluation is not necessarily optimal for CRS decoding implementations, since the optimality of the individual

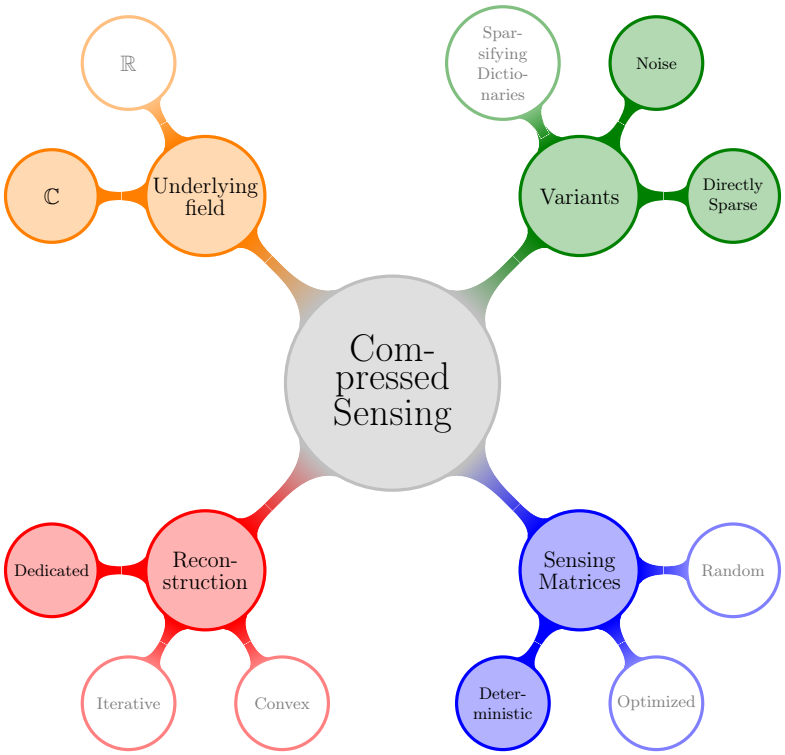


Figure 7.9.: Classification of this chapter's contents within the overall picture of Compressed Sensing (cf. Figure 4.1 on page 33).

steps cannot be guaranteed. Consequently, joint approaches, which combine error location and evaluation, are potentially more powerful. As an example, the IEE approach has been subsequently discussed which aims to determine the error positions iteratively. This is accomplished by checking the validity of the estimated codeword by verifying the corresponding syndrome. In case of a negative verification, the set of estimated error positions is iteratively increased based on the estimate of the error locator polynomial. All previous CRS decoding schemes are intrinsically limited to decode only until $\tau \leq \tau_{\mathrm{m}}$. In Section 7.3, it is explained how a low-degree Padé approximation can be used to provide reliability-like soft information for $\tau_{\mathrm{m}} < \tau$ which allows to decode even beyond τ_{m}. This is exploited by the proposed CAD approach. The recovery performance of the presented decoding variants and their potential compared to other noisy CS has been numerically evaluated by corresponding simulations.

The proposed CAD approach resembles a proof of concept and provides a first impression on the potential of the observed reliability-like soft information which is based on a low-degree Padé approximation. There is still space for further improvements during future research. For example, CAD utilizes a threshold for the validity check, cf. (7.5) and Algorithm 7.1, however, it might be more promising to apply step detection methods which observe the variation of $\widetilde{\boldsymbol{s}}$ in the outer loop. Furthermore, the reliability-like information from $\widehat{\boldsymbol{\Lambda}}(z)$ might be combined with the magnitudes of the sparse vector components $|\widetilde{\chi}_i|$, where they are treated separately in IEE or CAD (cf. reduction in Algorithm 7.1). Additionally, the potential of further soft-information decoding algorithms in combination with the observed reliability-like information needs to be investigated. Currently, Mostafa Hosni Mohamed, a

valued colleague with whom CRS codes have been researched, pursues these new research approaches.

The efforts of the current chapter on deterministic CRS-based CS schemes are put into the overall context of CS in Figure 7.9. As the name of CRS codes suggests, the complex field $\mathbb{C}$ has been considered as underlying field for all operations. The described sensing scheme relies on CRS codes: The transposed parity check matrices, which correspond to partial DFT matrices, are utilized as deterministic sensing matrices. Conventional decoding methods for (complex) RS codes result in dedicated reconstruction algorithms which exploit the structure of the used sensing matrices. Since the focus of this chapter is on dedicated algorithms and the exploitation of the embedded matrix structure, the performance of general reconstruction methods has not been further discussed. Since the structure of CRS based sensing matrices typically hinders such general CS reconstruction algorithms, their application is possible but not recommendable. Due to the fixed deterministic sensing matrices, sparsifying dictionaries cannot be applied, and therefore, directly sparse scenarios are considered. Since the robustness is an often described problem for CRS decoding algorithms, it is explicitly considered in this chapter. As it has been demonstrated by simulations, the presented schemes are capable of withstanding additional noise of moderate intensity.

8

Concluding Remarks

Within this thesis, three independent contributions to CS have been presented. Each contribution covers different aspects or variants in the large field of CS. The presented work is mainly the result of interdisciplinary research which investigated methods from channel coding for the application in CS. In the following, the three contributions are individually concluded.

Sparsity aware simplex methods of Chapter 5 resemble an interesting approach and are less motivated by channel coding. The BP approaches a sparse recovery problem by an LP, where the ℓ_1-norm is used as convex relaxation. Standard solvers can be used to solve these problems. However, the relaxation might fail with an increased number of non-zero components, and consequently, BP fails as well. Sparsity aware simplex methods are based on the observation that the sparsest solution is located at a degenerated vertex of the polytope describing the solution space. As shown in Chapter 3, such a degenerated vertex is unique with high probability for systems in general position. Since the simplex method passes several vertices on its way to the optimum, degeneracy can be used to identify the actual solution for the sparse recovery problem, and thereby, to improve the reconstruction performance of BP. Future work might apply the presented approach to more elaborate variants of the simplex method and concern noisy or complex CS scenarios.

Within Chapter 6, an approach for maximizing the minimal distance of real-valued spherical codes is extended to coherence optimization and is generalized to complex vector spaces by introducing BCASCs. The proposed search approach outperforms other algorithms and can be used for the construction of low-coherence sensing matrices for CS which lead to increased reconstruction performance. Furthermore, this approach allows the adaptation of a measurement matrix with respect to a given dictionary by a simple modification. Due to the generality of the coherence, potential applications are not limited to CS. The proposed search approach relies on an integral which is currently solved numerically. Future work might focus on advanced numerical integration methods or even on an analytical solution.

Chapter 7 considers deterministic CS schemes in noisy environments which are based on CRS codes and corresponding decoding techniques. Within this contribution, well-known RS power decoding approaches have been adapted for CRS codes. Furthermore, low-degree Padé-approximation can be used for CRS codes in order to decode beyond the power decoding radius which is not possible for conventional RS codes. Future work might focus on applying soft-decoding strategies of RS codes in order to exploit the reliability-like information obtained by the Padé-approximation as planed for a recently accepted DFG project.

Appendices

A. Pseudocode for the Schmidt–Sidorenko Algorithm **103**

B. Simulation Results for Evaluation of Sparsity Aware Simplex Methods **105**

C. Further Results on Coherence Optimization **107**

D. Simulation Results for Evaluation of Measurement Adaptation **111**

E. Iterative BCASC Search Approach for Measurement Matrix Adaptation **113**

F. Iterative Erasure and Evaluation Scheme **115**

G. Simulation Results for Error Location Algorithms Based on Overestimation **117**

H. Simulation Details **119**
H.1. General Settings 119
H.1.1. Evaluation of CS Schemes 119
H.1.2. Reconstruction Algorithms 120
H.2. Settings for Sparsity Aware Simulations 120
H.3. Settings for B(C)ASC Based Coherence Optimization 120
H.4. Settings for Simulations Based on CRS Codes 121

I. Boxplots **123**

A

Pseudocode for the Schmidt–Sidorenko Algorithm

The SS algorithm for varying length multi-sequence LFSR synthesis [SS06] is given in Algorithm A.1 for completeness and is discussed on page 13. For its description, interim buffer variables are used for each sequence $\boldsymbol{s}^{(l)}$ with $l \in [1, l_{\max}]$ which are denoted by the superscript $^{(l)}$. Temporary variables are denoted with a tilde.

Algorithm A.1: Schmidt–Sidorenko algorithm [SS06]

Input : $\boldsymbol{s}^{(l)} = \left\{ s_i^{(l)} \right\}_{i=0}^{n-k^{(l)}-1} \quad \forall\ l \in [1, l_{\max}],\ n,\ k$

Output: $\boldsymbol{\Lambda}(z)$

$\nu \leftarrow 0,\ \boldsymbol{\Lambda}(z) \leftarrow 1$ /* initial shift register */

$\mu^{(l)} \leftarrow k^{(l)} - k,\ \nu^{(l)} \leftarrow 0 \quad \forall\ l \in [1, l_{\max}]$ /* initialize buffers for all $l_{\max}$ sequences */

$\boldsymbol{\Lambda}^{(l)}(z) \leftarrow 0,\ \Delta^{(l)} \leftarrow 1 \quad \forall\ l \in [1, l_{\max}]$

for *each* i *from* 0 *to* $n-k-1$ **do**

  for *each* l *from* 1 *to* $l_{\max}$ **do**

    if $i > k - k^{(l)} + \nu$ **then** /* ensure causality */

      $\Delta \leftarrow S_{i+k^{(l)}-k}^{(l)} + \sum_{\xi=1}^{\nu} \Lambda_\xi S_{i+k^{(l)}-k-\xi}^{(l)}$ /* calculate discrepancy */

      if $\Delta \neq 0$ **then** /* shift-register has to be modified */

        if $i - \mu^{(l)} \leq \nu - \nu^{(l)}$ **then** /* no prolongation necessary */

          $\boldsymbol{\Lambda}(z) \leftarrow \boldsymbol{\Lambda}(z) - \frac{\Delta}{\Delta^{(l)}} z^{i-\mu^{(l)}} \boldsymbol{\Lambda}^{(l)}(z)$ /* modification */

        else

          $\widetilde{\nu} \leftarrow \nu,\ \widetilde{\boldsymbol{\Lambda}}(z) \leftarrow \boldsymbol{\Lambda}(z)$ /* temporary variables */

          $\boldsymbol{\Lambda}(z) \leftarrow \boldsymbol{\Lambda}(z) - \frac{\Delta}{\Delta^{(l)}} z^{i-\mu^{(l)}} \boldsymbol{\Lambda}^{(l)}(z)$ /* modification */

          $\nu \leftarrow i - \mu^{(l)} + \nu^{(l)}$ /* update shift-register length */

          $\nu^{(l)} \leftarrow \widetilde{\nu},\ \boldsymbol{\Lambda}^{(l)}(z) \leftarrow \widetilde{\boldsymbol{\Lambda}}(z)$ /* update interim buffers */

          $\Delta^{(l)} \leftarrow \Delta,\ \mu^{(l)} \leftarrow i$

B

Simulation Results for Evaluation of Sparsity Aware Simplex Methods

Additionally to the success rate plot of Figure 5.5 on page 46, boxplots for the squared error are given in Figure B.1 for different levels of sparsity τ. The increasing performance

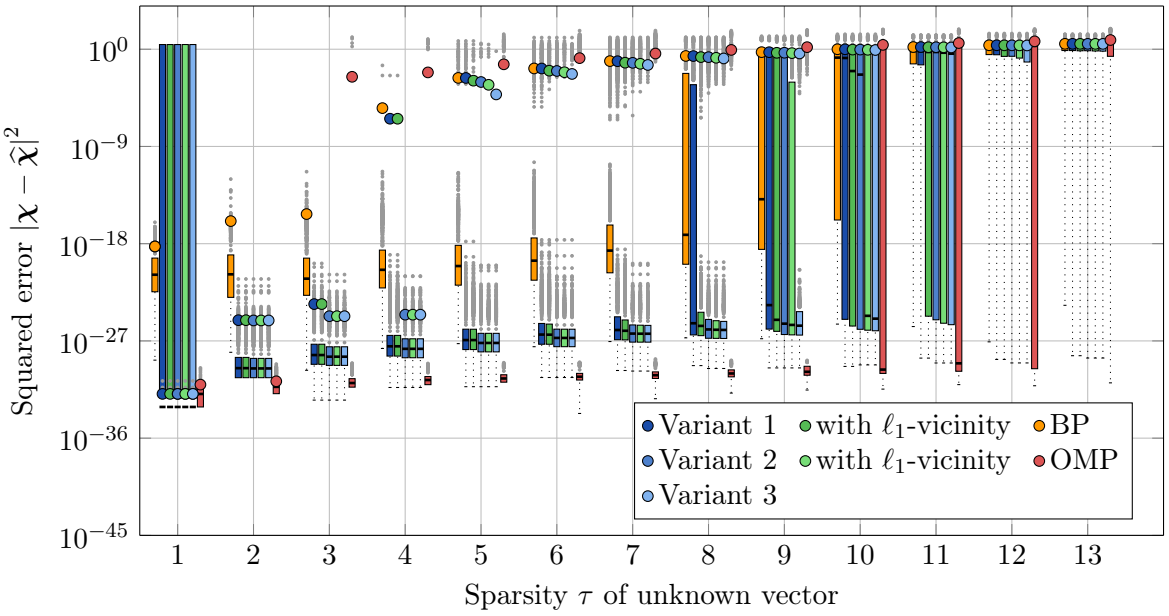


Figure B.1.: Evaluation of sparsity aware simplex methods. Boxplots illustrating the squared error of the reconstruction algorithms over the sparsity. Direct path search is used as SASA in each first phase. In the second phase, the used SASA is denoted by the legend. BP (orange) and OMP (red) are given as reference.

of the more complex sparsity aware simplex methods is also clear from this figure. As for the success rate plot of Figure 5.5, it can be observed that OMP starts to return erroneous results already at $\tau = 3$. BP and the sparsity aware simplex methods based on the direct path search (variant 1) follow at $\tau = 4$, where the other sparsity aware approaches start to fail with $\tau = 5$. Starting with $\tau = 10$, the OMP provides more successful results than the other reconstruction algorithms, which is even more clear at $\tau = 11$ if the median is observed.

