This electronic thesis or dissertation has been downloaded from the King's Research Portal at https://kclpure.kcl.ac.uk/portal/

Heuristics for computing sparse solutions for ill-posed inverse problems in signal and image recovery

Apostolopoulos, Theofanis

Awarding institution:
King's College London

The copyright of this thesis rests with the author and no quotation from it or information derived from it may be published without proper acknowledgement.

END USER LICENCE AGREEMENT

Unless another licence is stated on the immediately following page this work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International licence. https://creativecommons.org/licenses/by-nc-nd/4.0/

You are free to copy, distribute and transmit the work

Under the following conditions:

- Attribution: You must attribute the work in the manner specified by the author (but not in any way that suggests that they endorse you or your use of the work).
- Non Commercial: You may not use this work for commercial purposes.
- No Derivative Works - You may not alter, transform, or build upon this work.

Any of these conditions can be waived if you receive permission from the author. Your fair dealings and other rights are in no way affected by the above.

Take down policy

If you believe that this document breaches copyright please contact librarypure@kcl.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.

Download date: 26. Dec. 2024

Heuristics for computing sparse solutions for ill-posed inverse problems in signal and image recovery

Theofanis Apostolopoulos

May 2016

Department of Informatics
King's College London
Strand, WC2R2LS
London, United Kingdom

A thesis submitted to King's College, London in accordance with the requirements for the degree of Doctor of Philosophy

© 2016 Theofanis Apostolopoulos

Keywords: Compressive Sampling (CS), Compressed Sensing, sparse recovery, sparse representation, sparse approximation, non-linear optimisation, linear ill-posed inverse problems, l_0-heuristic, complexity theory, NP-hard problems, random matrices, applied harmonic analysis, compression, Unitary transforms, Vector spaces.

Abstract

For almost a century the famous theorem of Shannon-Nyquist has been very important in digital signal processing applications as the basis for the number of samples required to efficiently reconstruct any type of signal, such as speech and image data. However, signals and images are mainly stored and processed in huge files, which require more storage space, they take longer to transmit and they demand a large computational cost for processing. For this purpose many compression techniques have been introduced including the emerging field of Compressed Sensing (CS). CS is a novel and fast sampling and recovery process, which has attracted considerable research interest with several new application areas. By exploiting the signal and the measurements structure we are able to recover a signal from what was previously considered as highly under-sampled measurements, according to the Shannon-Nyquist criterion. This reconstruction is accomplished by finding the sparsest solution for an ill-posed system of linear equations, which is an NP-hard combinatorial optimisation problem. This thesis focuses on the l_0-norm based minimisation problem which arises from sparse signal or image recovery, using the CS technique. A new, fast heuristic is proposed to directly minimise a continuous function of the l_0 norm. This swarm based stochastic method provides better sparse solutions for highly under-sampled and over-sampled cases even under the presence of noise with small error and less time complexity, compared with several well-known competing approaches. The evaluation methodology includes different parameters of the l_0-heuristic and is based on measuring recovery error and execution time under various sparsity levels, sample sizes, sampling matrices and transform domains. The mathematical background of CS, including the key aspects of sparsity and incoherence in measurements are also provided, together with applications and further open research questions, such as weakly sparse signals in noisy environments.

"Amat victoria curam"
(Victory loves diligence)
Latin proverb

Acknowledgments

A long and copious journey has come to an end and I am finally ready to write the preface for my PhD thesis. At this moment, I would like to express my deepest gratitude and countless thanks to all of those who have helped me during my PhD. I would like to particularly thank Prof. Maxime Crochemore and Prof. Tomasz Radzik for their time, guidance, constant support and valuable comments and suggestions to improve my thesis. I would also like to thank Dr. Caroline Chaux and Dr. Igor Potapov for agreeing to be my external examiners and for their valuable comments and suggestions to further improve my PhD thesis. Last, but certainly not least I would like to thank my family and friends for supporting me throughout my PhD studies.

Theofanis Apostolopoulos

London, June 2015.

Disclaimer

I declare the following to be my own work, unless otherwise referenced, as defined by the University's policy on plagiarism. The following articles have been accepted for publication, as part of my PhD research, and some parts are reproduced in this work:

- Theofanis Apostolopoulos, A Heuristic for sparse signal reconstruction, ICCSW 2012 Workshop, OpenAccess Series in Informatics (OASIcs), Schloss Dagstuhl - Leibniz Center for Informatics, Vol. 28, pp. 8-14, September, 2012, doi: 10.4230/OASIcs.ICCSW.2012.8.

- Theofanis Apostolopoulos and Tomasz Radzik, A swarm based method for sparse signal recovery, Second Joint International Workshop on Applied Radio Systems Research and Smart Wireless Communications (ARSR/SWICOM 2013), pp. 1-5, Luton, May 2013.

- Theofanis Apostolopoulos, "A swarm based heuristic for sparse image recovery", ICCSW 2013 Workshop, OpenAccess Series in Informatics (OASIcs), Schloss Dagstuhl - Leibniz Center for Informatics, Vol. 35, pp. 3-10, September, 2013, doi: 10.4230/OASIcs.ICCSW.2013.3

- Theofanis Apostolopoulos, "Sparse signal recovery as a non-linear problem", BCTCS 2014 Workshop, EATCS Bulletin, vol. 113, p. 210, June 2014.

Contents

1 Introduction **21**

1.1 Overview of CS 21

1.2 The CS method vs traditional techniques 22

1.3 Sparse recovery as an optimisation problem 24

1.4 Methods for sparse recovery and stability 26

1.5 Application areas of CS 30

1.6 Contributions of the thesis 33

2 Background knowledge **37**

2.1 Fundamentals of signals 37

2.1.1 Basic types of signals 38

2.1.2 General signal formulation 40

2.2 Sampling techniques 42

2.2.1 The Dirac Delta and Impulse Response functions 45

2.3 Frequency domain 47

2.3.1 Time to Frequency transformation 47

2.3.2 The Discrete Fourier transform 52

2.3.3 CS method and Fourier transform 55

2.4 Probability models 58

2.4.1 Randomness in Compressive Sampling 58
2.4.2 Generating uniformly distributed random numbers 61
2.4.3 Generating normally distributed random numbers 62
2.5 Sparsity, incoherence and compression 65
2.5.1 Sparsity as compression 65
2.5.2 Incoherent measurements 68
2.5.3 A puzzling numerical problem 69
2.5.4 Key aspects of the CS method in a nutshell 71

3 The CS method 73

3.1 An example of sparse signal representation 73
3.2 Partial measurements 75
3.3 Mutual coherence 77
3.4 Recovery as an optimisation problem 78
3.5 Stability of the recovery 80
3.6 Sampling and reconstruction of a simple signal 81
3.7 CS as a sparse image recovery method 85
3.7.1 General Image representation 86
3.7.2 Sampling images 87
3.7.3 Sparse recovery in images 91

4 Sparse approximation as a problem 97

4.1 The Complexity of the sparse recovery problem 99
4.2 The Nullspace property 103
4.3 The Restricted Isometry Property 105
4.4 Stability of the solution 110
4.5 Design of CS matrices 114

4.6 Measurement bounds 116
4.7 Alternative recovery problems 118
4.7.1 The l_1 min problem 119
4.7.2 The l_2 min problem 121
4.7.3 The l_0 approximation min problem 124

5 Noisy or perturbed environments 127
5.1 Noise in signals 127
5.1.1 Problem definition 128
5.1.2 Solution formulation 129
5.2 Noise in images 130

6 Heuristics for the l_0-norm problem 133
6.1 Problem definition (revisited) 133
6.2 Problem reformulation 134
6.3 Convergence conditions for sparse recovery 138
6.4 Particle Swarm Optimization algorithms 142
6.5 Naive solution approaches 144
6.6 Design of the heuristic algorithm 148
6.6.1 The Classical Least Squares case 149
6.6.2 Regularised Least Squares 151
6.6.3 Generalised inverse as the best approximate solution 153
6.6.4 Initial Solution 155
6.6.5 Appropriate decrease sequence for the parameter 157
6.6.6 Solution Generation 159
6.6.7 Assumptions and conditions 161
6.6.8 Solution Correction 165