C

Further Results on Coherence Optimization

Additionally to the illustrations on the performance of the coherence optimization given in Figures 6.1 and 6.2 on page 63 and on page 64, the corresponding results for $m = 6, \cdots, 10$ are provided in Figures C.1 to C.3. As argued before, the plateau-like level for $m^2 < n < m^2 + m$

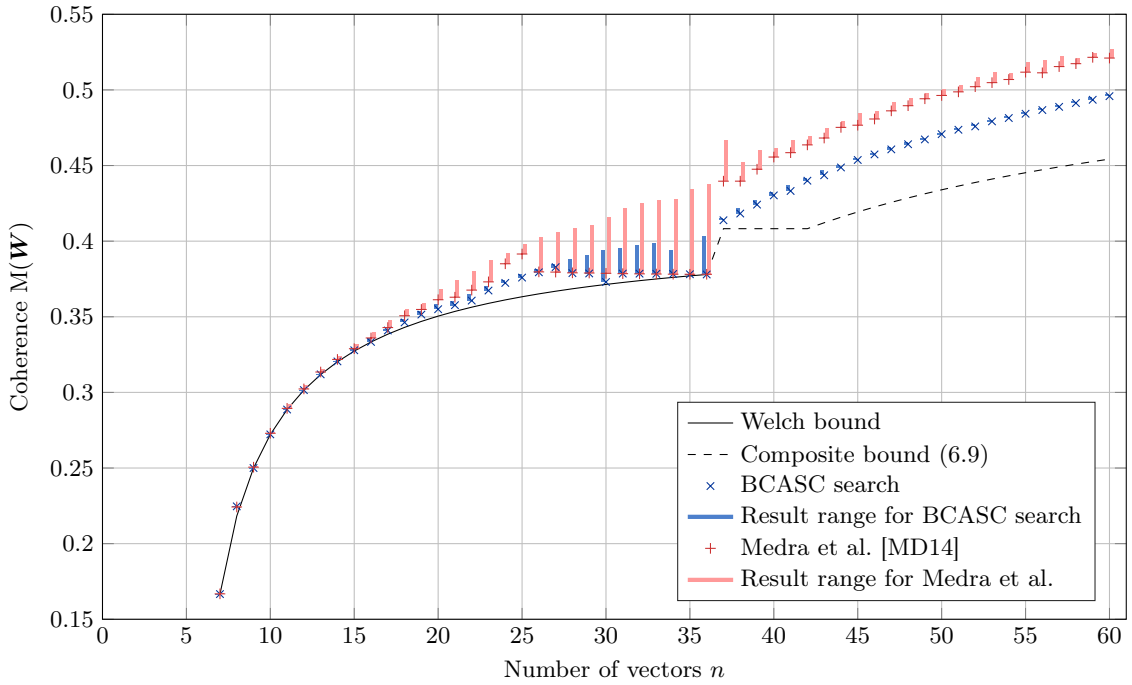
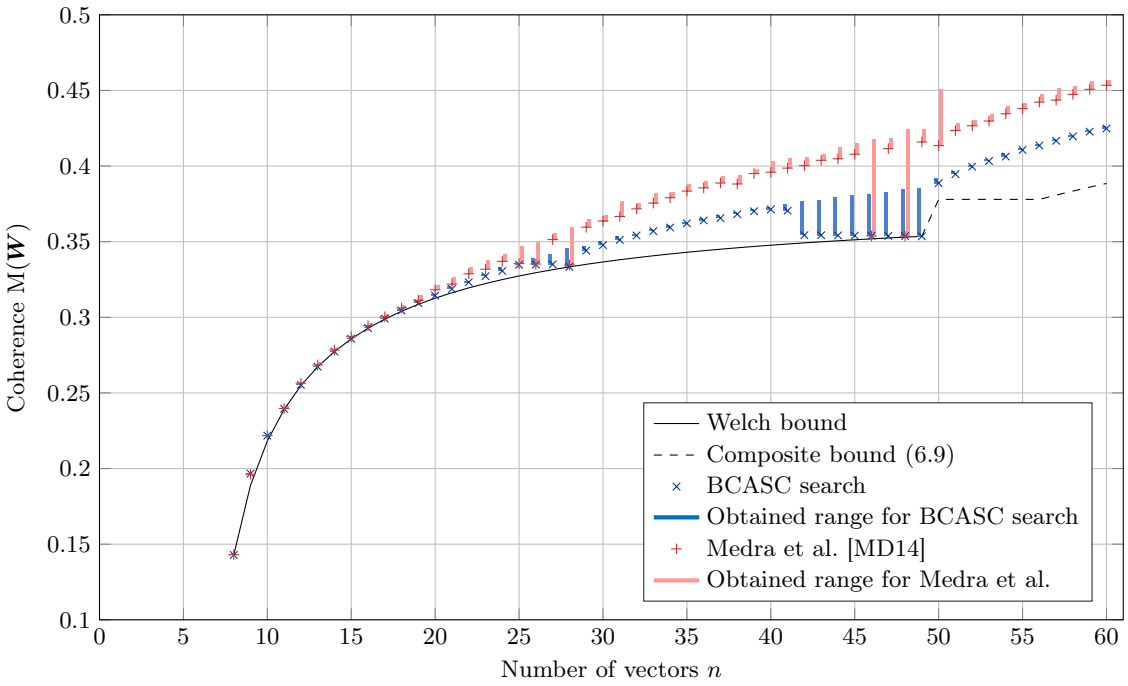


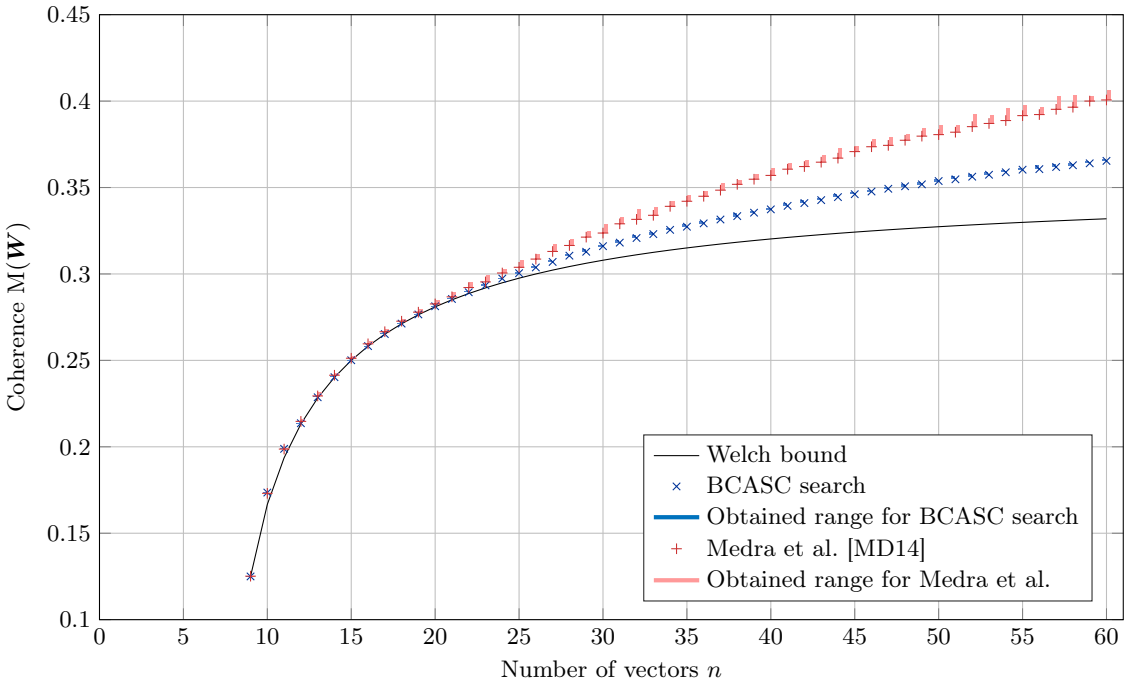
Figure C.1.: Best coherence out of ten runs for varying number of vectors n in $m = 6$ dimensions. Vertical bars indicate the range of obtained coherence values. The lower bound is drawn solid if the Welch bound (6.5) is fulfilled.

has not been found by the described methods for higher dimensions. Since $m = 6$ is not a prime power, it is already questioned in [WF89] whether a vector set of $n = 42$ vectors exists which achieves equality in (6.6) and (6.7).

With increasing dimensions, the approach of [MD14] is less often able to obtain vector sets with a coherence comparable to the BCASC search approach and the achieved coherence values are more stable as indicated by the vertical bars.

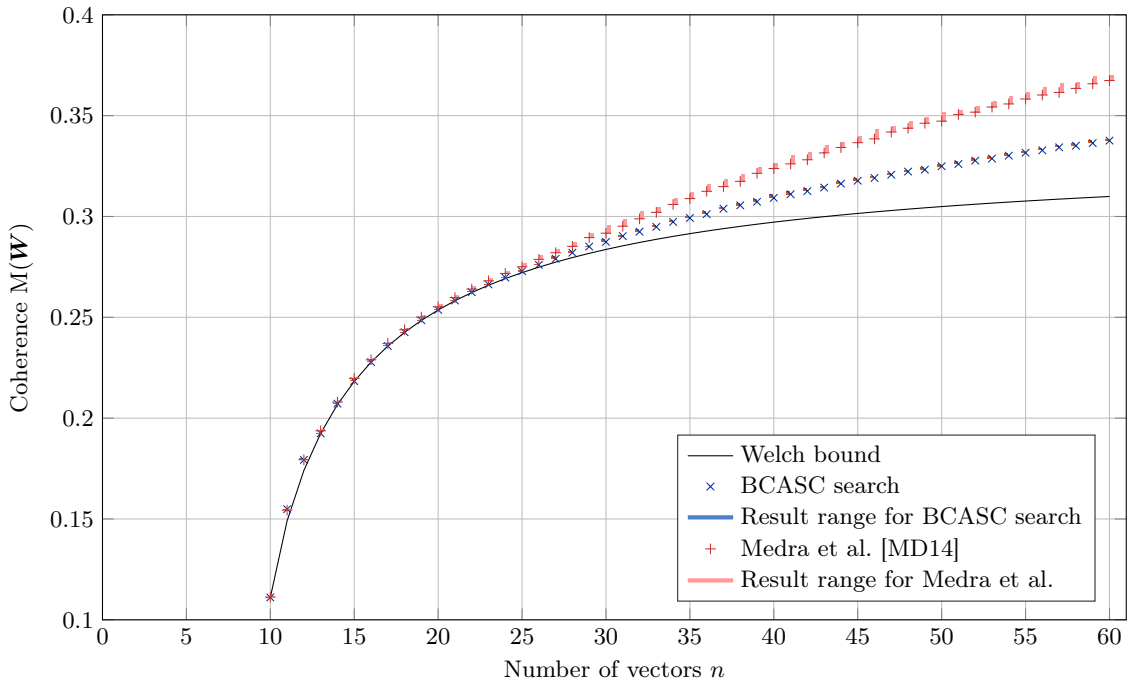


(a) $m = 7$

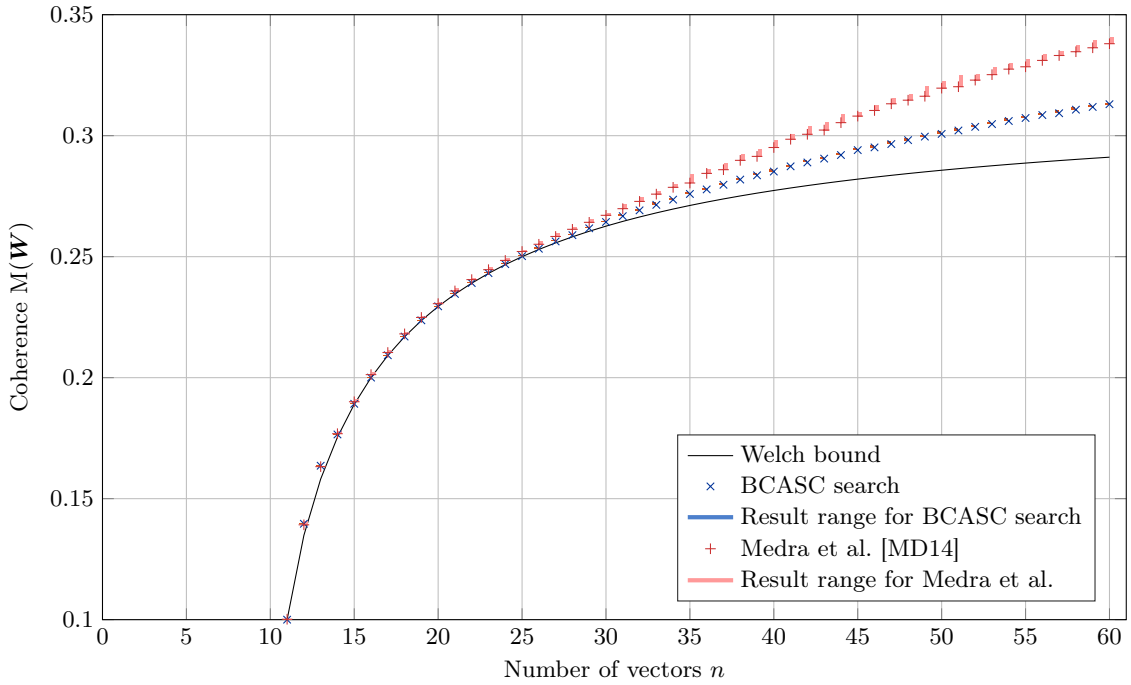


(b) $m = 8$

Figure C.2.: Best coherence out of ten runs for varying number of vectors n in m dimensions. Vertical bars indicate the range of obtained coherence values. The lower bound is drawn solid if the Welch bound (6.5) is fulfilled.



(a) $m = 9$



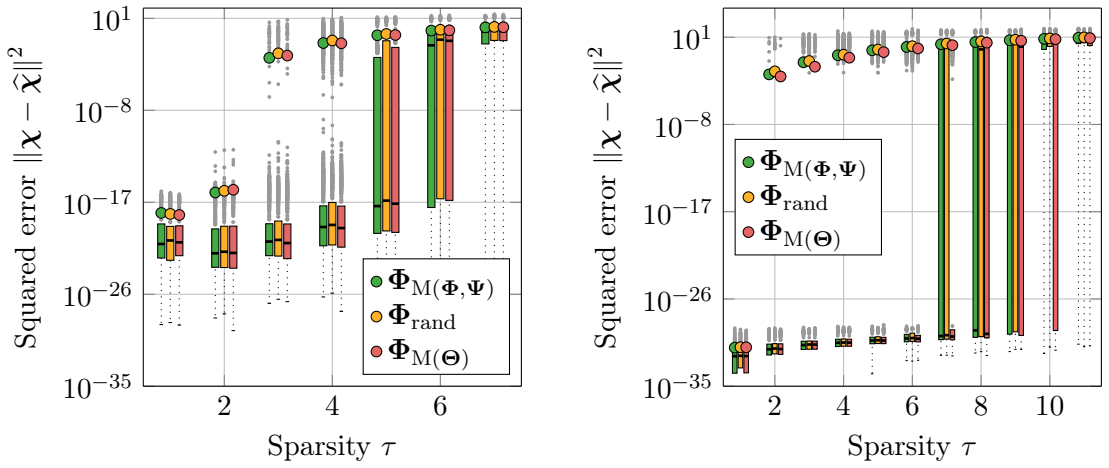
(b) $m = 10$

Figure C.3.: Best coherence out of ten runs for varying number of vectors n in m dimensions. Vertical bars indicate the range of obtained coherence values. The lower bound is drawn solid if the Welch bound (6.5) is fulfilled.

D

Simulation Results for Evaluation of Measurement Adaptation

Additionally to the success rate plot of Figure 6.7 on page 72, boxplots for the squared error are given in Figure D.1 for BP and OMP with respect to the sparsity τ. As it can be seen



(a) BP (b) OMP

Figure D.1.: Evaluation of CS schemes based on $\boldsymbol{\Psi}_{[\mathrm{I,DCT}]}$ and different measurement matrices. Boxplots illustrate the squared error of the reconstruction algorithms over the sparsity.

in this figure, the results are very close for the BP, where successful reconstruction is always achieved for $\tau < 3$. The differences of the proposed measurement matrices are more obvious for the case of OMP, for which the reconstruction has been always successful only in case of one single non-zero value. For $\tau = 7$, the majority of reconstruction are still successful for $\boldsymbol{\Phi}_{\mathrm{M}(\boldsymbol{\Theta})}$, in contrast to the other measurement matrices. For $\tau = 10$, the same trend can be observed, where the other matrices clearly fail for the majority of runs.

E

Iterative BCASC Search Approach for Measurement Matrix Adaptation

In Algorithm E.1, the optimization of $\mathrm{M}(\boldsymbol{\Phi}, \boldsymbol{\Psi})$ is summarized, cf. Section 6.3.2 on page 70. Thereby, the rows of the measurement matrix $\boldsymbol{\Phi} \in \mathbb{C}^{m \times l}$ are denoted by $\boldsymbol{\phi}_i \; \forall \; i \in [1, m]$, while the columns of the dictionary $\boldsymbol{\Psi} \in \mathbb{C}^{l \times n}$ are given by $\boldsymbol{\psi}_i \; \forall \; i \in [1, n]$.

Algorithm E.1: Iterative BCASC search approach for measurement matrix adaptation

Input : $\left\{\underline{\boldsymbol{s}}_p\right\}_{p=1}^{m} \leftarrow \left\{\underline{\boldsymbol{\phi}}_p\right\}_{p=1}^{m}$,
$\left\{\underline{\boldsymbol{s}}_p\right\}_{p=m+1}^{m+n} \leftarrow \left\{\underline{\boldsymbol{\psi}}_p\right\}_{p=1}^{n}$ /* $m+n$ codewords of in l dimensions */

Output: C_s /* optimized spherical code */

$\alpha_{\text{init}} \leftarrow 0.9$, $\epsilon \leftarrow 10^{-4}$ /* exemplary numerical parameters */
$\nu \leftarrow 2$, $\nu_{\max} \leftarrow 2^{10}$
$i_{\max} \leftarrow 10^5$, $\alpha \leftarrow \alpha_{\text{init}}$
while $\nu < \nu_{\max}$ **do**
    FixedPoint $\leftarrow$ **false** /* initialize indicator */
    $i \leftarrow 0$ /* initialize iteration counter */
    while $i < i_{\max}$ **and** FixedPoint $=$ **false** **do**
        for $p \leftarrow 1$ **to** m **do** /* for each vector */
            $\boldsymbol{f}_p \leftarrow \int_{\kappa=0}^{2\pi} \sum_{q \neq p} \frac{\underline{\boldsymbol{s}}_p - \underline{\boldsymbol{s}}_q e^{i\kappa}}{\left\| \underline{\boldsymbol{s}}_p - \underline{\boldsymbol{s}}_q e^{i\kappa} \right\|^{\nu}} d\kappa$ /* calculate superimposed forces */
        $\{\boldsymbol{s}_p\}_{p=1}^{m} \leftarrow \left\{\underline{\boldsymbol{s}_p + \alpha \boldsymbol{f}_p}\right\}_{p=1}^{m}$ /* apply accumulated forces */
        if $\left\| \underline{\boldsymbol{f}}_p - \underline{\boldsymbol{s}}_p \right\| < \epsilon \; \forall \; p \in [1, m]$ **then** /* check for fixed point */
            FixedPoint $\leftarrow$ **true** /* stop loop and proceed */
        $i \leftarrow i + 1$
    $\nu \leftarrow 2\nu$ /* adjust free parameter */
    $\alpha \leftarrow \frac{\alpha_{\text{init}}}{\nu - 1}$ /* adjust damping factor */
return $C_s \leftarrow \left\{\underline{\boldsymbol{s}}_p\right\}_{p=1}^{m}$ /* return obtained spherical code */

F

Iterative Erasure and Evaluation Scheme

An algorithmic description of the IEE scheme is given in Algorithm F.1, where $\texttt{GZ}(\boldsymbol{r}, \mathcal{E})$ denotes the GZ algorithm for a received word $\boldsymbol{r}$ and a set of erasures/error positions $\mathcal{E}$.

Algorithm F.1: Iterative Erasure Evaluation (IEE) scheme [Riz14, MRZB15]

Input : $\boldsymbol{r}$ /* received word */
$\widehat{\boldsymbol{\Lambda}}(z)$ /* estimate of the error locator polynomial */
$\widehat{\tau}_{\text{initial}}$ /* estimate on the number of errors */
τ_{m} /* maximum number of correctable errors */
ϵ /* threshold for verification */
Output: $\widehat{\boldsymbol{\chi}}$ /* estimate of the error vector */

$\boldsymbol{\lambda}(z) \leftarrow \mathscr{F}^{-1}\left[\boldsymbol{\Lambda}(z)\right]$, $\widetilde{\tau}_{\text{outer}} \leftarrow \widehat{\tau}_{\text{initial}}$ /* initialization */
while $\widetilde{\tau}_{\text{outer}} \leq n - k$ **do** /* outer loop */
  $\mathcal{E} \leftarrow$ locations corresponding to the $\widetilde{\tau}_{\text{outer}}$ coefficients with smallest magnitudes $|\lambda_i|$ /* determine erasures */
  $\widetilde{\boldsymbol{\chi}} \leftarrow \texttt{GZ}(\boldsymbol{r}, \mathcal{E})$ /* apply Gorenstein–Zierler */
  $\widetilde{\boldsymbol{s}} \leftarrow \boldsymbol{H}^{\mathrm{T}}\left(\boldsymbol{r} - \widetilde{\boldsymbol{\chi}}\right)$ /* calculate temporary syndrome */
  if $|\widetilde{S}_i| < \epsilon \quad \forall \quad i \in [1, N-K]$ **then** /* validity check */
    if $\widetilde{\tau}_{\text{outer}} \leq \tau_{\text{m}}$ **then**
      return $\widehat{\boldsymbol{\chi}} \leftarrow \widetilde{\boldsymbol{\chi}}$ /* terminate algorithm */
    else
      $\widetilde{\tau}_{\text{inner}} \leftarrow \widetilde{\tau}_{\text{outer}}$ /* initialization */
      while $\widetilde{\tau}_{\text{i}} > \tau_{\text{m}}$ **do** /* inner loop */
        $\widetilde{\tau}_{\text{inner}} \leftarrow \widetilde{\tau}_{\text{inner}} - 1$ /* reduce number of erasures */
        $\mathcal{E} \leftarrow$ positions of the $\widetilde{\tau}_{\text{inner}}$ elements with largest magnitude $|\widetilde{x}_i|$ /* determine erasures */
        $\widetilde{\boldsymbol{\chi}} \leftarrow \texttt{GZ}(\boldsymbol{r}, \mathcal{E})$ /* apply Gorenstein–Zierler */
      $\widetilde{\boldsymbol{s}} \leftarrow \boldsymbol{H}^{\mathrm{T}}\left(\boldsymbol{r} - \widetilde{\boldsymbol{\chi}}\right)$ /* calculate temporary syndrome */
      if $|\widetilde{S}_i| < \epsilon \quad \forall \quad i \in [1, N-K]$ **then** /* validity check */
        return $\widehat{\boldsymbol{\chi}} \leftarrow \widetilde{\boldsymbol{\chi}}$ /* terminate algorithm */
  $\widetilde{\tau}_{\text{outer}} \leftarrow \widetilde{\tau}_{\text{outer}} + 1$ /* increase number of erasures */
$\mathcal{E} \leftarrow$ locations corresponding to the $\widehat{\tau}_{\text{initial}}$ coefficients with smallest magnitudes $|\lambda_i|$ /* determine default erasures */
return $\widehat{\boldsymbol{\chi}} \leftarrow \texttt{GZ}(\boldsymbol{r}, \mathcal{E})$ /* return default result */

G

Simulation Results for Error Location Algorithms Based on Overestimation

As described in Section 7.2.2 on page 87, the presented error location algorithms can be utilized such that the degree of $\widehat{\boldsymbol{\Lambda}}(z)$ is overestimated to τ_{m}. Based on these error locator polynomials, the error positions are determined by the $\widehat{\tau}$ time-domain coefficients $|\lambda_i|$ of smallest magnitude, where the estimated number of errors $\widehat{\tau}$ is determined over the SVD-based Peterson type algorithm. For the BM and EE type algorithms, the overestimated error locator polynomials are obtained by setting the corresponding thresholds to sufficiently small values. The used values are given in Table H.2. For the Peterson type algorithm, the least squares solution to (2.17) on page 12 is obtained with $\tau = \tau_{\mathrm{m}}$ in order to overestimate the degree of $\boldsymbol{\Lambda}(z)$.

The performance comparison of these algorithms with respect to the average support recovery is given in Figure G.1 on the next page. All algorithms show similar recovery results, where the approach based on the Peterson type algorithm performs slightly better than the other two approaches. In comparison to Figure 7.3 on page 84, it can be observed that the overestimation also improved the estimation on the number of errors by the Peterson type algorithm, where the tendency to underestimation is reduced, which is particularly helpful for the extensions based thereon.

In Figure G.2, the performance of the two presented error evaluation algorithms is given with respect to the squared error. Compared to Figure 7.4 on page 85, the obtained results are more stable and less affected by the noise. However, the numerical drawbacks of the Forney algorithm remain.

Due to the overestimation, more equations are utilized in the subsequent calculations, which improves the robustness especially for demanding noisy scenarios. This highlights also an important difference between the decoding of conventional RS codes and the investigated CRS codes: For RS codes, the calculations are always exact and non-disturbed, and consequently, the main research focus is on reducing the necessary number of calculations (complexity) or on increasing the decoding radius. For CRS codes, robustness is an additional criterion which needs to be considered, whereby a trade-off between complexity and robustness is unavoidable. For example, the GZ approach is computationally more expensive than error evaluation by Forney, however, it is also significantly more robust, as it can be seen in Figure G.2.

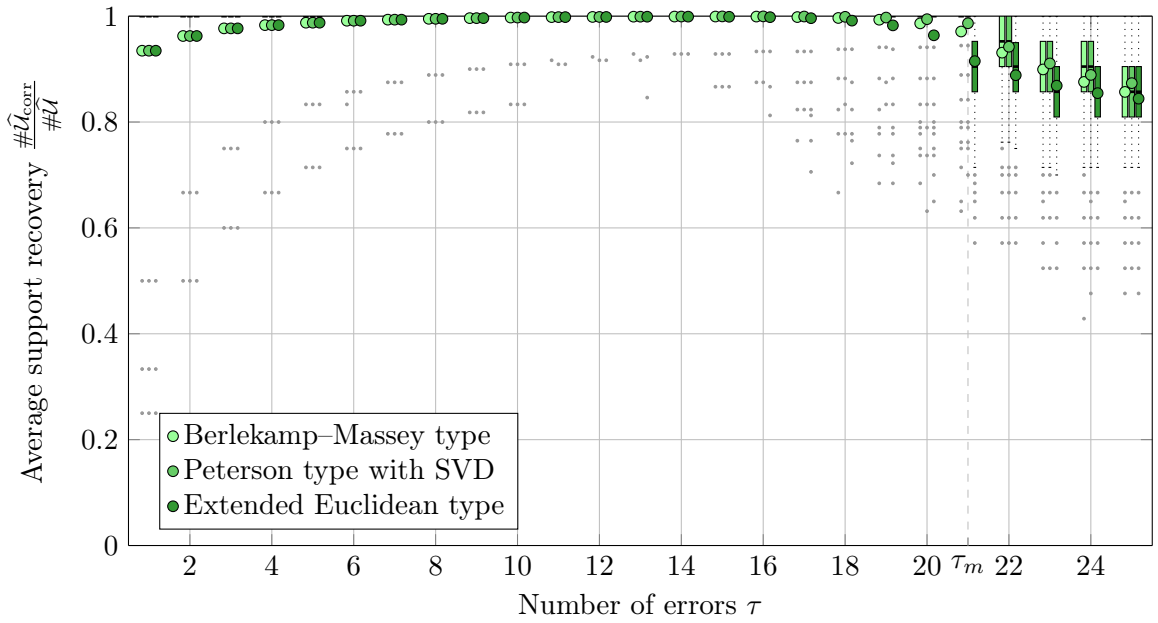


Figure G.1.: Boxplots illustrating the average support recovery provided by the presented error-overestimating power decoding error location algorithms. Simulation parameters: Appendix H.4.