6.6.9 Algorithm Description 167

7 Algorithms for sparse recovery 173

7.1 Orthogonal Matching Pursuit method 174

7.2 Lasso optimisation method 176

7.3 Iterative Hard Thresholding method 177

7.4 Iteratively Reweighted methods 179

7.5 A variation of Steepest Ascent method 181

8 Packages for sparse recovery 183

8.1 The l_1 Magic package 183

8.2 The FOCUSS package 185

8.3 The NESTA package 186

8.4 The TwIST package 188

8.5 The CVX package 189

9 Experiments - Simulations 191

9.1 Test signals 193

9.2 Test images 195

9.3 Recovery error metrics 196

9.4 Experimental results in signals 198

9.4.1 Sparse recovery using the l_0-heuristic 198

9.4.2 Performance analysis 205

9.4.3 Dependence of parameters 206

9.4.4 Effects on sparsity in recovery 214

9.4.5 Test run for a real-valued noiseless vector 221

9.4.6 Test run for a complex-valued noisy vector 224

9.4.7 The size of the Sensing matrix 231
9.4.8 Different Sensing matrices and design 238
9.4.9 Experiments in transform domains 241
9.4.10 Application of filters 246
9.5 Experimental results in images 253
9.5.1 Experiments description 253
9.5.2 Experiments without noise 258
9.5.3 Experiments with noise 262

10 Concluding remarks and future research 269

A Finite dimensional vector spaces 305

B Definitions and Glossary 315

List of Figures

1.1 CS as a lossy compression and sparse recovery as an optimisation problem. . 23

3.1 Estimation of a real one-dimensional sparse vector using l_1 minimisation. . . 83

3.2 Recovery error of a real one-dimensional sparse vector using l_1 minimisation. 84

9.1 Estimation of a simple real sparse vector using l_0 minimisation. 201

9.2 Recovery error of a simple real sparse vector using l_0 minimisation. 202

9.3 Estimation of a simple complex sparse vector using l_0 minimisation. 203

9.4 Recovery error of a simple complex sparse vector using l_0 minimisation. . . . 204

9.5 Plot of the execution of the l_0 heuristic in CPU cycles. 207

9.6 Plot of the quality of the solution generated by the l_0 heuristic. 208

9.7 Plot of the quality of the solution of the l_0 heuristic through iterations. . . . 215

9.8 Sparse recovery of an unknown highly under-sampled real vector using the l_0 heuristic. 217

9.9 Sparse recovery of an unknown moderately under-sampled real vector. 218

9.10 Sparse recovery of a real highly under-sampled real random vector. 219

9.11 Sparse recovery of a noisy highly under-sampled complex random vector. . . 222

9.12 Plot of benchmark results showing average time and recovery error of a real noiseless random vector. 225

9.13 Plot of benchmark results showing average execution time of a real noiseless random vector. 226

9.14 Plot of benchmark results showing average time and recovery error of a complex noisy random vector. 229

9.15 Plot of benchmark results showing average execution time of a complex noisy random vector. 230

9.16 Sparse recovery of a random vector under different sample sizes. 235

9.17 Sparse recovery of a real random highly sparse vector using the l_0 heuristic. . 236

9.18 Sparse recovery of a weakly sparse complex random vector using the l_0 heuristic. 237

9.19 Sparse recovery from random sample sizes drawn from different probability distributions. 242

9.20 Plot of a signal in time domain and representation of the samples collection. 247

9.21 Representation of a signal in a transform domain for samples collection. . . . 248

9.22 Representation of samples collection under different transform domains. . . . 249

9.23 Sparse recovery of a signal using partial measurements from transform domains. 250

9.24 Sparse recovery of a real sparse vector under different noisy sample sizes. . . 254

9.25 Sparse recovery of the Phantom image without noise using l_0 heuristic, l_1-Magic, TWIST and FOCUSS. 260

9.26 Sparse recovery of the Checkerboard image without noise using l_0 heuristic, l_1-Magic, TWIST and FOCUSS. 261

9.27 Sparse recovery of the Circles image with noise, using l_0 heuristic, l_1-Magic, TWIST and FOCUSS. 265

9.28 Sparse recovery of the Checkerboard image with noise, using l_0 heuristic, l_1-Magic, TWIST and FOCUSS. 266

List of Tables

1 Table of Definitions 19

2 Table of Notations 20

9.1 Comparison Table of different initial solutions under different sample sizes for the l_0 heuristic. 211

9.2 Comparison Table of different approaches for generating new solutions under different sample sizes and iteration numbers for the l_0 heuristic. 213

9.3 Comparison Table of sparse recovery of three images without noise (Phantom, Circles and Checkerboard). 267

9.4 Comparison Table of sparse recovery of three images with noise (Phantom, Circles and Checkerboard). 268

Table 1: **Table of Definitions**

Singular X	iff	$\det(X) = 0$.
Upper triangular X	iff	$X_{ij} = 0$ whenever $i > j$.
Lower triangular X	iff	$X_{ij} = 0$ whenever $i < j$.
Triangular X	iff	X is either upper or lower triangular.
Hermitian X	iff	$X^* = X$, $X_{ij} = \bar{X}_{ji}$.
Normal X	iff	$XX^* = X^*X$.
Orthogonal X	iff	$X^TX = XX^T = I$, since $X^T = X^{-1}$.
Singular X	iff	$det(X) = 0$ and X is square.
Symmetric X	iff	$X^T = X$, $X_{ij} = X_{ji}$.
Unitary X	iff	$X^*X = XX^* = I$, since $X^* = X^{-1}$.
Orthonormal X	iff	$\langle X_i, X_j \rangle = 0, (i \neq j)$ and $\|X\|_{l_2} = 1$.
Positive X	iff	all its elements strictly $X_{ij} > 0$.
Non-negative X	iff	all its elements $X_{ij} \geq 0$.
Positive definite X	iff	$X^{*T}AX > 0$, $A \neq 0$ vector and X is non-singular square.
Positive semi-definite X	iff	$X^{*T}AX \geq 0$, $A \neq 0$ vector and X is non-singular square.
Periodic $f(x)$	iff	$f(x + c) = f(x)$, $\forall x \in \mathbb{R}$ and $c \in \mathbb{Z}$.
Convex $f(x)$	iff	$f(p_1x_1 + p_2x_2) \leq p_1f(x_1) + p_2f(x_2)$, $\forall p_1, p_2 > 0$, $(x_1, x_2) \in \mathbb{R}^2$ and $p_1 + p_2 = 1$.
Concave $f(x)$	iff	$f((1-\lambda)x_1 + \lambda x_2) \geq (1-\lambda)f(x_1) + \lambda f(x_2)$, $\forall x_1, x_2 \in S$, $S \subseteq \mathbb{R}^2$ and $\lambda \in [0, 1]$.
Linear independent	iff	$c_1v_1 + \ldots + c_nv_n = 0$ implies $c_1 = \cdots = c_n = 0$ for any set $S = \{v_1, \ldots, v_n\}$ with $v_1, \ldots, v_n$ vectors in $\mathbb{R}^n$.

Table 2: **Table of Notations**

$\mathbb{R}$	$\triangleq$	set of real numbers.
$\mathbb{C}$	$\triangleq$	set of complex numbers.
$\mathcal{R}(C)$	$\triangleq$	real subspace of transformation matrix C.
$\mathbb{R}^N$	$\triangleq$	N element vector with real values or vector space with scalar field.
$\mathbb{R}^{M\times N}$	$\triangleq$	$M \times N$ element vector with real values.
$\text{Real}(x)$, $\text{Imag}(x)$	$\triangleq$	the real and imaginary part of a complex variable x.
$L^2(S)$	$\triangleq$	vector space of square-integrable functions on a domain $S \subseteq \mathbb{R}$.
$f \star g$	$\triangleq$	convolution between functions f and g.
$\mathcal{N}(\mu, \sigma^2)$	$\triangleq$	Normal probability density function with mean μ and variance σ^2.
$\epsilon \sim \mathcal{N}(\mu, \sigma^2)$	$\triangleq$	ϵ is normally distributed with mean μ and variance σ^2.
X_i	$\triangleq$	i-th element of vector X.
δ_{ij}	$\triangleq$	Kronecker delta function, $\delta_{ij} = 1, i = j$.
θa	$\triangleq$	partial derivative of order a.
$f'(x)$	$\triangleq$	differential (first derivative) of a function $f(x)$.
$\Theta\Omega$	$\triangleq$	boundary of an open set Ω.
X^T	$\triangleq$	transpose of vector X, $X^T = \|x_{ji}\|$.
X^*	$\triangleq$	adjoint (conjugate transpose) of vector X, $X^* = \|\bar{x}_{ji}\|$.
I_N	$\triangleq$	identity matrix of dimension $N \times N$.
e_i	$\triangleq$	unit vector with i-th entry equal to 1.
diag	$\triangleq$	diagonal vector, $D = \|d_i * \delta_{ij}\| = \text{diag}[d_i]$.
X^{-1}	$\triangleq$	inverse of vector X, $X^{-1}X = XX^{-1} = I$.
$\bar{a}$	$\triangleq$	complex conjugate, e.g. if $a = x + iy$ then $\bar{a} = x - iy$.
$\det(X)$	$\triangleq$	determinant of square matrix X.
$\text{rank}(X)$	$\triangleq$	rank of matrix X, maximal number of linearly independent rows.
$\text{trace}(X)$	$\triangleq$	trace of X, $\sum_{i=1}^{n} x_{ii}, X \in \mathbb{R}^{N\times N}$.
$kernel/Null(A)$	$\triangleq$	$\{u \in X : Y(u) = 0\}, A : X \to Y$.
$\|N\|$ or $\sharp(N)$	$\triangleq$	cardinality of a finite set N.
$O(.)$	$\triangleq$	Asymptotic upper bound of a method.
T	$\triangleq$	$T = 1/f = 2\pi/\omega$, period of signal (seconds/cycle).
$\sup(f(x))$	$\triangleq$	$\{x \in M \| f(x) \neq 0\}, f : M \to N$, set theoretic support.
$\|.\|_F$	$\triangleq$	Frobenius matrix norm (square root of sum of absolute squares).
$\|.\|_{l_2}$	$\triangleq$	l_2 norm (euclidean vector norm).
$\|.\|_{l_1}$	$\triangleq$	l_1 norm (absolute value vector norm).
$\|.\|_{l_0}$	$\triangleq$	l_0 norm (number of non-zero entries vector norm).
$\|.\|_{l_\infty}$	$\triangleq$	l_∞ norm (maximum value vector norm).
$\|.\|_{TV}$	$\triangleq$	total variation (semi) norm (l_2 of discrete gradient).
$\|\langle,\rangle\|$	$\triangleq$	absolute value of inner product, e.g. $\|\langle(x_1,\ldots,x_n),(y_1,\ldots,y_n)\rangle\| = \|x_1y_1 + \ldots + x_ny_n\|$.