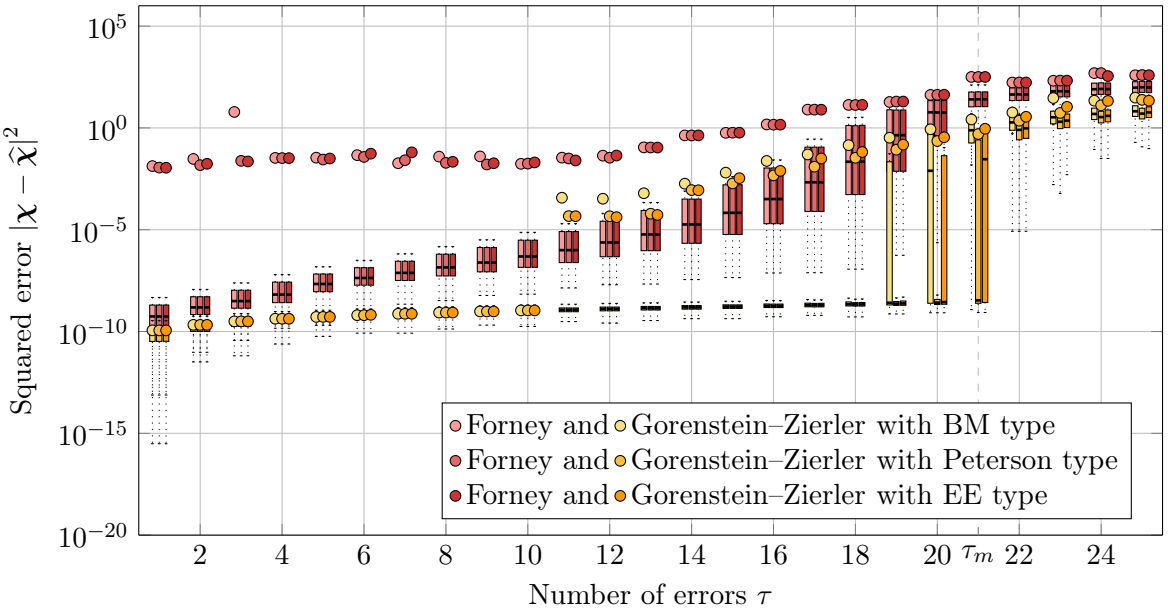


Figure G.2.: Boxplots illustrating the squared error for the presented power decoding error evaluation algorithms: Forney (red) and GZ (orange), in combination with the presented error-overestimating power decoding error location algorithms (brightness). Outliers are not explicitly shown. Simulation parameters: Appendix H.4.

H

Simulation Details

In order to provide both, readability and accuracy, details regarding simulation parameters are given separately in this appendix.

Statistical results are often visualized by boxplots. Therefore, a brief introduction to boxplots is provided in Appendix I on page 123 together with the corresponding numerical definitions used for the presented boxplots. For the visualization of the boxplots used within this thesis, the script of [Bik] has been used and extended.

H.1. General Settings

Within this section, common settings for the simulations of this thesis are provided. All described simulations have been performed with MATLAB® [Mat]. The plots have been exported with the help of [Sch].

H.1.1. Evaluation of CS Schemes

For the numerical evaluation of the different CS schemes, 10 000 realizations are considered for each level of sparsity τ. The obtained statistic of the squared error $\|\boldsymbol{\chi} - \widehat{\boldsymbol{\chi}}\|^2$ can be visualized by boxplots over the sparsity τ. Alternatively, the success rate can be plotted over τ, whereby a reconstruction is considered successful if $\|\boldsymbol{\chi} - \widehat{\boldsymbol{\chi}}\|^2 < 10^{-10}$.

Often, column-normalized random sensing matrices $\boldsymbol{\Theta}$ are considered as reference, their elements are drawn from a normal distributed source with zero mean in case of $\boldsymbol{\Theta} \in \mathbb{R}^{m \times n}$. For the complex valued case $\boldsymbol{\Theta} \in \mathbb{C}^{m \times n}$, the real and imaginary parts are drawn from the same source.

The τ non-zero components of the sparse vector $\boldsymbol{\chi}$ are uniformly distributed over the whole vector length. In case of $\boldsymbol{\chi} \in \mathbb{R}^n$, these components are drawn from a standard normal distributed source. For a complex-valued vector $\boldsymbol{\chi}$, the real and imaginary parts are drawn from a normal distributed random source with zero mean and standard deviation $\sigma = 1/\sqrt{2}$.

Some simulations consider a noisy environment (cf. Section 4.4.1 on page 31), where the real and imaginary parts of the complex-valued noise components are drawn from a normal distributed random source with zero mean and standard deviation $\sigma = \sigma_{\boldsymbol{\eta}}/\sqrt{2}$ with $\sigma_{\boldsymbol{\eta}} = 10^{-5}$.

H.1.2. Reconstruction Algorithms

For comparison, the BP is often considered. It is realized as LP according to (4.6) on page 25 for the simulations within this thesis. The corresponding LP is subsequently solved by a common and fast interior point algorithm by MOSEK® [Mos].

Additionally, the OMP is usually used for comparison. Therein, the Euclidean norm on the residual has been used as main stopping criterion, where the algorithm terminated as soon as this norm was less than a certain threshold ϵ_{OMP} or after the number of iterations equals the count of available columns, cf. Section 4.1.2 on page 26. In case of noiseless CS schemes, the threshold is chosen to $\epsilon_{\mathrm{OMP}} = 10^{-7}$. For the previously described noisy case with $\sigma_{\boldsymbol{\eta}} = 10^{-5}$, the threshold is equal to $\epsilon_{\mathrm{OMP}} = 10^{-4}$.

H.2. Settings for Sparsity Aware Simulations

Beyond the general settings of Appendix H.1, the following details have been used in Chapter 5. In the sparsity aware simplex algorithms, values with a magnitude of less than 10^{-10} are considered as zero in order to assure numerical stability with respect to the used floating point number representation.

For the simulations investigating the potential of sparsity aware simplex algorithms within LPs, CS based scenarios can be used, which have to be extended to LPs as described in (4.6) on page 25 and to be transformed into a simplex tableau as described in Section 5.1.2 on page 38. Consequently, a simulated LP with n unknowns, m equations in standard form originates from a CS scheme, where the sensing matrix is of size $m \times n/2$. For the provided comparisons of Figures 5.3 and 5.5, 10 000 realizations of CS-schemes with random sensing matrices of size 32×128, resulting in LPs with $n = 256$, $m = 32$, have been considered. Similarly, Table 5.1 has been created for 10 000 realizations of random sensing matrices each of size 16×32.

H.3. Settings for B(C)ASC Based Coherence Optimization

The following parameters, which have been used in Chapter 6, are given in addition to the general settings of Appendix H.1. For the proposed BCASC search approach, the exemplary parameters of Algorithm 6.1 on page 62 have been used for all runs. Column-normalized matrices, where the real and imaginary parts of the matrix elements are drawn from a standard normal distributed source, have been selected as random seeds. If not stated otherwise, the numerical integration of (6.24) has been performed with the help of the QAG adaptive integration from the GSL [gsl, GDT+09]. Thereby, a 61 point Gauss-Kronrod integration rule has been applied and a relative error of 10^{-4} with a maximal number of 1000 subintervals has been chosen as stopping criterion.

The numerically obtained coherence values of Tables 6.1 to 6.3 correspond to the best vector set out of ten independent runs. Similarly, the results of Figures 6.1 and 6.2 and Figures C.1 to C.3 are based on ten runs of each algorithm for every value of n. The running times given in Table 6.3 have been averaged over the corresponding ten runs. For the distribution plots in Figures 6.4a and 6.6, a binning with a width of 0.01 has been applied.

H.4. Settings for Simulations Based on CRS Codes

Beyond the general settings of Appendix H.1, the following parameters are used if not stated otherwise for simulations regarding CRS codes and CS schemes built thereon (cf. Chapter 7 on page 75): A CRS code of length $n = 50$ and dimension $k = 12$ is considered which results in a sensing matrix $\mathbf{\Theta}$ with $m = 38$ rows and $n = 50$ columns. The simulations consider a noisy environment as described in Appendix H.1.

For the given standard deviation of the noise $\sigma_{\boldsymbol{\eta}} = 10^{-5}$, the thresholds of Table H.1 have been empirically determined for the basic error location algorithms described in Section 7.2.1.

Table H.1.: Thresholds used for Figures 7.2 to 7.4

Threshold	Variable	Value
BM type	ϵ_{BM}	$3 \cdot 10^{-5}$
Peterson type	ϵ_{Pet}	$8 \cdot 10^{-5}$
EE type	ϵ_{EE}	$1 \cdot 10^{-5}$

Similarly, the thresholds of Table H.2 have been used for the overestimating versions of the discussed error location algorithms described in Section 7.2.2 for the same standard deviation.

Table H.2.: Thresholds for Overestimation used for Table 7.1, Figures 7.5 to 7.8, G.1 and G.2

Threshold	Variable	Value
BM type	ϵ_{BM}	10^{-11}
Peterson type	ϵ_{Pet}	$8 \cdot 10^{-5}$
EE type	ϵ_{EE}	10^{-11}

For the IEE and CAD algorithms, $\epsilon = 3 \cdot 10^{-5}$ has been used for the validity check (7.5) in case of the described noisy scenario with $\sigma_{\boldsymbol{\eta}} = 10^{-5}$.

I

Boxplots

Within this thesis, boxplots are used to visualize and to compare the distribution of given datasets. Such plots are advantageous, since they combine multiple properties of the actual distribution.

There are several different variants of boxplots [FHI89]. Within this work, mainly the model of Tukey [Tuk77] is used. The median m separates a dataset $\mathcal{X}$ such that one half is smaller or equal to the median and the other half is larger. In case of a dataset with an even number of elements, the median is defined as the mean of the two middle values. Hinges h_l and h_u (sometimes also denoted as first and third quartile) are again the median of all elements which are either $\leq m$ or $\geq m$, respectively. The main part of the boxplot is built by a rectangle, which resembles the values between both hinges. The median (also known as second quartile) itself corresponds to a horizontal band within this box. Attached to this box, there are the upper and lower whiskers, which are defined with the help of the interquartile range $i_\mathrm{qr} = h_\mathrm{u} - h_\mathrm{l}$. The upper and lower whiskers are defined as

$$w_\mathrm{u} = \max_{x \in \mathcal{X}} x \quad \text{with } x \leq h_\mathrm{u} + 3/2\ i_\mathrm{qr},$$
$$w_\mathrm{l} = \min_{x \in \mathcal{X}} x \quad \text{with } x \geq h_\mathrm{l} - 3/2\ i_\mathrm{qr}.$$

Data elements, which are not covered by the whiskers or the box itself, are denoted as outliers and are marked by corresponding dots within the plot. Additionally, the mean is given as circle within the boxplots. In Figure I.1, the concept is illustrated for an exemplary dataset $\mathcal{X}$.

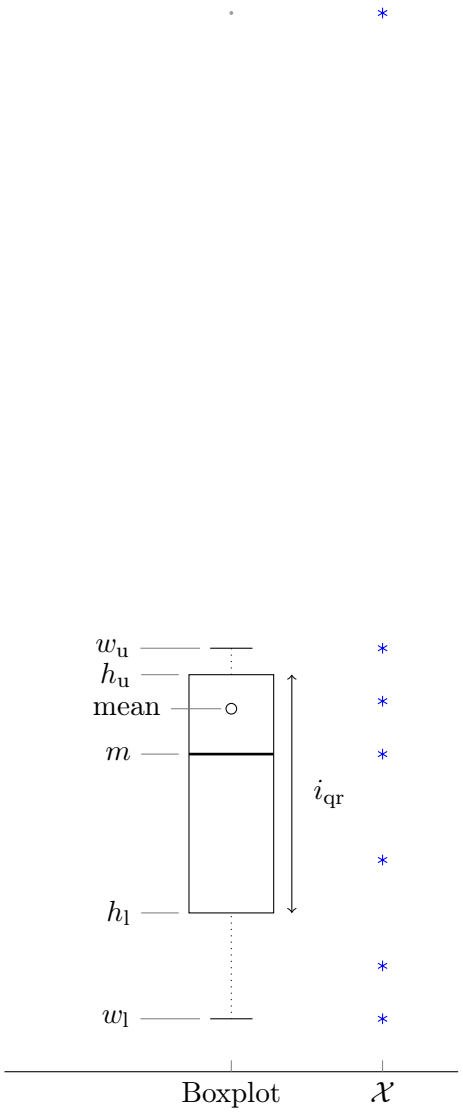


Figure I.1.: Boxplot of $\mathcal{X}$.

Bibliography

References

[ADAA08] F. Abdelkefi, P. Duhamel, F. Alberge, and J. Ayadi. "On the Use of Cascade Structure to Correct Impulsive Noise in Multicarrier Systems". *IEEE Transactions on Communications*, vol. 56, no. 11, pp. 1844–1858, Nov. 2008. Cited on pages 75, 81, 86, 87, and 93.

[AFMS10] V. Abolghasemi, S. Ferdowsi, B. Makkiabadi, and S. Saeid. "On Optimization of the Measurement Matrix for Compressive Sensing". In "Proceedings European Signal Processing Conference (EUSIPCO)", pp. 427–431. Aalborg, Denmark, Aug. 2010. Cited on page 70.

[AHK+11] T. Arens, F. Hettlich, C. Karpfinger, U. Kockelkorn, K. Lichtenegger, and H. Stachel. *Mathematik*. Spektrum Akademischer Verlag, Heidelberg, 2nd ed., Oct. 2011. Cited on page 36.

[AHSC09] L. Applebaum, S. D. Howard, S. Searle, and R. Calderbank. "Chirp Sensing Codes: Deterministic Compressed Sensing Measurements for Fast Recovery". *Applied and Computational Harmonic Analysis*, vol. 26, no. 2, pp. 283–290, Mar. 2009. Cited on pages 2, 31, and 75.

[AL69] H. Althaus and R. Leake. "Inverse of a Finite-Field Vandermonde Matrix". *IEEE Transactions on Information Theory*, vol. 15, no. 1, p. 173, Jan. 1969. Cited on page 7.

[AMM12] A. Amini, V. Montazerhodjat, and F. Marvasti. "Matrices With Small Coherence Using p-Ary Block Codes". *IEEE Transactions on Signal Processing*, vol. 60, no. 1, pp. 172–181, Jan. 2012. Cited on pages 2, 23, 31, and 75.

[ARU01] D. Agrawal, T. J. Richardson, and R. L. Urbanke. "Multiple-Antenna Signal Constellations for Fading Channels". *IEEE Transactions on Information Theory*, vol. 47, no. 6, pp. 2618–2626, Sep. 2001. Cited on pages 49 and 55.

[AT08] M. Akcakaya and V. Tarokh. "A Frame Construction and a Universal Distortion Bound for Sparse Representations". *IEEE Transactions on Signal Processing*, vol. 56, no. 6, pp. 2443–2450, Jun. 2008. Cited on page 75.

[BB13] M. Bossert and S. Bezzateev. "A Unified View on Known Algebraic Decoding Algorithms and New Decoding Concepts". *IEEE Transactions on Information Theory*, vol. 59, no. 11, pp. 7320–7336, Nov. 2013. Cited on pages 5 and 13.

[BD09] T. Blumensath and M. E. Davies. "Iterative Hard Thresholding for Compressed Sensing". *Applied and Computational Harmonic Analysis*, vol. 27, no. 3, pp. 265–274, Nov. 2009. Cited on page 26.