Chapter 1

Introduction

1.1 Overview of CS

For over half a century the sampling theory has been focusing on the well known Nyquist-Shannon criterion which states that a signal (or image) has to be sampled at a rate greater than twice its bandwidth [1] in order to be effectively reconstructed. In particular, the Nyquist-Shannon rate theorem requires that the sampling rate must be at least twice the highest frequency of the original signal. Recently the scientific focus has shifted to the under-sampled technique called Compressive Sensing (CS).

Compressive Sensing (CS) or Compressed Sensing, Compressive Sampling, Sketching and Heavy-Hitters are just different terms referred in the literature in order to describe a recently developed technique for reconstructing and representing a signal from highly incomplete frequency samples or measurements. Typically these measurements are far below the expected number of samples required by the Nyquist-Shannon sampling theorem (or the Nyquist rate theorem). This methodology was introduced by the mathematicians Candès [39, 42, 46], Tao [43, 48], Romberg [169, 170] and Donoho [74–76] as an effective way to reduce the sampling, re-construction and computation costs for important physical signals having a certain property concerning their structure or regularity (i.e. their energy or largest values

[1] In signal processing and related disciplines in general, bandwidth is expressed in terms of the difference between the highest-frequency and the lowest-frequency signal component or term. Frequency is the rate that a repetitive event or more precisely the number of events are repeated in a time interval [136, 157, 186].

are concentrated on a small number of elements) and their randomly collected measurements, in a noiseless environment.

In order to apply the CS technique a signal or image must be what we call sparse or compressible and the sampling or sensing method to be random [11, 46, 75]. A sparse or compressible signal is a specific type of signal that can be stored in a compressed/sparse form by expressing it in terms of a linear combination of elementary/primitive elements which are called atoms (i.e. coefficients of signal basis). Moreover, such types of signals can be well-approximated (i.e. they have small recovery error) using a small number of basis elements or have a small finite number of degrees of freedom in each time interval (discrete number of coefficients or amplitudes in time). In essence, many signals are sparse if they contain many coefficients close or equal to zero in some domain or representation basis. Another important assumption is about the measurements performed in the signal which have to be incoherent. This means that they are random in such a way so as to extract the maximum amount of information from a given signal without much perceptual loss [46, 119, 169]. Every signal with this representation (in a fixed basis or transform domain) can be efficiently recovered from a much smaller number of random measurements, in contrast to conventional recovery approaches, which act as a universal measurement (representation) basis with high probability. In general, the required number of samples needed for efficient recovery is proportional to the information content of the signal which can be significantly lower (at least half of the lower order) than the number of samples required by traditional sampling methods [39, 41]. This is actually a lossy compression and the decompression is performed as a solution of an optimisation problem which is the main theme of this thesis.

1.2 The CS method vs traditional techniques

The traditional (practical) way of signal processing involves initially the acquisition of the whole signal (sampling the whole signal), then the computation of the complete set of transform coefficients and finally the compression of the data (signal). This conventional approach computes and encodes the largest coefficients (i.e. trading off signal representation complexity) and then discards all the unnecessary ones after the sampling process [39, 46, 119]. This long-established approach is clearly very computationally expensive (twice the bandwidth of the transform domain) and a sheer waste of valuable sensing resources in the calculation of the complete set of large coefficients, the majority of which is discarded

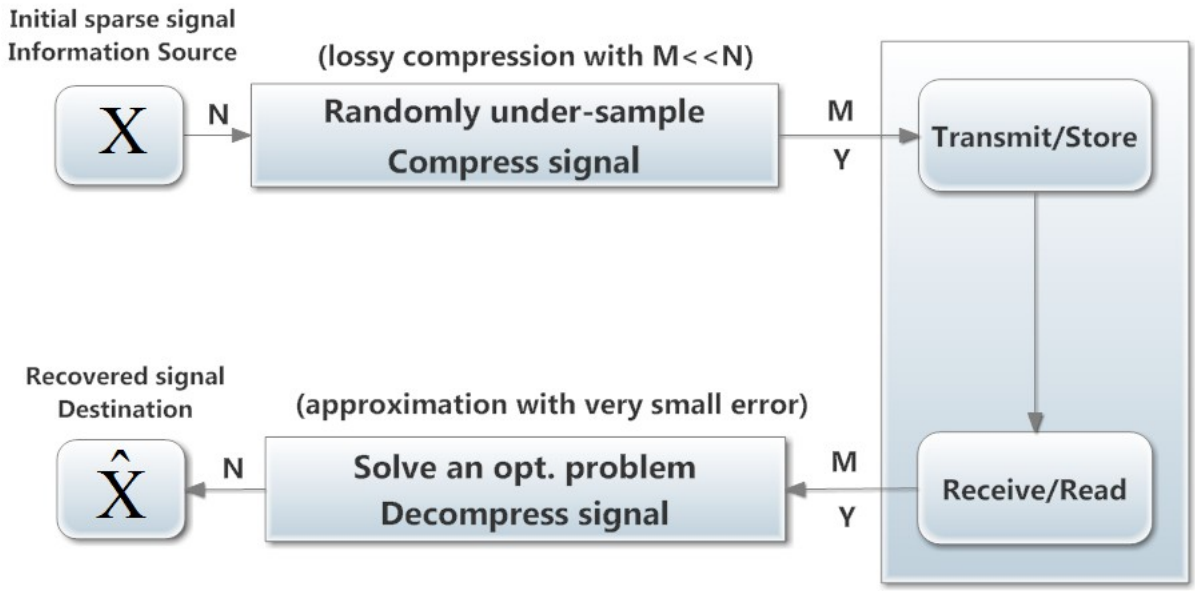

Figure 1.1: The CS technique as a lossy (random) compression approach and the recovery or decompression as a solution of an optimisation problem. Note that the original signal X (one dimensional vector) is not the same as the recovered one $\hat{X}$ but only a good approximation with small recovery error.

afterwards due to compression. Clearly, the new theory of Compressive Sampling is opposed to this traditional way as the signal is sampled and compressed simultaneously (thus the name) at a greatly reduced rate in order to keep only the most important elements/features (half the bandwidth of the transform domain). This scheme is represented graphically in Figure (1.1), where we assume a discrete communication channel (the signal is a sequence of discrete time values). The initial N element vector X (i.e. signal) is randomly under-sampled and compressed so as to efficiently be transmitted or stored as an M element vector ($M \ll N$). Then the compressed sampled vector is read or received and decompressed by solving a non-linear optimisation problem. Note that this is a lossy compression and thus what we will derive is a good approximation of the original vector ($\hat{X}$) with small recovery error.

Compressive sampling is based on the idea that not all measurements are equally important and thus discarding some measurements does not affect the recovery of a signal/image [11, 39, 46, 75]. This particular aspect is very important in a number of statistical applications where the number of variables or parameters N is much larger than the number of

observations M. The recovery in such cases is not affected by the method of data acquisition, or sensors in general, as a new recovery method for better reconstruction can be incorporated in the whole process [72, 119, 145, 153]. In fact, it has been shown that as the number of measurements decreases the reconstruction error decreases at an optimal rate up to some standard point [39, 75]. All these nice characteristics make CS a robust, highly efficient and computationally inexpensive method [11, 75, 119].

1.3 Sparse recovery as an optimisation problem

In many research fields, we want to recover a vector $X \in \mathbb{R}^N$ (representing a digital image or signal), from linear measurements (samples) $Y \in \mathbb{R}^M$ ($M \ll N$) possibly corrupted by additive noise [11, 41, 75]:

$$Y_i =< X, C_i >, i = 1, \ldots, M, \quad \text{or} \quad Y_{M\times 1} = C_{M\times N} X_{N\times 1}, \tag{1.1}$$

where Y_i represents the i-th element of vector Y and C_i is the i-th row vector ($C_i \in \mathbb{R}^N$) used to acquire information about the unknown signal by sampling it. In essence, the aim is to find the best fit of the given data (Y) as a linear combination of a small number of the basis functions C (sometimes referred to as over-complete basis or dictionary). In this case we have an ill-posed inverse problem with much fewer measurements than unknown values; a problem which typically arises in many areas, such as radiology and biomedical imaging [39, 42, 46]. Based on the mathematician's Jacques Hadamard definition on these types of problems we say that a problem is well-posed (i.e. the opposite of an ill-posed problem) if there is a unique solution for all the data given and this solution depends continuously on the data [113, 194]. Indeed, numerically calculating an approximate solution of the system of linear algebraic linear Equations in (1.1) amounts to deterrent (perturbed) data not uniquely defined, as the system is ill-conditioned (more unknowns than equations). Moreover, in most cases the matrix C is not square and thus its inverse C^{-1} does not exist which means that SVD and interpolating methods cannot be used in this case.

Recently, it has been suggested [11, 42, 46, 75] that solving the under-determined system of linear equations defined in (1.1) can be modelled in an equivalent way as an optimisation

principle:

$$\min_{\hat{X}} \|\hat{X}\|_{l_0} \quad \text{s.t.} \quad \|C\hat{X} - Y\|_{l_2} \leq \epsilon, \tag{1.2}$$

where the norm $\|.\|_{l_2}$ represents the Euclidean distance to be decreased between the measurements and the data, while the norm $\|.\|_{l_0}$ counts the number of non-zero components which enforces sparsity (minimum number of non-zero entries). The parameter ϵ indicates the error term assumption (data misfit relaxation), in the collection of noisy measurements Y, with bounded values $\|\epsilon\|_{l_2} \leq c$ with $0 < c < 1$ a small constant. The aim is to find the sparsest solution of the non-convex problem (1.2) in a form of an estimate $\hat{X}$ of the original vector X, given the highly incomplete and inaccurate measurements Y. This process is also called signal recovery (reconstruct the original signal when $\epsilon = 0$) or de-noising (remove the noise from the signal when $0 < \epsilon < 1$). Since most of recovery methods end up performing some sort of smoothing (relaxation) operation on X to produce an approximate $\hat{X}$, the process is sometimes also called smoothing [28].

We can easily see that in the problem (1.2) for the case of $\epsilon = 0$ the constraints are linear while the objective function is non-linear, while for $\epsilon > 0$ both the objective function and constraints are non-linear. In fact this problem is NP-Hard (reduction from the Set Covering problem) [92, 146]. Instead of replacing the l_0 norm based problem with its convex equivalent (l_1 norm based problem assuming that $\|CX\|_{l_2} \approx \|X\|_{l_2}$ based on the number of measurements) as has been suggested in [11, 39, 42, 75], we will approximate it with a smooth, easy to differentiate function subject to the same as previously constraints [6, 142, 161, 199, 213]:

$$\|X\|_{l_0} \approx f(\|X\|, \sigma) = \sum_{i=1}^{N} 1 - f(\|X_i\|, \sigma) = N - \sum_{i=1}^{N} \exp(-\frac{|X_i|^2}{2\sigma^2}), \tag{1.3}$$

where $f(\|X\|, \sigma)$ is a smooth non-linear objective function which is minimised together with the real parameter σ. The aim now is to gradually decrease both the objective function $f(\|X\|, \sigma)$ of the problem in (1.3) together with the σ parameter, which is actually a decreasing sequence of constants $[\sigma_1, \sigma_2, \ldots, \sigma_j]$. In fact σ parameter determines the quality of the approximation: the larger the σ, the smoother the function $f(\|X\|, \sigma)$ but the worse the approximation to the l_0 norm and vice versa. It has been shown in [6, 142, 161] that

the solution approaches to the sparsest one when $\sigma \to 0$, while for small values of σ the approximation becomes more difficult due to a number of local minima.

In this thesis, the problem in Equation (1.3), given the constraints in (1.1), is solved by a novel, swarm-based stochastic heuristic which aims to achieve a fast sparse recovery with a reduced number of measurements compared to the l_1 minimisation principle and other sparse recovery approaches. This is achieved by an iterative process which finds an approximation of the l_0-norm based problem, viewed as a combinatorial optimization problem. The l_0 heuristic uses the gradient of the function defined in (1.3) to rank the slightly different feasible solutions carried by swarms at each iteration. With the use of a stochastic step in the generation of the solution, the heuristic avoids stalling and thus being trapped into the numerous local minima of the problem in (1.3). In general, the quality of the solution provided is very good in terms of linear execution time and small recovery error under the presence of highly under-sampled measurements with and without noise. Efficient sparse solution of over-determined linear systems (over-sampled measurements where $M \gg N$) is also possible using the l_0 heuristic, compared to other sparse recovery methods which are only conjectured to work. For further details see Chapter 9.

1.4 Methods for sparse recovery and stability

Due to the fact that the measurement process in (1.1) is not adaptive and the reconstruction process in (1.2) of a signal/image is non-linear, a variety of measurement and recovery algorithms have been developed and proposed during the recent years. These algorithms are referred to CS literature as sparse recovery algorithms and include conventional linear programming algorithms [42, 130], which replace the combinatorial problem in (1.2) with a convex optimization problem and exploit the problem's structure and some tailored combinations, such as a re-weighted l_1 norm approach [45], the least squares variations [29, 52, 56] and modified Newton based methods [13, 17, 38, 92]. Most methods in this category start from an initial solution guess $X^{(0)}$ and then gradually generate a sequence of solutions iteratively $X^{(k)}$, $k = 1, 2, \ldots$ from the previously generated solution. Other major computational techniques for solving sparse approximation problems are greedy approaches[2]. This selection

[2] A general notion of a greedy approach, we adopt in this thesis, refers to an algorithm that selects an item from a sequence that is the most "appropriate" or "best" according to some criterion set, without keeping the previous choices or steps made [62, 147].

of choices does not follow any pattern and the algorithm simply selects the "best" current choice. Sometimes greedy algorithms are also called exhaustive search [135, 147], such as IHT [21, 22] and OMP [35, 177] and their variations [196, 197], which iteratively refine a sparse solution by successively identifying one or more components that yield the greatest improvement in quality (e.g. minimising the residual $R = \|Y - CX^{(k)}\|_{l_2}$).

In general, these methods cannot guarantee to find the sparsest solution to the problem and they are usually computationally more complex than other sparse recovery algorithms. Although many tailored approaches have been introduced, such as AIHT, STOMP, ROMP, PMP [21, 35, 72, 80, 148], they still remain very computationally expensive and approximate in reconstruction performance, as they do not consider the signal's structure in terms of sparsity and sampling pattern (see [141, 196, 215] for further discussion and analysis). A simple typical example of failure in greedy methods is when a linear combination of the transformed vector (signal or image) happens to be highly correlated with a third vector belonging to the same dictionary, fact which will result in wrong solution choices and thus non-optimal representations [132]. Recently, we have also seen interest in heuristic algorithms directly solving the l_0 norm based problem in (1.2) [6, 142, 161] or even model-based compressive sampling algorithms, which use additional a-priori information such as the tree structure of wavelet coefficients to guide the recovery process [66, 82, 118]. It comes with no surprise the plethora of striking sparse recovery methods for sparse approximation being available since CS is an important and popular topic of research in recent years. For a quick overview of the most common and efficient solvers see [132, 196, 215].