[BDDW08] R. G. Baraniuk, M. A. Davenport, R. A. DeVore, and M. B. Wakin. "A Simple Proof of the Restricted Isometry Property for Random Matrices". *Constructive Approximation*, vol. 28, no. 3, pp. 253–263, Dec. 2008. Cited on pages 29 and 32.

[BDMS13] A. S. Bandeira, E. Dobriban, D. G. Mixon, and W. F. Sawin. "Certifying the Restricted Isometry Property is Hard". *IEEE Transactions on Information Theory*, vol. 59, no. 6, pp. 3448–3450, Jun. 2013. Cited on page 28.

[Bea54] E. M. L. Beale. "An Alternative Method for Linear Programming". *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 50, no. 04, pp. 513–523, Oct. 1954. Cited on pages 35 and 47.

[Ber68] E. R. Berlekamp. *Algebraic Coding Theory*. McGraw–Hill, 1968. Cited on page 13.

[BF03] J. J. Benedetto and M. Fickus. "Finite Normalized Tight Frames". *Advances in Computational Mathematics*, vol. 18, no. 2-4, pp. 357–385, Feb. 2003. Cited on page 53.

[BGM96] G. A. Baker and P. R. Graves-Morris. *Padé Approximants*. Encyclopedia of Mathematics and Its Applications. Cambridge University Press, 1996. Cited on page 89.

[Bik] A. Bikfalvi. "Advanced Box Plot for Matlab Version Beta 1". `http://alex.bikfalvi.com/research/advanced_matlab_boxplot/` Cited on page 119.

[Bje51] A. Bjerhammar. "Rectangular Reciprocal Matrices, With Special Reference to Geodetic Calculations". *Bulletin géodésique*, vol. 20, no. 1, pp. 188–220, Jun. 1951. Cited on pages 18 and 81.

[Bla77] R. G. Bland. "New Finite Pivoting Rules for the Simplex Method". *Mathematics of Operations Research*, vol. 2, no. 2, pp. 103–107, May 1977. Cited on page 38.

[Bla03] R. E. Blahut. *Algebraic Codes for Data Transmission*. Cambridge University Press, New York, 2003. Cited on pages 5, 6, and 9.

[Blu12] Blu-ray Disc Association. "White Paper Blu-ray Disc™ Format – General", Dec. 2012. 3rd edition. Cited on page 8.

[Bos99] M. Bossert. *Channel Coding for Telecommunications*. Wiley, 1999. Cited on pages 1, 5, 6, 9, 15, and 77.

[Bos12] M. Bossert. *Einführung in die Nachrichtentechnik*. Oldenbourg Wissenschaftsverlag, 2012. Cited on page 31.

[BRC60] R. C. Bose and D. K. Ray-Chaudhuri. "On A Class of Error Correcting Binary Group Codes". *Information and Control*, vol. 3, no. 1, pp. 68–79, March 1960. Cited on page 5.

[BT97] D. Bertsimas and J. N. Tsitsiklis. *Introduction to Linear Optimization*. Athena Scientific, Belmont, Mass, Jan. 1997. Cited on pages 35 and 36.

[BT15] J. D. Blanchard and J. Tanner. "Performance Comparisons of Greedy Algorithms in Compressed Sensing". *Numerical Linear Algebra with Applications*, vol. 22, no. 2, pp. 254–282, Mar. 2015. Cited on page 23.

[Bus52] K. A. Bush. "Orthogonal Arrays of Index Unity". *Transactions of the American Mathematical Society*, vol. 23, no. 3, pp. 426–434, 1952. Cited on page 8.

[BV09] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 7th ed., 2009. Cited on page 19.

[Bé79] E. Bézout. *Théorie générale des équations algébrique*. Ph.D. thesis, University of Gent, Gent, Belgium, 1779. Cited on page 14.

[Can08] E. J. Candès. "The Restricted Isometry Property and Its Implications for Compressed Sensing". *Comptes Rendus Mathematique*, vol. 346, no. 9-10, pp. 589 – 592, 2008. Cited on page 28.

[Cap92] A. Capelli. "Sopra la compatibilita o incompatibilita di piu equazioni di primo grado fra piu incognite". *Rivista di matematica*, vol. 2, pp. 54–58, 1892. Cited on page 17.

[CBL89] S. Chen, S. A. Billings, and W. Luo. "Orthogonal Least Squares Methods and Their Application to Non-Linear System Identification". *International Journal of Control*, vol. 50, pp. 1873–1896, 1989. Cited on page 26.

[CDD09] A. Cohen, W. Dahmen, and R. A. DeVore. "Compressed Aensing and Best k-term Approximation". *Journal of the American Mathematical Society*, vol. 22, pp. 211–231, 2009. Cited on page 29.

[CDS98] S. S. Chen, D. L. Donoho, and M. A. Saunders. "Atomic Decomposition by Basis Pursuit". *SIAM Journal on Scientific Computing*, vol. 20, pp. 33–61, 1998. Cited on pages 23, 24, 25, and 35.

[CENR11] E. J. Candès, Y. C. Eldar, D. Needell, and P. Randall. "Compressed Sensing With Coherent and Redundant Dictionaries". *Applied and Computational Harmonic Analysis*, vol. 31, no. 1, pp. 59–73, Jul. 2011. Cited on pages 28 and 32.

[CHJ10a] R. Calderbank, S. Howard, and S. Jafarpour. "Construction of a Large Class of Deterministic Sensing Matrices That Satisfy a Statistical Isometry Property". *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 2, pp. 358–374, Apr. 2010. Cited on pages 2, 23, 28, 31, and 75.

[CHJ10b] R. Calderbank, S. Howard, and S. Jafarpour. "Sparse Reconstruction via the Reed–Muller Sieve". In "Proceedings IEEE International Symposium on Information Theory (ISIT)", pp. 1973–1977. Jun. 2010. Cited on pages 79 and 84.

[CHS96] J. H. Conway, R. H. Hardin, and N. J. A. Sloane. "Packing Lines, Planes, etc.: Packings in Grassmannian Spaces". *Experimental Mathematics*, vol. 5, no. 2, pp. 139–159, Jan. 1996. Cited on pages 49, 51, 53, and 54.

[CKP13] P. G. Casazza, G. Kutyniok, and F. Philipp. "Introduction to Finite Frame Theory". In P. G. Casazza and G. Kutyniok (eds.), "Finite Frames", Applied and Numerical Harmonic Analysis, pp. 1–53. Birkhäuser Boston, Jan. 2013. Cited on page 53.

[CP10] E. J. Candès and Y. Plan. "Matrix Completion With Noise". *Proceedings of the IEEE*, vol. 98, no. 6, pp. 925–936, Jun. 2010. Cited on page 23.

[CR07] E. J. Candès and J. Romberg. "Sparsity and Incoherence in Compressive Sampling". *Inverse Problems*, vol. 23, no. 3, pp. 969–985, 2007. Cited on pages 32, 70, and 74.

[CRTV05] E. J. Candès, M. Rudelson, T. Tao, and R. Vershynin. "Error Correction via Linear Programming". In "Proceedings IEEE Symposium on Foundations of Computer Science (FOCS)", pp. 668 –681. Oct. 2005. Cited on page 28.

[CS99] J. H. Conway and N. J. A. Sloane. *Sphere Packings, Lattices, and Groups*. Grundlehren der Mathematischen Wissenschaften. Springer, 3rd ed., 1999. Cited on page 55.

[CS03] D. Coppersmith and M. Sudan. "Reconstructing Curves in Three (and Higher) Dimensional Space From Noisy Data". In "Proceedings Symposium on Theory of Computing", pp. 136—142. ACM, New York, USA, 2003. Cited on page 77.

[CT05] E. J. Candès and T. Tao. "Decoding by Linear Programming". *IEEE Transactions on Information Theory*, vol. 51, no. 12, pp. 4203–4215, Dec. 2005. Cited on pages 28 and 29.

[CT06] E. J. Candès and T. Tao. "Near-Optimal Signal Recovery From Random Projections: Universal Encoding Strategies?" *IEEE Transactions on Information Theory*, vol. 52, no. 12, pp. 5406–5425, Dec. 2006. Cited on pages 1, 23, 28, and 29.

[CW08] E. J. Candès and M. B. Wakin. "An Introduction to Compressive Sampling". *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21–30, Mar. 2008. Cited on pages 32, 70, 74, and 75.

[CWX10] T. T. Cai, L. Wang, and G. Xu. "Shifting Inequality and Recovery of Sparse Signals". *IEEE Transactions on Signal Processing*, vol. 58, no. 3, pp. 1300 – 1308, 2010. Cited on page 28.

[CXZ09] T. T. Cai, G. Xu, and J. Zhang. "On Recovery of Sparse Signals via ℓ_1 Minimization". *IEEE Transactions on Information Theory*, vol. 55, no. 7, pp. 3388 –3397, 2009. Cited on page 28.

[CZ13] T. T. Cai and A. Zhang. "Sharp RIP Bound for Sparse Signal and Low-Rank Matrix Recovery". *Applied and Computational Harmonic Analysis*, vol. 35, no. 1, pp. 74–93, Jul. 2013. Cited on page 28.

[DCS09] J. M. Duarte-Carvajalino and G. Sapiro. "Learning to Sense Sparse Signals: Simultaneous Sensing Matrix and Sparsifying Dictionary Optimization". *IEEE Transactions on Image Processing*, vol. 18, no. 7, pp. 1395–1408, Jul. 2009. Cited on page 70.

[DDEK12] M. A. Davenport, M. F. Duarte, Y. C. Eldar, and G. Kutyniok. "Introduction to Compressed Sensing". In Y. C. Eldar and G. Kutyniok (eds.), "Compressed Sensing: Theory and Applications", Cambridge University Press, Cambridge, New York, May 2012. Cited on pages 23, 26, and 29.

[DDT+08] M. F. Duarte, M. A. Davenport, D. Takhar, J. N. Laska, T. Sun, K. F. Kelly, and R. G. Baraniuk. "Single-Pixel Imaging via Compressive Sampling". *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 83 –91, Mar. 2008. Cited on pages 1 and 23.

[DE03] D. L. Donoho and M. Elad. "Optimally Sparse Representation in General (Nonorthogonal) Dictionaries via ℓ_1 Minimization". *Proceedings of the National Academy of Sciences of the U.S.A.*, vol. 100, no. 5, pp. 2197–2202, 2003. Cited on pages 27 and 32.

[DeV07] R. A. DeVore. "Deterministic Constructions of Compressed Sensing Matrices". *Journal of Complexity*, vol. 23, no. 4-6, pp. 918 – 925, 2007. Cited on pages 2, 31, and 75.

[DF07] C. Ding and T. Feng. "A Generic Construction of Complex Codebooks Meeting the Welch Bound". *IEEE Transactions on Information Theory*, vol. 53, no. 11, pp. 4245–4250, Nov. 2007. Cited on page 54.

[DF08] C. Ding and T. Feng. "Codebooks From Almost Difference Sets". *Designs, Codes and Cryptography*, vol. 46, no. 1, pp. 113–126, Jan. 2008. Cited on page 54.

[DFG14] DFG - Deutsche Forschungsgemeinschaft. "Priority Programme "Compressed Sensing in Information Processing" (SPP 1798)", May 2014. Information für die Wissenschaft Nr. 28.
`http://www.dfg.de/foerderung/info_wissenschaft/2014/info_wissenschaft_14_28/` Cited on page 23.

[DGM86] I. Daubechies, A. Grossmann, and Y. Meyer. "Painless Nonorthogonal Expansions". *Journal of Mathematical Physics*, vol. 27, no. 5, pp. 1271–1283, May 1986. Cited on page 52.

[DGS75] P. Delsarte, J. M. Goethals, and J. J. Seidel. "Bounds for Systems of Lines, and Jacobi Polynomials". *Philips Research Reports*, vol. 30, 1975. Cited on page 51.

[DH01] D. L. Donoho and X. Huo. "Uncertainty Principles and Ideal Atomic Decomposition". *IEEE Transactions on Information Theory*, vol. 47, no. 7, pp. 2845 –2862, Nov. 2001. Cited on pages 27 and 32.

[DHJST08] I. S. Dhilon, R. W. Heath Jr., T. Strohmer, and J. A. Tropp. "Constructing Packings in Grassmannian Manifolds via Alternating Projection". *Experimental Mathematics*, vol. 17, no. 1, pp. 9 – 35, 2008. Cited on pages 49, 53, 55, 65, and 66.

[Din06] C. Ding. "Complex Codebooks From Combinatorial Designs". *IEEE Transactions on Information Theory*, vol. 52, no. 9, pp. 4229–4235, Sep. 2006. Cited on page 54.

[DL92] D. Donoho and B. Logan. "Signal Recovery and the Large Sieve". *SIAM Journal on Applied Mathematics*, vol. 52, no. 2, pp. 577–591, Apr. 1992. Cited on pages 23 and 35.

[DMM09] D. L. Donoho, A. Maleki, and A. Montanari. "Message-Passing Algorithms for Compressed Sensing". *Proceedings of the National Academy of Sciences of the U.S.A.*, vol. 106, no. 45, pp. 18914–18919, 2009. Cited on page 26.

[DMM10a] D. L. Donoho, A. Maleki, and A. Montanari. "Message Passing Algorithms for Compressed Sensing: I. Motivation and Construction". In "2010 IEEE Information Theory Workshop (ITW)", Cairo, Egypt, Jan. 2010. Cited on page 26.

[DMM10b] D. L. Donoho, A. Maleki, and A. Montanari. "Message Passing Algorithms for Compressed Sensing: II. Analysis and Validation". In "2010 IEEE Information Theory Workshop (ITW)", Cairo, Egypt, Jan. 2010. Cited on page 26.

[DMZ94] G. M. Davis, S. G. Mallat, and Z. Zhang. "Adaptive Time-Frequency Decompositions". *Optical Engineering*, vol. 33, no. 7, pp. 2183–2191, 1994. Cited on page 26.

[Don06a] D. L. Donoho. "Compressed Sensing". *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289 –1306, Apr. 2006. Cited on pages 1, 23, and 29.

[Don06b] D. L. Donoho. "For Most Large Underdetermined Systems of Linear Equations the Minimal ℓ_1-Norm Solution is also the Sparsest Solution". *Communications on Pure and Applied Mathematics*, vol. 59, no. 6, pp. 797–829, Jun. 2006. Cited on page 28.

[DS52] R. J. Duffin and A. C. Schaeffer. "A Class of Nonharmonic Fourier Series". *Transactions of the American Mathematical Society*, vol. 72, no. 2, pp. 341–366, Mar. 1952. Cited on page 52.