Nevertheless, many interesting classes of sparse approximation problems solved by numerous computationally tractable algorithms have been introduced. We will not provide a complete overview of the possible recovery algorithms here. Instead, we will merely outline the most important approaches, which are computationally practical and lead to provably correct solutions under predefined conditions as we will see in Chapter 4. More details about the methods and their corresponding recovery approaches can be found in Chapter 7 and Chapter 8. The most popular basic recovery algorithms ([13, 38, 42]) solve the following problem, often known as Basis Pursuit, de-noising, or simply as l_1-minimisation:

$$\min_{\hat{X}} \|\hat{X}\|_{l_1} \quad s.t. \;\; C\hat{X} = Y (= CX), \tag{1.4}$$

where $\hat{X} = [\hat{X}_1, \hat{X}_2, \ldots, \hat{X}_N]$ is a sparse vector estimate representing the signal/image which can be recovered from a small number of linear measurements $Y = [Y_1, Y_2, \ldots, Y_M]$ with $M \ll N$. The $\|.\|_{l_1}$ norm denotes the sum of absolute values. Despite the number of quick solvers available, these methods are still very slow to converge for large scale problems. Several improvements have been proposed to increase the quality of approximation and handle noisy cases, such as re-weighted l_1 [45] or l_2 norm [52, 56] and Dantzig selector (l_∞ norm) [13, 49]. In case of the Dantzig selector the l_1 norm based objective function remains the same but it is subject to slightly different set of constraints, namely:

$$\min_{\hat{X}} \|\hat{X}\|_{l_1} \quad s.t. \;\; \|C^T(C\hat{X} - Y)\|_{l_\infty} \leq \epsilon, \tag{1.5}$$

where $-1 \leq \epsilon \leq 1$ is a small quantity measuring the estimation error for recovering $\hat{X}$ while projecting back to N elements (instead of M) using the transpose C^T. In the case of re-weighting techniques the new value of the solution obtained is recalculated in every iteration using a simple weight:

$$W^{(t)} = ((\hat{X}^{(t-1)})^2 + \lambda)^{p/2-1}, \tag{1.6}$$

where t is the current iteration of the method, $0 \leq \lambda \leq 1$ is a small regularisation parameter (in case $\hat{X}^{(t-1)}$ is close to zero) depending on $\hat{X}$ values and $p \in [0, 2]$. Note the special case where $p = 0$ where the weight becomes $W^{(t)} = \log((\hat{X}^{(t-1)})^2 + \lambda))$ [196].

In addition to the general optimisation principle defined in Equation (1.4) there is also a variety of greedy and iterative methods such as IHT and OMP (see Chapter 7) which treat vector C^TY as a rough estimate of X and obtain $\hat{X}$ by iteratively identifying likely non-zeroes. Alternatively, some sparse recovery methods try to minimise an objective function which measures both the quadratic error term and a sparsity regularisation term [25, 28, 96, 196]:

$$\min_{\hat{X}} \frac{1}{2}\|C\hat{X} - Y\|_{l_2} + T\|\hat{X}\|_{l_1}, \tag{1.7}$$

where C is a dense matrix of corresponding dimension and $\|.\|_{l_2}$ denotes the standard Euclidean (l_2) norm. The term $\|C\hat{X} - Y\|_{l_2}$ is the data fitting term while the $T\|\hat{X}\|_{l_1}$ is the regularisation term, with T a non-negative parameter controlling the sparsity of solution.

The presence of additive term of l_1 norm very often increases the sparsity of optimal solution (see [17, 18, 59, 116] for more details). This feature is used in many areas, such as signal processing and statistical inference. In general, the objective in (1.7) is a non-smooth convex function, where several gradient based methods are used for generating a solution. An alternative yet similar problem formulation which is intermediate for the l_0 and l_2 norm based minimisations can be defined as follows [16, 26, 52, 59]:

$$\min_{\hat{X}} \frac{1}{2}\|C\hat{X} - Y\|_{l_2} + T\|\hat{X}\|_{l_0} \tag{1.8}$$

Note that IRLS [52, 56], FOCUSS [116] and TWIST [17, 18] are a family of recovery algorithms that treat these types of problems and their variations. Although they are much quicker than BP methods the quality of their solution estimation is worse especially for weakly sparse or approximately sparse vectors [142, 196].

Along with the development of many efficient sparse recovery algorithms there has also been significant interest in ensuring the necessary conditions so as the recovery methods can obtain provably accurate estimations. One of the first known assumptions is the Restricted Isometry Property (RIP) of the Sensing matrix C which essentially requires that $\|CX\|_{l_2} \approx \|X\|_{l_2}$ for any K-sparse vector (signal or image with K-non zero entries). However, any direct construction of matrix C that satisfies this property turns out to be very difficult; in fact it is NP-Hard [40, 41]. On the other hand, it is possible to show that if we construct C at random, then with high probability it will satisfy RIP [39, 189]. In fact, there is a number of different distributions, such as Normal, Bernoulli and Rademacher ensemble (i.e. each entry is either $1/\sqrt{M}$ or $-1/\sqrt{M}$ with equal probability) where the entries of C matrix are drawn so as to efficiently achieve dimensionality reduction in the sampling process. All these measurement matrices are universal with respect to the sparsity inducing basis and with high probability satisfy RIP conditions, given that the measurements are $M \approx K\log(N/K)$ (see for example [39, 40, 42] for details on measurement bounds). The performance guarantee is finite and proportional to measurements size and approximation error [39, 42, 50]:

$$\|X - \hat{X}\|_{l_2} \leq \epsilon_1 K^{-1/2}\|X - X_K\|_{l_1}, \tag{1.9}$$

where $\hat{X}$ is the derived estimate of X using the l_1-norm based optimisation approach, X_K is the best possible K-sparse approximation to the original X and $0 \leq \epsilon_1 \leq 1$ is a small absolute

constant, while the l_2 norm refers to the Euclidean distance. In the special (theoretical) case where $X = \hat{X}$ the Equation (1.9) holds as an equality. However, this is not the case in real life applications where X_K represents the best possible K-term (sparse) approximation of the N-term X ($K \ll N$) using a different domain of representation, such as the application of a Unitary transform (Fourier transform). In this case we have [39, 50]:

$$\frac{\|X - X_K\|_{l_2}}{N} \leq K^{-\alpha}, \tag{1.10}$$

where α is a small absolute constant depending on the values of X. For example if X_K represents the best 250-term sparse approximation of 1000-term X using FFT (Fast Fourier transform) then $\alpha \approx 50$. Further discussion regarding Unitary transforms and sparse approximation can be found in Section 2.3 and Section 2.5, while specific cases, measurement bounds and assumption models in the context of CS are provided in Chapter 4.

1.5 Application areas of CS

The CS technique has a number of applications apart from the obvious areas of signal and image recovery [132, 186]. In radiology and biomedical imaging for instance we can typically collect much fewer measurements about an image of interest than the known number of pixels (i.e. every pixel represents a sample). Representative examples are MRI, Hyper Spectral Imaging and tomography images (e.g. x-ray transmission tomography, positron-emission tomography (PET) and single-photon emission tomography (SPECT)) [42, 134, 170, 213]. Computational Biology and particularly gene expression constitutes another example, where highly dimensional data frequently arise [154, 186]. Gene expression studies typically involve a relatively low number of observations while the total number of genes is aggregated in thousands. Similar problem arises in seismic and geophysical data analysis in general [82, 154, 186]. Other examples in statistical signal processing, error correction and non-parametric estimation include the recovery of a continuous-time curve or surface from a finite number of noisy samples [48, 66, 72, 74, 142]. Other notable areas include sensor networks [71], analog-to-digital Information Conversion [139], Compressive Radar [88], Astronomy [58, 186], Communications [102], Cognitive Radio network [122], Acoustics and Time-Frequency Analysis [48, 87], Geo-science and Remote Sensing [58], Robotics and Control [109], Optics, Holography and MRI [123, 132, 169, 186], Fault Identification [191, 200] and Music Genre

Classification [54]. For a more detailed treatment of direct and indirect applications of CS the interested reader is advised to visit the Compressive Sampling website of Rice University [100] which contains several application areas updated regularly.

In all these applications the underlying problem is the estimation of a sparse signal, image or system of parameters. Some efforts have also been made for incorporating the information about sparsity into the estimation problem in order to design more accurate or less complex estimation algorithms with a number of other potential applications. In fact, it is believed that further progress and experimentation in CS may have far reaching impact across many new areas of statistics [39, 196]. However, CS in general, is particularly useful in cases where hardware limitations affect the collection of samples directly in a frequency domain due to the fact, for example, that measurements are expensive or not possible to be obtained [85].

CS can also be used in cases where the computations at the sensor end are very expensive but at the receiver end very cheap, on condition that a signal or image is sparse in a known basis (transform domain). MRI (Magnetic Resonance Imaging), agiography, tomographic imaging and optical microscopy are some of the first applications of CS, where we assume sparsity and incoherence in measurement samples, which are selected from the $2D$ continuous Fourier transform (limited frequency domain samples) [42, 85, 170]. In these cases the application of CS can be viewed as a low pass filter directly applied to Fourier coefficients for efficient sparse recovery, which can integrate sensing, compression and processing at the same time. Other well-known areas of application for sparse representations and descriptions are denoising, smoothing, channel coding and data compression where CS can be used for efficient error recovery in transmission and storage [28, 39, 87]. Here, the dictionary or basis is known to both sender and receiver as a universal encoder, which is randomly designed with or without knowledge of structure of signal. Using the values obtained by the compressed, sparse signal transmitted by the sender, the receiver can reconstruct approximately the original one without much loss.

Efficient recovery from under-sampled measurements under the presence of noise constitutes the current scientific forefront together with further research in the interplay between signal/image models, sampling operators and efficient new reconstruction algorithms. In this context we seek to recover $\hat{X} \in \mathbb{R}^N$ from partial measurements $\hat{Y} \in \mathbb{R}^M$ $(M \ll N)$ under some additive noise $e \in \mathbb{R}^M$ in the model and under the assumption that the elements of C matrix follow an unknown probability distribution instead of the Normal distribution. In

this case the noisy model of the under-determined system of equations can now be defined as follows [43, 76, 105]:

$$\hat{Y}_{M\times 1} = C_{M\times N} X_{N\times 1} + e_{M\times 1}, \tag{1.11}$$

where $e \sim N(\mu, \sigma^2)$ is a small bounded noise with $0 \leq \|e\|_{l_2} \leq 1$. This typically models simple errors (e.g. loss in transmission, malfunctions, environment noise, etc.) in the sampling process which are stochastic and not correlated with signal vector X we seek to recover.

Another scope of current scientific research is conducted in the design of the specification of accurate signal/image models that exploit signal/image structure, the development and study of sampling systems that preserve the relevant signal/image information and the derivation of fast and provably efficient algorithms to reconstruct the signal/image form these samples. In this case the model of the under-determined system of equations can now be defined as follows [39, 79, 186]:

$$\hat{Y}_{M\times 1} = \Phi_{M\times N} \Psi_{N\times N} X_{N\times 1} + e_{M\times 1}, \tag{1.12}$$

$e \sim N(\mu, \sigma^2)$ typically models simple errors in the domain of representation of signal $\Psi_{N\times N}$ which is the sparsifying transform for enhancing sparsity (e.g. Fourier transform), Φ is the Sensing or Sampling matrix (which defines the random selection of the columns in Ψ) and $X \in \mathbb{C}^N$ is the unknown vector we want to recover from the partial measurements $\hat{Y} \in \mathbb{C}^M$ in Ψ domain of representation. The purpose is to define the optimal bounds for the measurements M and sparsity levels K for the best possible solution stability and robustness to data corruption ($0 \ll \epsilon \ll 1$) and compressibility (K as small as possible). The design of Ψ and Φ and how can be efficiently combined into the global Sensing matrix (compression and representation) is another area of research in CS. For example, it has been suggested in [6, 161, 199] that it is possible to design new Sensing/Compression matrices C, which follow different distributions than Normal distribution (e.g. Levy, Poisson, etc.), but can still derive optimal bounds for sparse recovery with much smaller number of measurements.

1.6 Contributions of the thesis

The aim of this thesis is to provide a more consistent view on the field of CS and introduce a new alternative approach for solving the l_0-norm minimisation problem, which arises in the sparse signal/image recovery problem. In particular, the aim is to develop an efficient method to recover any type of signal, such as speech and image data, using what was previously considered as highly incomplete and inaccurate (under-sampled) measurements. This is an ill-posed inverse problem, which can be solved as an l_0-norm based (non-linear) optimization problem (NP-hard problem) with the help of a new swarm based heuristic. By combining the concepts of neighbourhood and gradient based methods [112, 206, 214], this population based evolutionary (swarm based) heuristic finds an approximation of the l_0-norm based problem, viewed as a combinatorial optimization problem. This is achieved by iteratively re-weighting and minimising the difference between the solution and the observations based on a gradient of a function which approximates the l_0 norm. In each iteration every swarm calculates and carries a slightly different feasible solution based on the current best (optimal) solution, which is necessary so as to avoid being trapped to one of the numerous local minima. In general, the l_0 heuristic achieves fast and efficient recovery of signals and images, viewed as real and complex vectors with high sparsity. A detailed description of the l_0 heuristic method, including its parameters, initial and new solutions, is provided in Chapter 6.

In this thesis we will also define and present different recovery methods and optimisation principles for sparse recovery of both real and complex-valued vectors. These methods and principles use slightly different optimisation problems where the constraints are the same but the definition of the objective function changes (e.g. TV-norm and re-weighted l_2-norm metrics). Chapter 9 presents a number of experiments where we compare the performance of the l_0 heuristic with other sparse recovery methods, such as OMP, COSAMP, AIHT, LASSO, SL0, IRLS and packages, such as L1-Magic, NESTA, CVX, FOCUSS. The performance of each sparse recovery method is measured in terms of execution time (in CPU cycles) and recovery error (e.g. expressed as a mean-square error between the original and estimate) under different sparsity levels (from highly sparse to weakly sparse vectors), samples sizes (from highly under-sampled to over-sampled cases), types of Sensing matrices (Uniform, zero-one, Normal, exponential, geometric, Poisson, Pareto), Unitary transforms (FWHT, FFT, DCT) and different levels of noise during the sampling process.

The l_0 heuristic is able to recover real and complex-valued sparse vectors under linear time complexity, in contrast to other recovery methods, such as LASSO, l_1 minimisation principle, NESTA and L1-Magic, which require quadratic time complexity. Moreover, the l_0 heuristic can recover sparse vectors with small recovery error from highly under-sampled or over-sampled measurements, on condition that the vector is sparse enough. For example, for an 70-element vector with 10 non-zero entries, efficient recovery (10^{-1}) can be achieved from less than $23 - 25$ measurements, in contrast to 30 measurements required by AIHT, IRLS, LASSO and 35 measurements required by OMP and l_1 minimisation principle to achieve the same or better level of accuracy. Efficient recovery (10^{-50}) for the same vector is also possible from moderately under-sampled and over-sampled measurements ($50 - 80$ samples), for the l_0 heuristic, where the solution guarantees are only conjectured to work and sparse recovery methods, such as OMP, IRLS, AIHT, NESTA, fail. However, for samples sizes between 35 and 50 the recovery error of the l_1 minimisation principle is 10^{-40} and between 10^{-25} and 10^{-40} for the SL0 method, in contrast to the recovery error between 10^{-8} and 10^{-30} for the l_0 heuristic. Similar results are also possible under the presence of noisy measurements, on condition that we have bounded small additive noise and enough measurements (at least twice the sparsity level). Sparse recovery using the l_0-heuristic is also possible using sampling matrices with entries drawn from probability distributions (e.g. exponential, geometric, Poisson) which are not supported by the majority of sparse recovery methods, such as OMP, AIHT, LASSO, AIHT, NESTA. These probability distributions, in contrast to Normal and Rademacher distributions are not used in CS since they do not provide sufficient guarantees about the stability of the solution. For example, a 200-element real-valued vector with 20 non-zero entries can be efficiently recovered from 90 samples, drawn from the Poisson and Pareto distribution, with 10^{-10} recovery error.

Finally, it is also important to briefly note the drawbacks of the l_0 heuristic, some of which are also common to other sparse recovery methods; the need of randomness as an inherent step of the sampling process (common for all known sparse recovery methods), the a-priori knowledge of the sparsity level of the vector to be recovered (also common in OMP, AIHT, LASSO), the small bounded additive noise levels for efficient recovery (also common in OMP, AIHT, LASSO, NESTA) and the samples size (very efficient mainly for highly under-sampled or over-sampled cases). Towards this direction the use of different types of filters (hard-threshold, median and moving average) can be an important improvement

in the performance of the l_0 heuristic for sparse vector recovery from noisy under-sampled measurements (see Chapter 9 for further details).

The remainder of this thesis is organised as follows. Chapter 2 provides an introduction and brief analysis on signals, sampling techniques, measurements collection, images, sparsity, incoherence and Unitary transforms. Chapter 3 provides an analysis of the Compressive Sampling (CS) method using a simple example in signals and images. Chapter 4 discusses in detail the sparse approximation as an optimisation problem together with the necessary description on measurement bounds and the stability of the optimal solution. Chapter 5 provides an introduction on noisy and perturbed environments. Chapter 6 extensively describes the design and analysis of the l_0 heuristic. Chapter 7 describes the most representative algorithms used for sparse recover, while Chapter 8 describes the most important Matlab packages used for the same purpose. Chapter 9 describes the numerical experiments conducted, namely 10 for sparse signal approximation and 2 for sparse image approximation. Chapter 10 provides some concluding remarks and future work on the areas of Compressive Sampling and Sparse recovery in general.