[DT97] G. B. Dantzig and M. N. Thapa. *Linear Programming 1: Introduction*. Springer Series in Operations Research. Springer, 1997. Cited on pages 19, 35, 36, 39, and 47.

[DTDS12] D. L. Donoho, Y. Tsaig, I. Drori, and J.-L. Starck. "Sparse Solution of Underdetermined Systems of Linear Equations by Stagewise Orthogonal Matching Pursuit". *IEEE Transactions on Information Theory*, vol. 58, no. 2, pp. 1094–1121, Feb. 2012. Cited on page 26.

[DY07] C. Ding and J. Yin. "Signal Sets From Functions With Optimum Nonlinearity". *IEEE Transactions on Communications*, vol. 55, no. 5, pp. 936–940, May 2007. Cited on pages 52, 54, and 65.

[EB02] M. Elad and A. M. Bruckstein. "A Generalized Uncertainty Principle and Sparse Representation in Pairs of Bases". *IEEE Transactions on Information Theory*, vol. 48, no. 9, pp. 2558 – 2567, Sep. 2002. Cited on page 27.

[Ela07] M. Elad. "Optimized Projections for Compressed Sensing". *IEEE Transactions on Signal Processing*, vol. 55, no. 12, pp. 5695 –5702, Dec. 2007. Cited on pages 70, 71, and 72.

[Ela10] M. Elad. *Sparse and Redundant Representations, From Theory to Applications in Signal and Image Processing*. Springer, Aug. 2010. Cited on pages 24 and 32.

[End10] J. H. G. Ender. "On Compressive Sensing Applied to Radar". *Signal Processing*, vol. 90, no. 5, pp. 1402 – 1414, 2010. Cited on pages 1 and 23.

[EZ01] T. Ericson and V. Zinoviev. *Codes on Euclidean Spheres*. North-Holland Mathematical Library. Elsevier, 2001. Cited on page 55.

[FG92] J. J. Forrest and D. Goldfarb. "Steepest-Edge Simplex Algorithms for Linear Programming". *Mathematical Programming*, vol. 57, no. 1-3, pp. 341–374, May 1992. Cited on page 35.

[FHI89] M. Frigge, D. C. Hoaglin, and B. Iglewicz. "Some Implementations of the Boxplot". *The American Statistician*, vol. 43, no. 1, pp. 50–54, Feb. 1989. Cited on page 123.

[Fit95] P. Fitzpatrick. "On the Key Equation". *IEEE Transactions on Information Theory*, vol. 41, no. 5, pp. 1290–1302, Sep. 1995. Cited on page 90.

[FJ65] G. D. Forney Jr. "On Decoding BCH Codes". *IEEE Transactions on Information Theory*, vol. 11, no. 4, pp. 549–557, Oct. 1965. Cited on page 16.

[FJ66] G. D. Forney Jr. "Generalized Minimum Distance Decoding". *IEEE Transactions on Information Theory*, vol. 12, no. 2, pp. 125–131, Apr. 1966. Cited on page 12.

[FL09] S. Foucart and M.-J. Lai. "Sparsest Solutions of Underdetermined Linear Systems via ℓ_q-Minimization for $0 < q \leq 1$". *Applied and Computational Harmonic Analysis*, vol. 26, no. 3, pp. 395 – 407, 2009. Cited on page 28.

[FMT12] M. Fickus, D. G. Mixon, and J. C. Tremain. "Steiner Equiangular Tight Frames". *Linear Algebra and its Applications*, vol. 436, no. 5, pp. 1014–1027, Mar. 2012. Cited on pages 49 and 54.

[FN03] A. Feuer and A. Nemirovski. "On Sparse Representation in Pairs of Bases". *IEEE Transactions on Information Theory*, vol. 49, no. 6, pp. 1579 – 1581, Jun. 2003. Cited on page 27.

[Fou10] S. Foucart. "A Note on Guaranteed Sparse Recovery via ℓ_1-Minimization". *Applied and Computational Harmonic Analysis*, vol. 29, no. 1, pp. 97 – 103, 2010. Cited on page 28.

[FR13] S. Foucart and H. Rauhut. *A Mathematical Introduction to Compressive Sensing*. Applied and Numerical Harmonic Analysis. Birkhäuser, 2013. Cited on pages 20, 21, 23, 24, 25, 26, 28, 30, 31, 35, 75, and 78.

[Fro81] G. Frobenius. "Ueber Relationen zwischen den Näherungsbrüchen von Potenzreihen". *Journal für die reine und angewandte Mathematik*, vol. 90, pp. 1–17, 1881. Cited on page 89.

[Fuc03] J.-J. Fuchs. "More on Sparse Representations in Arbitrary Bases". In "13th IFAC-IFORS Symposium on Identification and System Parameter Estimation", vol. 2, pp. 1357–1362. Rotterdam, 2003. Cited on page 27.

[Fuc04] J.-J. Fuchs. "On Sparse Representations in Arbitrary Redundant Bases". *IEEE Transactions on Information Theory*, vol. 50, no. 6, pp. 1341 – 1344, Jun. 2004. Cited on page 27.

[Gal30] E. Galois. "Sur la théorie des nombres". *Bulletin des Sciences Mathématiques*, vol. 13, pp. 428–435, 1830. Translation into English: [Neu11]. Cited on page 6.

[Gal62] R. G. Gallager. "Low-Density Parity-Check Codes". *IRE Transactions on Information Theory*, vol. 8, no. 1, pp. 21–28, Jan. 1962. Cited on page 5.

[GD09] R. H. Gohary and T. N. Davidson. "Noncoherent MIMO Communication: Grassmannian Constellations and Efficient Detection". *IEEE Transactions on Information Theory*, vol. 55, no. 3, pp. 1176–1205, Mar. 2009. Cited on pages 49, 55, and 68.

[GDT+09] M. Galassi, J. Davies, J. Theiler, B. Gough, G. Jungman, P. Alken, M. Booth, and F. Rossi. *GNU Scientific Library Reference Manual*. Network Theory Ltd., Bristol, 3rd revised ed., Jan. 2009. Cited on page 120.

[GN03] R. Gribonval and M. Nielsen. "Sparse Representations in Unions of Bases". *IEEE Transactions on Information Theory*, vol. 49, no. 12, pp. 3320 – 3325, Dec. 2003. Cited on page 27.

[GS99] V. Guruswami and M. Sudan. "Improved Decoding of Reed–Solomon and Algebraic-Geometry Codes". *IEEE Transactions on Information Theory*, vol. 45, no. 6, pp. 1757–1767, Sep. 1999. Cited on pages 5 and 8.

[gsl] "GSL - GNU Scientific Library Version 1.16". `http://www.gnu.org/software/gsl/` Cited on page 120.

[GVL96] G. H. Golub and C. F. Van Loan. *Matrix Computations*. The John Hopkins University Press, Baltimore, 3rd ed., 1996. Cited on page 18.

[GZ61] D. Gorenstein and N. Zierler. "A Class of Error-Correcting Codes in p^m Symbols". *Journal of the Society of Industrial and Applied Mathematics*, vol. 9, no. 2, pp. 207–214, Jun. 1961. Cited on page 16.

[Ham50] R. W. Hamming. "Error Detecting and Error Correcting Codes". *Bell System Technical Journal*, vol. 29, no. 2, pp. 147–160, 1950. Cited on pages 1 and 5.

[Har97] R. Hartshorne. *Algebraic Geometry*. Graduate Texts in Mathematics. Springer, 8th ed., 1997. Cited on page 20.

[HCS08] S. D. Howard, A. R. Calderbank, and S. J. Searle. "A Fast Reconstruction Algorithm for Deterministic Compressive Sensing Using Second Order Reed–Muller Codes". In "Conference in Information Sciences and Systems", pp. 11–15. Princeton, 2008. Cited on pages 2, 31, and 75.

[Hen89] W. Henkel. *Zur Decodierung algebraischer Blockcodes über komplexen Alphabeten*. Fortschritt-Berichte VDI. VDI-Verlag, 1989. Cited on pages 75, 79, and 80.

[Hen00] W. Henkel. "Analog Codes for Peak-to-Average Ratio Reduction". In "Proceedings International ITG Conference on Source and Channel Coding (SCC)", Munich, Jan. 2000. Cited on page 75.

[Hen05] O. Henkel. "Sphere-Packing Bounds in the Grassmann and Stiefel Manifolds". *IEEE Transactions on Information Theory*, vol. 51, no. 10, pp. 3445–3456, Oct. 2005. Cited on page 51.

[HH05] W. Henkel and F. N. Hu. "OFDM and Analog RS/BCH Codes". In "Proceedings OFDM-Workshop", Hamburg, Aug. 2005. Cited on page 75.

[HHH12] M. Huemer, C. Hofbauer, and J. B. Huber. "Non-Systematic Complex Number RS Coded OFDM by Unique Word Prefix". *IEEE Transactions on Signal Processing*, vol. 60, no. 1, pp. 285–299, Jan. 2012. Cited on page 75.

[HHL11] J. P. Haldar, D. Hernando, and Z.-P. Liang. "Compressed-Sensing MRI With Random Encoding". *IEEE Transactions on Medical Imaging*, vol. 30, no. 4, pp. 893–903, 2011. Cited on pages 1 and 23.

[HHZ11] F. N. Hu, W. Henkel, and M. J. Zhao. "Analog Codes for Gross Error Correction in Signal Transmission". *Advanced Materials Research*, vol. 341–342, pp. 514–518, Sep. 2011. Cited on pages 75, 80, and 81.

[Hig96] N. J. Higham. *Accuracy and Stability of Numerical Algorithms*. Society for Industrial and Applied Mathematics, 1996. Cited on page 79.

[HJSP03] R. W. Heath Jr., T. Strohmer, and A. J. Paulraj. "Grassmannian Signatures for CDMA Systems". In "Global Telecommunications Conference (GLOBECOM)", vol. 3, pp. 1553–1557. 2003. Cited on page 54.

[HJSP06] R. W. Heath Jr., T. Strohmer, and A. J. Paulraj. "On Quasi-Orthogonal Signatures for CDMA Systems". *IEEE Transactions on Information Theory*, vol. 52, no. 3, pp. 1217–1226, 2006. Cited on pages 49 and 68.

[HMR+00] B. M. Hochwald, T. L. Marzetta, T. J. Richardson, W. Sweldens, and R. Urbanke. "Systematic Design of Unitary Space-Time Constellations". *IEEE Transactions on Information Theory*, vol. 46, no. 6, pp. 1962–1973, Sep. 2000. Cited on pages 49, 54, 55, and 68.

[HN07a] J. Haupt and R. Nowak. "Compressive Sampling for Signal Detection". In "Proceedings IEEE International Conference on Accoustics, Spech, and Signal Processing (ICASSP)", vol. 3, pp. III 1509 – III 1512. Honolulu , Hawaii, Apr. 2007. Cited on page 23.

[HN07b] J. Haupt and R. Nowak. "A Generalized Restricted Isometry Property". Tech. Rep. ECE-07-1, University of Wisconsin - Madison, 2007. Cited on page 28.

[Hoc59] A. Hocquenghem. "Codes Correcteurs d'Erreurs". *Chiffres*, vol. 2, pp. 147–156, Sep. 1959. Cited on page 5.

[HS09] M. A. Herman and T. Strohmer. "High-Resolution Radar via Compressed Sensing". *IEEE Transactions on Signal Processing*, vol. 57, no. 6, pp. 2275–2284, 2009. Cited on pages 1 and 23.

[HW14] H. Hu and J. Wu. "New Constructions of Codebooks Nearly Meeting the Welch Bound With Equality". *IEEE Transactions on Information Theory*, vol. 60, no. 2, pp. 1348–1355, Feb. 2014. Cited on page 54.

[IEE08] IEEE. *Standard for Floating-Point Arithmetic*, Aug. 2008. IEEE Std 754-2008. Cited on page 79.

[Jac46] C. G. J. Jacobi. "Über die Darstellung einer Reihe gegebner Werthe durch eine gebrochne rationale Function". *Journal für die reine und angewandte Mathematik*, vol. 30, pp. 127–156, 1846. Cited on page 89.

[JH04] J. Justesen and T. Hoholdt. *A Course In Error-Correcting Codes*. European Mathematical Society, Zürich, Switzerland, Feb. 2004. Cited on page 5.

[JMF14] J. Jasper, D. G. Mixon, and M. Fickus. "Kirkman Equiangular Tight Frames and Codes". *IEEE Transactions on Information Theory*, vol. 60, no. 1, pp. 170–181, Jan. 2014. Cited on page 54.

[Jos58] D. D. Joshi. "A Note on Upper Bounds for Minimum Distance Codes". *Information and Control*, vol. 1, no. 3, pp. 289–295, Sep. 1958. Cited on page 8.

[Kar84] N. Karmarkar. "A New Polynomial-time Algorithm for Linear Programming". In "Proceedings ACM Symposium on Theory of Computing (STOC)", p. 302–311. ACM, New York, 1984. Cited on pages 19 and 35.

[KC07a] J. Kovacevic and A. Chebira. "Life Beyond Bases: The Advent of Frames (Part I)". *IEEE Signal Processing Magazine*, vol. 24, no. 4, pp. 86–104, Jul. 2007. Cited on page 53.

[KC07b] J. Kovacevic and A. Chebira. "Life Beyond Bases: The Advent of Frames (Part II)". *IEEE Signal Processing Magazine*, vol. 24, no. 5, pp. 115–125, Sep. 2007. Cited on page 53.

[KCB07] I. Kammoun, A. M. Cipriano, and J. C. Belfiore. "Non-Coherent Codes over the Grassmannian". *IEEE Transactions on Wireless Communications*, vol. 6, no. 10, pp. 3657–3667, Oct. 2007. Cited on pages 49 and 55.

[KL97] V. Y. Krachkovsky and Y. X. Lee. "Decoding for Iterative Reed–Solomon Coding Schemes". *IEEE Transactions on Magnetics*, vol. 33, no. 5, pp. 2740–2742, Sep. 1997. Cited on page 10.

[KM72] V. Klee and G. J. Minty. "How Good is the Simplex Algorithm?" In O. Shisha (ed.), "Inequalities", vol. III, pp. 159–175. Academic Press, New York, 1972. (Proceedings of the Third Symposium on Inequalities held at the University of California, Los Angeles, Sep. 1–9). Cited on pages 19 and 35.

[Kom53] Y. Komamiya. "Application of Logical Mathematics to Information Theory". In "Proceedings 3rd National Congress for Applied Mathematics", p. 437. 1953. Cited on page 8.