Chapter 2

Background knowledge

This Chapter aims to provide key notions that CS theory relies on. In particular, we briefly describe the notions of image/signal analysis, randomness in sampling, sparsity, probability and transformations; all of which constitute the scientific fields or techniques that CS takes advantage for efficient compression and recovery of a signal or image. In later chapters we will define the CS method (Chapter 3), the recovery problem needs to be solved for sparse signal or image recovery (Chapter 4) and representative algorithms/packages in bibliography for formulating and solving relevant recovery problems (Chapters 7, 8). Other key notions defined and discussed include random matrices, norms, sparsity, Unitary transforms and incoherence. At this point we also have to pinpoint that CS is actually a lossy compression technique rather than just a sampling or signal/image processing procedure.

2.1 Fundamentals of signals

In Signal Analysis the term signal generally refers to a function that represents one or more independent variables of a set I (domain of a function) in one or more dependent variables of a set U (range of a function) [32, 155, 210]. In essence, a signal is a description of how one parameter varies with another parameter. For instance, voltage changing over time in an electronic circuit, sound and electromagnetic waves, or a brightness varying with distance in an image. A relevant notion is a system; it is any process that produces an output signal in response to an input signal [32, 157, 210]. Continuous systems input and output continuous signals, such as in analog electronics, while discrete systems input and output

discrete signals, such as computer programs computing values stored in arrays. Representative examples of signals with high research interest are human speech (function of time), video signals, music signals, voice signals, 3D images (representation of a certain property as a function of coordinates, such as the (x, y) spatial variables in a plane), biomedical signals (used in electrocardiography, MRI, PET, SPECT etc.) or even tracking signals (using some sort of sensors for finding the coordinates or velocity of vehicles) [136, 157, 205, 210]. Moreover, a voltage or current of a circuit, the telephone voice transmitted through the copper wires/lines or even the Dow Jones Industrial average can also expressed as a measurable quantity (function of one or more independent variables) in space or time and thus as a signal [157, 205, 210].

It is useful to categorise and characterise the signals based on their type. In particular, if the values of the variables of the signal belong to a continuous set, we have a continuous time signal, while in the discrete time signal the values of its variables belong to a discrete set [136, 157, 205]. Similarly, when the values of the signal cover a continuous set then the signal is specified as analog, while if its values cover a discrete set then we have a digital signal [136, 157, 205]. It is important to pinpoint that discrete time signals arise usually via sampling continuous time signals in some time interval using specialised interface circuits since computers and other digital devices are usually restricted to discrete time signals instead. In the CS theory, for simplicity, we only presume discrete time digital signals represented as vectors, which is a commonly adopted approach, see for example in [132, 186, 215].

2.1.1 Basic types of signals

Simply put, a signal is actually a time-varying function in any field relevant to signal processing, electrical engineering or communications. More precisely, a one dimensional signal expresses the change of a variable, let it be f, in terms of another variable, let it be t [157, 205, 210]. This can be defined as a function $f(.)$ such that $t \rightarrow f(t)$ [3]. Usually, the independent variable t refers to time or space and receives values from the domain I

[3] In general, a signal can be represented by two alternative approaches. The most common one is the time domain where the signal is usually represented as $f(t)$ for one dimension. An alternative representation as we will see in Section 2.3 is the frequency domain where the signal $F(\omega)$ is represented in terms of the independent frequency variable ω. Any signal can be represented in either domain as both representations are equivalent [99, 132, 186].

of the signal, while the value of the signal at time t is symbolised as $f(t)$ and is defined in some range of values in U [157, 205, 210]. If this variable is defined in a discrete space, which means that the set I is a discrete set then the corresponding signal is called discrete time signal. The most common cases of discrete time signals are defined in $I = \mathbb{Z}$ or when $I = \mathbb{Z}^+$, where $\mathbb{Z}^+$ defines the set of non-negative integer numbers. Such types of signals can be represented as a sequence of numbers where the independent variable is represented as t, the corresponding value of the signal with $f(t)$ and the chronological progress of the signal (signal cycle) as $t \to f(t)$.

Another important classification refers to whether a signal is represented by an aperiodic or periodic function $f(t)$. For practical purposes we usually consider signals which repeat themselves over a period of time interval without significant distortion. In CS theory, for simplicity and analytical purposes, we presume only discrete and periodic time digital signals. This is also the common approach of signals of scientific interest, even with long duration, see for example [39, 46, 132]. Some basic well-known discrete time signals, used for the representation of acoustic waves, vibrating structures, human speech and several mathematical and physical models, are the following [136, 157, 205, 210]:

- The unit step which is defined for $I = \mathbb{Z}^+$ as: $f_s(t) = 1, t \in \mathbb{Z}^+$
- The linear signal which is defined for $I = \mathbb{Z}^+$ as: $f_r(t) = t, t \in \mathbb{Z}^+$
- The sinusoidal signal (or pattern) is defined as: $f(t) = \sin(\omega t)$, or alternatively the cosine wave, a signal waveform with similar representation: $f(t) = \cos(\omega t)$, for $t \in \mathbb{Z}^+$ and a parameter $0 < \omega < \pi$. Both sine and cosine waves have identical shape and same frequency though the cosine wave leads the sine wave by 90 degrees of phase. Particularly each point on the cosine wave occurs exactly $\frac{1}{4}$ cycle earlier than the corresponding point on the sine wave [157, 205, 210]. Both sine and cosine waves are important in Fourier analysis and Unitary transforms Section (2.3, as more complex waveforms are defined in terms of sine and cosine waves.

Note that the signals we describe here are what we call "one dimensional" since the domain of defining a signal is characterized by only one variable. If we have a domain I defined by a two dimensional variable then the signal is defined as two-dimensional, while if we have a variable of m dimensions then the signal is defined as multidimensional or m-dimensional

[157, 205, 210]. In the first case the signal expresses the change of the dependent variable f in terms of the two independent variables t_1 and t_2 and the function $f(...)$ is defined in the domain $I = I_1 \times I_2$, such that $(t_1, t_2) \rightarrow f(t_1, t_2)$. In the second case we assume that the domain of I has the form of a Cartesian product $I = I_1 \times I_2 \times \ldots \times I_m$, where $I_1, I_2, \ldots, I_m$ are subsets of the set of integers $\mathbb{Z}$ [157, 205, 210]. In all cases we assume that the variables can get discrete values. A representative example of a two dimensional representation is the digital images stored in computers. As we will see in Section 3.7 an image can be represented as a two dimensional array of points of the same size as the image. Every element of this array represents the light intensity of the image's pixel and its coordinates, its position inside the array grid and thus in the image.

2.1.2 General signal formulation

As we discussed previously, in signal processing, a signal can be represented in time or space as a function $f(t)$, which is referred usually in the literature as a waveform and is defined as a combination of sines and/or cosines over a specific domain of numbers, usually discrete integers. In signal processing theory almost every periodic signal of period T_0 can be represented as a function $f(t)$, formed as a linear combination of coefficients (amplitudes) and waveforms (sines or cosines), which represent the basis of the signal. A general expansion of a signal can be given in trigonometric form as [136, 157, 205, 210]:

$$f(t) = \sum_{k=1}^{\infty} (\omega_k \cos(k\omega_0 t + A) + v_k \sin(k\omega_0 t + B)), \tag{2.1}$$

where ω_k and v_k are the peak amplitude of the wave, while A, B and ω_0 are the set of real numbers forming the coefficients, where every term $\omega_k \cos(k\omega_0 t + A) + v_k \sin(k\omega_0 t + B)$ define one harmonic of the function $f(t)$. For $k = 1$ the term is called first harmonic or fundamental harmonic [157, 205, 210]. In general, if we know all the frequency components ω_k and v_k then we can efficiently recover the signal $f(t)$.

In fact, function $f(t)$ represents the signal through time t, ω_0 is the fundamental frequency of the signal (i.e. the duration and the intensity of the signal) and A, B are some constants that represent the time offsets (phase or displacement offsets) of the signal [57, 157, 210]. Note that the angles $(k\omega_0 t + A)$ and $(k\omega_0 t + B)$ are usually measured in radians, where π (in radians) $= 180^o$, while ω_0 is sometimes also called angular frequency. Usually, the

values of ω_0 span in the range $[0, \pi]$, while the values of A, B span in the range $[0, 2\pi]$. It is also important to mention here that the frequency ω_0 can also be represented as [136, 157, 205, 210]:

$$\omega = 2\pi f_0, \text{ or } f_0 = \frac{1}{T_0} \tag{2.2}$$

where T_0 is the period (i.e. the number of occurrences of a repeating event per unit time), f_0 is the frequency number of rotations in a period of time, usually 1 second, and ω_0 is again the angular velocity or angle in an amount of time. In general, f and ω use different units to express the frequency; f expresses the frequency in terms of revolutions or number of complete cycles, while ω uses radians for the same purpose and since one revolution (or full circle) is 2π radians (or 360 degrees), the two terms are related as $\omega = 2\pi f$.