[Kro03] L. Kronecker. *Vorlesungen über die Theorie der Determinanten*. B.G. Teubner, Leipzig, Germany, 1903. Cited on page 17.

[Kum85] R. Kumaresan. "Rank Reduction Techniques and Burst Error-Correction Decoding In Real/Complex Fields". In "Proceedings Asilomar Conference on Circuits, Systems and Computers", pp. 457–461. Nov. 1985. Cited on pages 75 and 79.

[KV03] R. Koetter and A. Vardy. "Algebraic Soft-Decision Decoding of Reed–Solomon Codes". *IEEE Transactions on Information Theory*, vol. 49, no. 11, pp. 2809–2825, Nov. 2003. Cited on page 12.

[Lay12] D. C. Lay. *Linear Algebra and Its Applications*. Addison–Wesley, 4th ed., 2012. Cited on page 17.

[Laz80] D. E. Lazic. "Class of Block Codes for the Gaussian Channel". *Electronics Letters*, vol. 16, no. 5, pp. 185–186, Feb. 1980. Cited on page 57.

[Laz13] D. E. Lazich. "Principles of Compressed Sensing", Jan. 2013. Tutorial for the 9th International ITG Conference on Systems, Communications and Coding (SCC), Munich, Germany. Cited on page 20.

[Laz14] D. E. Lazich. "Lecture on Compressed Sensing", Summer term 2014. Ulm University, Germany. Cited on page 20.

[LBK86] D. E. Lazic, T. Bece, and P. J. Krstajic. "On the Construction of the Best Spherical Code by Computing the Fixed Point". In "Proceedings IEEE International Symposium on Information Theory (ISIT)", p. 74. Ann Arbor, Michigan, USA, 1986. Cited on pages 57 and 62.

[Lee57] J. Leech. "Equilibrium of Sets of Particles on a Sphere". *The Mathematical Gazette*, vol. 41, no. 336, pp. 81 – 90, 1957. Cited on page 57.

[Lem54] C. E. Lemke. "The Dual Method of Solving the Linear Programming Problem". *Naval Research Logistics Quarterly*, vol. 1, no. 1, pp. 36–47, Mar. 1954. Cited on pages 35 and 47.

[Lev83] V. I. Levenshtein. "Bounds for Packings of Metric Spaces and Some of Their Applications". *Problemy Kibernetiсki*, vol. 40, pp. 43—110, 1983. (in Russian). Cited on page 52.

[LHJS03] D. J. Love, R. W. Heath Jr., and T. Strohmer. "Grassmannian Beamforming for Multiple-Input Multiple-Output Wireless Systems". *IEEE Transactions on Information Theory*, vol. 49, no. 10, pp. 2735–2747, 2003. Cited on pages 49 and 56.

[LK78] V. I. Levenshtein and G. A. Kabatiansky. "On Bounds for Packings on a Sphere and in Space". *Problemy Peredachi Informatsii*, vol. 14, no. 1, pp. 3—25, 1978. (in Russian). Cited on page 52.

[LN97] R. Lidl and H. Niederreiter. *Finite Fields*. No. 20 in Encyclopedia of Mathematics and Its Applications. Cambridge University Press, Cambridge, New York, 2nd ed., 1997. Cited on page 6.

[Log65] B. Logan. *Properties of High-Pass Signals*. Ph.D. thesis, Columbia University, New York, 1965. Cited on pages 23 and 35.

[Lov] D. J. Love. "Grassmannian Subspace Packing". 2014-03-26. `https://engineering.purdue.edu/~djlove/grass.html` Cited on page 65.

[LSZ88] D. E. Lazic, V. Senk, and R. Zamurovic. "An Efficient Numerical Procedure for Generating Best Spherical Arrangements of Points". In "Proceedings of the International AMSE Conference "Modeling and Simulation"", vol. 1C, pp. 267 – 278. Istanbul, Turkey, 1988. Cited on pages 57, 58, 59, 62, and 73.

[Mal08] S. G. Mallat. *A Wavelet Tour of Signal Processing, The Sparse Way*. Academic Press, 3rd ed., 2008. Cited on page 32.

[Mar81] T. G. Marshall, Jr. "Real Number Rransform and Convolutional Codes". In "Proceedings Midwest Symposium on Circuits and Systems", pp. 650–653. Albuquerque, New Mexico, Jun. 1981. Cited on pages 75 and 77.

[Mar84] T. G. Marshall, Jr. "Coding of Real-Number Sequences for Error Correction: A Digital Signal Processing Problem". *IEEE Journal on Selected Areas in Communications*, vol. 2, no. 2, pp. 381–392, Mar. 1984. Cited on pages 75, 77, and 79.

[Mas69] J. L. Massey. "Shift-Register Synthesis and BCH Decoding". *IEEE Transactions on Information Theory*, vol. 15, no. 1, pp. 122–127, Jan. 1969. Cited on page 13.

[Mas84] J. L. Massey. "Information Theory: The Copernican System of Communications". *IEEE Communication Magazine*, vol. 22, no. 12, pp. 26–28, Dec. 1984. Cited on page 5.

[Mat] MathWorks. "MATLAB® Version R2014b (8.4.0.150421)". Natick, Massachusetts. `http://www.mathworks.com/` Cited on page 119.

[MD14] A. Medra and T. N. Davidson. "Flexible Codebook Design for Limited Feedback Systems via Sequential Smooth Optimization on the Grassmannian Manifold". *IEEE Transactions on Signal Processing*, vol. 62, no. 5, pp. 1305–1318, Mar. 2014. Cited on pages 49, 55, 63, 65, 66, and 107.

[Med] A. Medra. "Grassmannian Design Package". 2014-03-26. `http://www.mathworks.com/matlabcentral/fileexchange/41652-grassmannian-design-package` Cited on page 63.

[Mey00] C. D. Meyer. *Matrix Analysis and Applied Linear Algebra*. SIAM, Jun. 2000. Cited on page 17.

[MG15] M. Mayer and N. Goertz. "Improving Approximate Message Passing Recovery of Sparse Binary Vectors by Post Processing". In "Proceedings International ITG Conference on Systems Communications and Coding (SCC)", Hamburg, Germany, Feb. 2015. Cited on page 26.

[MHET99] F. Marvasti, M. Hasan, M. Echhart, and S. Talebi. "Efficient Algorithms for Burst Error Recovery Using FFT and Other Transform Kernels". *IEEE Transactions on Signal Processing*, vol. 47, no. 4, pp. 1065–1075, Apr. 1999. Cited on pages 75 and 79.

[Mil10] G. A. Miller. "On the Solution of a System of Linear Equations". *The American Mathematical Monthly*, vol. 17, no. 6/7, pp. 137–139, Jun. 1910. Cited on page 17.

[MM93] J. L. Massey and T. Mittelholzer. "Welch's Bound and Sequence Sets for Code-Division Multiple-Access Systems". In R. Capocelli, A. D. Santis, and U. Vaccaro (eds.), "Sequences II", pp. 63–78. Springer New York, Jan. 1993. Cited on pages 51 and 54.

[MN95] D. J. MacKay and R. M. Neal. "Good Codes Based on Very Sparse Matrices". In "Cryptography and Coding", pp. 100–111. Springer, 1995. Cited on page 5.

[Moo20] E. H. Moore. "On the Reciprocal of the General Algebraic Matrix". *Bulletin of the American Mathematical Society*, vol. 26, pp. 394–395, 1920. Summary of the fourteenth western meeting of the American Mathematical Society. Cited on pages 18 and 81.

[Moo05] T. K. Moon. *Error Correction Coding: Mathematical Methods and Algorithms*. John Wiley & Sons, Hoboken, NJ, Jul. 2005. Cited on pages 5, 15, and 16.

[Mos] Mosek ApS. "The MOSEK® Optimization Software Version 7.0.0.103". Copenhagen, Denmark.
`http://www.mosek.com/` Cited on page 120.

[MP13] G. L. Mullen and D. Panario. *Handbook of Finite Fields*. Discrete Mathematics and Its Applications. CRC Press Taylor & Francis Group, Boca Raton, 2013. Cited on page 6.

[MS85] Y. Maekawa and K. Sakaniwa. "An Extension of DFT Code and the Evaluation of Its Performance". In "Proceedings IEEE International Symposium on Information Theory (ISIT)", pp. 24–28. Jun. 1985. Cited on page 75.

[MS88] F. J. MacWilliams and N. J. A. Sloane. *The Theory of Error-Correcting Codes*. North Holland Publishing Co., Jun. 1988. Cited on pages 5, 14, and 15.

[MS03] T. Mulders and A. Storjohann. "On Lattice Reduction for Polynomial Matrices". *Journal of Symbolic Computation*, vol. 35, no. 4, pp. 377–401, Apr. 2003. Cited on page 15.

[MSEA03] K. K. Mukkavilli, A. Sabharwal, E. Erkip, and B. Aazhang. "On Beamforming With Finite Rate Feedback in Multiple-Antenna Systems". *IEEE Transactions on Information Theory*, vol. 49, no. 10, pp. 2562–2579, 2003. Cited on page 52.

[Mul54] D. E. Muller. "Application of Boolean Algebra to Switching Circuit Design and to Error Detection". *IRE Transactions on Electronic Computers*, vol. EC-3, no. 3, pp. 6–12, Sep. 1954. Cited on page 5.

[MV02] M. L. McCloud and M. K. Varanasi. "Modulation and Coding for Noncoherent Communications". *Journal of VLSI signal processing systems for signal, image and video technology*, vol. 30, no. 1-3, pp. 35–54, 2002. Cited on page 49.

[MZ93] S. G. Mallat and Z. Zhang. "Matching Pursuits With Time-Frequency Dictionaries". *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3397 –3415, Dec. 1993. Cited on pages 23 and 25.

[Nat95] B. K. Natarajan. "Sparse Approximate Solutions to Linear Systems". *SIAM Journal on Computing*, vol. 24, no. 2, pp. 227–234, Apr. 1995. Cited on pages 23, 24, and 35.

[Neu11] P. M. Neumann. *The Mathematical Writings of Evariste Galois*. European Mathematical Society, Zürich, 2011. Cited on page 130.

[Nie13] J. S. R. Nielsen. *List Decoding of Algebraic Codes*. Ph.D. thesis, Technical University of Denmark, Lyngby, Denmark, 2013. Cited on page 15.

[NS95] S. Nash and A. Sofer. *Linear and Nonlinear Programming*. McGraw-Hill, New York, Dec. 1995. Cited on page 35.

[NT09] D. Needell and J. A. Tropp. "CoSaMP: Iterative Signal Recovery From Incomplete and Inaccurate Samples". *Applied and Computational Harmonic Analysis*, vol. 26, no. 3, pp. 301–321, May 2009. Cited on pages 26 and 28.

[NV08] D. Needell and R. Vershynin. "Uniform Uncertainty Principle and Signal Recovery via Regularized Orthogonal Matching Pursuit". *Foundations of Computational Mathematics*, vol. 9, no. 3, pp. 317–334, Jun. 2008. Cited on page 26.

[Pad92] H. Padé. *Sur la représentation approchée d'une fonction par des fractions rationnelles*. Ph.D. thesis, Gauthier-Villars et Fils, Paris, 1892. Cited on page 89.

[Pen55] R. Penrose. "A Generalized Inverse for Matrices". *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 51, no. 03, pp. 406–413, Jul. 1955. Cited on pages 18 and 81.

[Pet60] W. W. Peterson. "Encoding and Error-Correction Procedures for the Bose-Chaudhuri Codes". *IRE Transactions on Information Theory*, vol. 6, no. 4, pp. 459–470, Sep. 1960. Cited on page 13.

[PH08] F. Parvaresh and B. Hassibi. "Explicit Measurements With Almost Optimal Thresholds for Compressed Sensing". In "Proceedings IEEE International Conference on Accoustics, Spech, and Signal Processing (ICASSP)", pp. 3853–3856. Mar. 2008. Cited on pages 2, 31, 75, 77, and 79.

[Pol71] J. M. Pollard. "The Fast Fourier Transform in a Finite Field". *Mathematics of Computation*, vol. 25, no. 114, pp. 365–374, 1971. Cited on page 6.

[Pra94] V. V. Prasolov. *Problems and Theorems in Linear Algebra*. American Mathematical Society, Providence, RI, 1994. Cited on page 17.

[PRK93] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad. "Orthogonal Matching Pursuit: Recursive Function Approximation With Applications to Wavelet Decomposition". In "Proceedings Asilomar Conference on Signals, Systems and Computers", pp. 40–44 vol.1. Pacific Grove , California, Nov. 1993. Cited on pages 25 and 26.

[Pro95] R. Prony. "Essai expérimental et analytique sur les lois de la dilatabilité des fluides élastiques et sur celles de la force expansive de la vapeur de l'eau et de la vapeur de l'alcool à différentes températures". *Journal on de l'Ecole Polytechnique*, vol. 1, pp. 24–76, 1795. Cited on page 23.

[PS08] J. G. Proakis and M. Salehi. *Digital Communications*. McGraw-Hill, Boston, 5th ed., 2008. Cited on page 54.

[PTB11] R.-A. Pitaval, O. Tirkkonen, and S. D. Blostein. "Low Complexity MIMO Precoding Codebooks From Orthoplex Packings". In "Proceedings IEEE International Conference on Communication (ICC)", pp. 1–5. Jun. 2011. Cited on page 51.

[Ran56] R. A. Rankin. "On the Minimal Points of Positive Definite Quadratic Forms". *Mathematika*, vol. 3, no. 01, pp. 15–24, 1956. Cited on page 51.

[Ran11] S. Rangan. "Generalized Approximate Message Passing for Estimation With Random Linear Mixing". In "Proceedings IEEE International Symposium on Information Theory (ISIT)", pp. 2168–2172. St. Petersburg , Russia, Jul. 2011. Cited on pages 23 and 26.

[Red00] G. R. Redinbo. "Decoding Real Block Codes: Activity Detection Wiener Estimation". *IEEE Transactions on Information Theory*, vol. 46, no. 2, pp. 609–623, Mar. 2000. Cited on pages 75, 79, 80, 81, 86, 87, and 93.

[Red14] G. R. Redinbo. "Correcting DFT Codes With a Modified Berlekamp-Massey Algorithm and Kalman Recursive Syndrome Extension". *IEEE Transactions on Computers*, vol. 63, no. 1, pp. 196–203, Jan. 2014. Cited on pages 75 and 80.

[Ree54] I. Reed. "A Class of Multiple-Error-Correcting Codes and the Decoding Scheme". *IRE Transactions on Information Theory*, vol. 4, no. 4, pp. 38–49, Sep. 1954. Cited on page 5.

[RG04] G. Rath and C. Guillemot. "Subspace Algorithms for Error Localization With Quantized DFT Codes". *IEEE Transactions on Communications*, vol. 52, no. 12, pp. 2115–2124, Dec. 2004. Cited on pages 75, 80, and 81.

[Riz14] S. M. M. Rizkalla. *Reed–Solomon Codes over the Complex Field - Connection to Compressed Sensing.* Master's thesis, Ulm University, Apr. 2014. Supervised by Mostafa H. Mohamed and Henning Zörlein. Cited on pages 80, 86, and 115.

[Rom09] J. Romberg. "Compressive Sensing by Random Convolution". *SIAM Journal on Imaging Sciences*, vol. 2, no. 4, pp. 1098–1128, Jan. 2009. Cited on page 29.

[Rot82] T. Rothman. "Genius and Biographers: The Fictionalization of Évariste Galois". *The American Mathematical Monthly*, vol. 89, pp. 84–106, 1982. Cited on page 6.

[Rot06] R. M. Roth. *Introduction to Coding Theory.* Cambridge UP, 2006. Cited on pages 5, 6, and 16.

[RS60] I. Reed and G. Solomon. "Polynomial Codes over Certain Finite Fields". *Journal of the Society of Industrial and Applied Mathematics*, vol. 8, no. 2, pp. 300–304, Jun. 1960. Cited on pages 5 and 8.

[Rus13] C. Rusu. "Design of Incoherent Frames via Convex Optimization". *IEEE Signal Processing Letters*, vol. 20, no. 7, pp. 673–676, Jul. 2013. Cited on page 55.

[Sar99] D. V. Sarwate. "Meeting the Welch Bound With Equality". In C. Ding, T. Helleseth, and H. Niederreiter (eds.), "Sequences and Their Applications", Discrete Mathematics and Theoretical Computer Science, pp. 79–102. Springer London, Jan. 1999. Cited on pages 51, 54, and 55.

[Sch] N. Schlömer. "matlab2tikz Version 0.5.0".
`https://github.com/matlab2tikz/matlab2tikz` Cited on page 119.

[Sch60] J. Schwinger. "Unitary Operator Bases". *Proceedings of the National Academy of Sciences of the U.S.A.*, vol. 46, no. 4, pp. 570–579, Apr. 1960. Cited on page 54.

[Sha48] C. E. Shannon. "A Mathematical Theory of Communication". *Bell System Technical Journal*, vol. 27, pp. 379–423, 623–656, 1948. Cited on pages 1, 5, and 31.

[SHJ03] T. Strohmer and R. W. Heath Jr. "Grassmannian Frames With Applications to Coding and Communication". *Applied and Computational Harmonic Analysis*, vol. 14, no. 3, pp. 257–275, May 2003. Cited on page 53.

[Sin64] R. C. Singleton. "Maximum Distance q-nary Codes". *IEEE Transactions on Information Theory*, vol. 10, no. 2, pp. 116–118, Apr. 1964. Cited on page 8.

[SJW09] K. Schober, P. Janis, and R. Wichman. "Geodesical Codebook Design for Precoded MIMO Systems". *IEEE Communication Letters*, vol. 13, no. 10, pp. 773–775, Oct. 2009. Cited on pages 49, 55, and 68.

[SKHN75] Y. Sugiyama, M. Kasahara, S. Hirasawa, and T. Namekawa. "A Method for Solving Key Equation for Decoding Goppa Codes". *Information and Control*, vol. 27, no. 1, pp. 87–99, Jan. 1975. Cited on page 14.

[Slo] N. J. A. Sloane. "Spherical Codes: Nice Arrangements of Points on a Sphere in Various Dimensions". 2015-05-21.
`http://neilsloane.com/packings/` Cited on page 55.

[Som29] D. M. Y. Sommerville. *An Introduction to the Geometry of n Dimensions.* Methuen & Co. Ltd., London, 1929. Cited on page 20.

[SS06] G. Schmidt and V. R. Sidorenko. "Multi-Sequence Linear Shift-Register Synthesis: The Varying Length Case". In "Proceedings IEEE International Symposium on Information Theory (ISIT)", pp. 1738–1742. Seattle, Jul. 2006. Cited on pages 13, 14, 81, and 103.

[SS12] S. Som and P. Schniter. "Compressive Imaging Using Approximate Message Passing and a Markov-Tree Prior". *IEEE Transactions on Signal Processing*, vol. 60, no. 7, pp. 3439–3448, Jul. 2012. Cited on page 26.

[SSB09] G. Schmidt, V. R. Sidorenko, and M. Bossert. "Collaborative Decoding of Interleaved Reed–Solomon Codes and Concatenated Code Designs". *IEEE Transactions on Information Theory*, vol. 55, no. 7, pp. 2991–3012, Jul. 2009. Cited on page 10.

[SSB10] G. Schmidt, V. R. Sidorenko, and M. Bossert. "Syndrome Decoding of Reed–Solomon Codes Beyond Half the Minimum Distance Based on Shift-Register Synthesis". *IEEE Transactions on Information Theory*, vol. 56, pp. 5245–5252, Oct. 2010. Cited on pages 5, 8, 10, 11, 12, and 13.

[TBM79] H. Taylor, S. Banks, and J. McCoy. "Deconvolution With the ℓ_1 Norm". *Geophysics*, vol. 44, no. 1, pp. 39–52, Jan. 1979. Cited on pages 23 and 35.

[TDHJS05] J. A. Tropp, I. S. Dhillon, R. W. Heath Jr., and T. Strohmer. "Designing Structured Tight Frames via an Alternating Projection Method". *IEEE Transactions on Information Theory*, vol. 51, no. 1, pp. 188–209, Jan. 2005. Cited on page 53.

[TH08] G. Takos and C. N. Hadjicostis. "Determination of the Number of Errors in DFT Codes Subject to Low-Level Quantization Noise". *IEEE Transactions on Signal Processing*, vol. 56, no. 3, pp. 1043–1054, Mar. 2008. Cited on pages 75, 80, and 81.

[Tib96] R. Tibshirani. "Regression Shrinkage and Selection via the Lasso". *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, Jan. 1996. Cited on page 25.

[TKK14] E. V. Tsiligianni, L. P. Kondi, and A. K. Katsaggelos. "Construction of Incoherent Unit Norm Tight Frames With Application to Compressed Sensing". *IEEE Transactions on Information Theory*, vol. 60, no. 4, pp. 2319–2330, Apr. 2014. Cited on page 55.

[TLD+10] J. A. Tropp, J. N. Laska, M. F. Duarte, J. K. Romberg, and R. G. Baraniuk. "Beyond Nyquist: Efficient Sampling of Sparse Bandlimited Signals". *IEEE Transactions on Information Theory*, vol. 56, no. 1, pp. 520 –544, Jan. 2010. Cited on page 29.

[TP14] A. M. Tillmann and M. E. Pfetsch. "The Computational Complexity of the Restricted Isometry Property, the Nullspace Property, and Related Concepts in Compressed Sensing". *IEEE Transactions on Information Theory*, vol. 60, no. 2, pp. 1248–1259, Feb. 2014. Cited on page 28.

[Tro04] J. A. Tropp. "Greed is Good: Algorithmic Results for Sparse Approximation". *IEEE Transactions on Information Theory*, vol. 50, no. 10, pp. 2231 – 2242, Oct. 2004. Cited on page 27.

[Tuk77] J. W. Tukey. *Exploratory Data Analysis*. Addison-Wesley, 1977. Cited on page 123.

[VB98] R. Venkataramani and Y. Bresler. "Further Results on Spectrum Blind Sampling of 2D Signals". In "Proceedings International Conference on Image Processing ICIP", vol. 2, pp. 752–756 vol.2. Oct. 1998. Cited on pages 20 and 78.

[Ver12] R. Vershynin. "Introduction to the Non-Asymptotic Analysis of Random Matrices". In Y. C. Eldar and G. Kutyniok (eds.), "Compressed Sensing: Theory and Applications", chap. 5. Cambridge University Press, Cambridge, New York, May 2012. Cited on page 29.

[VL14] M. Vaezi and F. Labeau. "Generalized and Extended Subspace Algorithms for Error Correction With Quantized DFT Codes". *IEEE Transactions on Communications*, vol. 62, no. 2, pp. 410–422, Feb. 2014. Cited on page 75.

[vLS66] J. H. van Lint and J. J. Seidel. "Equilateral Point Sets in Elliptic Geometry". *Proceedings of the Koninklijke Nederlandse Akademie van Wetenschappen: Series A: Mathematical Sciences*, vol. 69, no. 3, pp. 335—348, 1966. Cited on page 54.

[Wak06] M. B. Wakin. *The Geometry of Low-dimensional Signal Models*. Ph.D. thesis, Rice University, Houston, Texas, 2006. Cited on page 78.

[WB99] S. B. Wicker and V. K. Bhargava (eds.). *Reed–Solomon Codes and Their Applications*. Wiley-IEEE Press, Piscataway, NJ, Oct. 1999. Cited on pages 1 and 8.

[Wel74] L. Welch. "Lower Bounds on the Maximum Cross Correlation of Signals (Corresp.)". *IEEE Transactions on Information Theory*, vol. 20, no. 3, pp. 397–399, 1974. Cited on page 51.

[WF89] W. K. Wootters and B. D. Fields. "Optimal State-Determination by Mutually Unbiased Measurements". *Annals of Physics*, vol. 191, no. 2, pp. 363–381, May 1989. Cited on pages 54, 65, and 107.

[WF15] F. Waeckerle and R. F. H. Fischer. "Multistage Bit-Wise Receivers for 4D Modulation Formats in Optical Communications". In "Proceedings International ITG Conference on Systems Communications and Coding (SCC)", Hamburg, Germany, Feb. 2015. Cited on pages 68 and 73.

[Wol83a] J. K. Wolf. "Analog Codes". In "Proceedings IEEE International Conference on Communication (ICC)", pp. 310–312. Boston, MA, Jun. 1983. Cited on page 75.

[Wol83b] J. K. Wolf. "Redundancy, the Discrete Fourier Transform, and Impulse Noise Cancellation". *IEEE Transactions on Communications*, vol. 31, no. 3, pp. 458–461, Mar. 1983. Cited on pages 75 and 77.

[WZP+15] Y. Wang, J. Zeng, Z. Peng, X. Chang, and Z. Xu. "Linear Convergence of Adaptively Iterative Thresholding Algorithms for Compressed Sensing". *IEEE Transactions on Signal Processing*, vol. 63, no. 11, pp. 2957–2971, June 2015. Cited on page 23.

[XZG05] P. Xia, S. Zhou, and G. B. Giannakis. "Achieving the Welch Bound With Difference Sets". *IEEE Transactions on Information Theory*, vol. 51, no. 5, pp. 1900–1907, 2005. Cited on pages 49, 52, 54, 55, 64, and 65.

[Yal68] P. B. Yale. *Geometry and Symmetry*. Holden–Day, San Francisco, California, 1968. Cited on pages 17 and 20.

[YFZ12] N. Y. Yu, K. Feng, and A. Zhang. "A New Class of Near-Optimal Partial Fourier Codebooks From an Almost Difference Set". *Designs, Codes and Cryptography*, pp. 1–9, Sep. 2012. Cited on page 54.

[You01] R. M. Young. *An Introduction to Nonharmonic Fourier Series*. Academic Press, revised first ed., 2001. Cited on page 52.

[Yu12a] N. Y. Yu. "A Construction of Codebooks Associated With Binary Sequences". *IEEE Transactions on Information Theory*, vol. 58, no. 8, pp. 5522–5533, Aug. 2012. Cited on page 54.

[Yu12b] N. Y. Yu. "New Construction of a Near-Optimal Partial Fourier Codebook Using the Structure of Binary m-Sequences". In "Proceedings IEEE International Symposium on Information Theory (ISIT)", pp. 2436–2440. Jul. 2012. Cited on page 54.

[ZF12a] A. Zhang and K. Feng. "Construction of Cyclotomic Codebooks Nearly Meeting the Welch Bound". *Designs, Codes and Cryptography*, vol. 63, no. 2, pp. 209–224, May 2012. Cited on page 54.

[ZF12b] A. Zhang and K. Feng. "Two Classes of Codebooks Nearly Meeting the Welch Bound". *IEEE Transactions on Information Theory*, vol. 58, no. 4, pp. 2507–2511, Apr. 2012. Cited on page 54.

[ZFLR14] Q. Zhang, Y. Fu, H. Li, and R. Rong. "Optimized Projection Matrix for Compressed Sensing". *Circuits, Systems, and Signal Processing*, vol. 33, no. 5, pp. 1627–1636, Nov. 2014. Cited on page 70.

Publications Containing Parts of this Thesis

[LZB13] D. E. Lazich, H. Zörlein, and M. Bossert. "Low Coherence Sensing Matrices Based on Best Spherical Codes". In "Proceedings 9th International ITG Conference on Systems, Communications and Coding (SCC)", Munich, Germany, Jan. 2013. Cited on page 3.

[ZLB13] H. Zörlein, D. E. Lazich, and M. Bossert. "On the Noise-Resilience of OMP with BASC-Based Low Coherence Sensing Matrices". In "Proceedings 10th International Conference on Sampling Theory and Applications (SampTA)", pp. 468–471. Bremen, Germany, Jul. 2013. Cited on page 3.

[ZAB13] H. Zörlein, F. Akram, and M. Bossert. "Dictionary Adaptation in Sparse Recovery Based on Different Types of Coherence". In "Proceedings 2nd International Workshop on Compressed Sensing applied to Radar (CoSeRa)", Bonn, Germany, Sep. 2013. `http://arxiv.org/abs/1307.3901` Cited on page 3.

[ZLB15] H. Zörlein, T. Leichtle, and M. Bossert. "Sparsity Aware Simplex Algorithms for Sparse Recovery". In "Proceedings 10th International ITG Conference on Systems, Communication and Coding (SCC)", Hamburg, Germany, Feb. 2015. Cited on page 3.

[MRZB15] M. Mohamed, S. Rizkalla, H. Zörlein, and M. Bossert. "Deterministic Compressed Sensing with Power Decoding for Complex Reed–Solomon Codes". In "Proceedings 10th International ITG Conference on Systems, Communication and Coding (SCC)", Hamburg, Germany, Feb. 2015. Cited on pages 3, 86, and 115.

[ZB15] H. Zörlein and M. Bossert. "Coherence Optimization and Best Complex Antipodal Spherical Codes", 2015. `http://arxiv.org/abs/1404.5889` Cited on page 3.

Henning Alexander Zörlein

Curriculum Vitae

For reasons of data privacy, the curriculum vitae has been removed.