CS theory is based on the idea that by exploiting the signal structure we are able to reduce the sampling size and thus the time required for digital processing and storage space. Throughout this thesis, for simplicity reasons we will only consider finite dimensional problems where every signal can be measured and presented as a vector in a finite dimensional subspace of $\mathbb{R}^N$ or $\mathbb{C}^N$, $F = [F_1, F_2, \ldots, F_N]$, where F_i (for $i = 1, \ldots, N$) represents the i-th element of vector F. More generally, we can model a signal F as the linear combination of N elementary waveforms, called signal atoms, such that [11, 46, 79]:

$$F = W\Psi = \sum_{i=1}^{N} W_i \Psi_i, \tag{2.3}$$

where vector $W = W(F) = [W_1, W_2, \ldots, W_N]$ is called the representation coefficients of F in dictionary $\Psi = [\Psi_1, \Psi_2, \ldots, \Psi_N]$, which is a $N \times N$ matrix which represents the domain of representation. The elements Ψ_i are columns of Ψ which represent a vector of atoms with sines, cosines, or any other combination with/without complex numbers. For example, based on general signal representation defined in Equation (2.1) W represents ω_k and v_k elements of a signal, while Ψ represents the elements $\cos(k\omega_0 t + A)$ and $\sin(k\omega_0 t + B)$ (in the main diagonal of Ψ). In fact, the coefficients vector W is characterized by the same sparsity profile or support as the signal F under Ψ representation, which means that W is sparse (it has many close to 0 entries), as we will see in Section 2.5. Note also that in order to facilitate later analysis, we adopt a more general notation that F_i represents the i-th element of signal

or image F.

We also assume that every discrete time signal has a sparse representation on a fixed basis, meaning that one can store only a small number of adaptively chosen transform coefficients without much loss. If the given signal is not sparse, we can easily enhance its sparsity by applying a Fourier or a similar transform, such as the Wavelet or Discrete Cosine transform, which simply changes the values of Ψ. Signals or images that are sparse in Ψ are those that can be written exactly as a superposition of a small fraction of the atoms in the family Ψ_i. In contrast to conventional sampling methods, CS is a protocol that senses and compresses a signal simultaneously in Ψ domain without going through the intermediate stage of acquiring all the samples. This is a distinctive property of CS entailing that the measurement process is not adaptive, in the sense that it will not depend on the explicit knowledge of the signal F. Note that in order to achieve this outcome, we need to use dictionaries Ψ in which sets of signals can be decomposed in a sparse manner and to find and use measurement ways that are incoherent (random) in terms of these dictionaries. Further details about the transformations can be found in Section 2.3, while a comprehensive comparison of different transform domains and how they affect the efficient signal recovery can be found in Section 9.4.9. Note also that the aim of this thesis is not to explicitly define and compare all the different transforms, but merely to briefly describe them in terms of their contribution towards enhancing the sparsity of a vector (image or signal).

2.2 Sampling techniques

In signal processing theory, sampling is a process of converting a function of continuous time or space (i.e. a signal or an image) into a function of a discrete time or space (a numeric sequence) [136, 157, 205, 210]. This is in general a process of reduction of a continuous time to a discrete time, as a sequence of samples of a continuous time signal in proper time intervals [136, 157, 205, 210]. A general sampling process derives this set of discrete values (points) uniformly spacing. In a simple mathematical form the sampling process can be expressed as a multiplication [157, 205, 210]:

$$f^*(t) = \sum_{k=1}^{\infty} f(kT_s)S(t, kT_s), \tag{2.4}$$

where $f^*(t)$ is the sampled output function at (kT_s) intervals (T_s is the sampling period or time), $f(kT_s)$ is the original function (signal in time) $f(t)$ sampled at (kT_s) intervals (replace the continuous variable t with the discrete value or index kT_s) and $S(t, kT_s)$ is the sampling function in time at (kT_s) intervals so as to convert a continuous (analogue) function $f(t)$ to a discrete (digital) function $f^*(t)$. This sampling function can be for example the unit pulse expressed as Dirac delta function as $S(t, kT_s) = \delta(t - kT_s)$, where T_s is the sampling rate or period which can also be expressed as sampling frequency $f_s = \omega_s/2\pi = 1/T_s$.

Obviously, the continuous signal $f(t)$ which can be recovered back from the sampled one $f^*(t)$ is not exactly the same, but a good approximation. This reconstruction (recovery from samples) as a mathematical form can be expressed as [157, 205, 210]:

$$f(t) = \sum_{k=1}^{\infty} f^*(kT_s)p(t - kT_s), \tag{2.5}$$

where p is a general or rectangle pulse function which is not based on any prior knowledge of signal's support. For example:

$$p(t) = \begin{cases} 0 & \text{for } |t| > 1/2 \\ 1/2 & \text{for } |t| = 1/2 \\ 1 & \text{for } |t| < 1/2, \end{cases} \tag{2.6}$$

or more generally [157, 205]:

$$p(t) = h\delta((t - c)/b), \tag{2.7}$$

where h is the height offset, c the centre offset and b the width offset of a time interval. Note also that the ordinary sampling scheme follows the pattern that the sampling rate must be at least the highest frequency of the original signal (Nyquist-Shannon theorem/limit for a signal). For example, given a simple signal $f(t) = 5\cos(2\pi 250t)$, the sampling frequency has to be $f_s \geq 500$ Hz, or the sampling period $T_s = 1/f_s \leq 0.002$ secs. This might represent for example, a 2 cycles/second cosine wave being sampled at 1000 samples/second. Undersampling (sampling using a lower frequency) can cause aliasing which means that samples taken

are not enough for efficient recovery (less than twice the bandwidth [4]). In a more general perspective the reconstruction process is exact (without loss of information) based on the following Sampling Theorem (i.e. Nyquist-Shannon Theorem):

Theorem 1: [157, 205, 210] *The sampling frequency must be at least twice greater than the highest frequency of the signal so as the signal can be exactly reconstructed from its samples. In other words, if B is the highest frequency of the original signal, then the sampling rate $f_s \geq 2B$ samples/second, so as to efficiently recover a signal $f(t)$ from its samples $f^*(kT_s)$ at (kT_s) intervals ($T_s = 1/f_s \leq 0.5B$). Alternatively, given an infinite sequence of samples a signal can be perfectly reconstructed if the sampling rate exceeds $2B$ samples per second.*

In other words, if a signal (function $f(t)$) does not have any frequency higher than B Hertz, then it can be obtained back by collecting a series of points spaced $1/(2B)$ seconds apart. Therefore, the value of B, which is also called Nyquist frequency, is indeed the highest frequency needed for deriving meaningful information (i.e. no error recovery) from a set of data/samples. For example, if we restrict the frequency $\omega = 2\pi f$ of a signal to take the values $(2\pi/N, 4\pi/N, 6\pi/N, \ldots, \pi)$, where N denotes the even number of terms/sums of the signal, then the upper bound (Nyquist frequency) is set to π, while the lower bound (lowest - fundamental frequency) is set to $2\pi/N$. Note that both fundamental and Nyquist frequencies are mainly used in cases where the signal is periodic (i.e. self-repeated) with period T so that $f(t + kT) = f(t)$, for all integer values of $k \geq 0$. However, the Nyquist frequency does not depend on N, but solely on the sampling frequency, in contrast to the lowest frequency that depends on N. This means that the higher the (necessary) frequency the longer the time period needed for measurements or observations.

On the other hand, if we sample in a frequency lower than B then the reconstructed (from its samples) signal will not be identical to the original signal. This ambiguity is called aliasing (under-sampling) and is valid for all types of signals or images [157, 205, 210]. Alternatively, if we sample with frequency higher than $4B$ then the signal will be aliased and appear as a lower frequency signal, since we have more than required samples [157, 205, 210].

[4] Bandwidth is the difference between upper and lower values of frequencies, magnitudes, wavelengths or in general energies, within a set of values in a signal [157, 210]. It is the main idea behind several signal processing applications, including systems (see Section 2.1) and filters (see Section 9.4.10).

A common remedy in such cases is the application of a filter or convolution mask, such as averaging (see Appendix B) [157, 205]. As we will see in the next Section, the sampling process is usually achieved by using a Dirac delta function δ already described, but sometimes in bibliography we also consider a sampling process as a sum of uniformly spaced discrete Dirac delta functions, aspect which will not consider in this thesis.

As a final note, it has already become clear that the conventional wisdom of sampling process dictated by the famous Nyquist-Shannon Theorem requires a very large number of samples as the sampling frequency is much more than twice the one required by the signal's limited bandwidth. Compressive Sampling or Compressed Sensing (CS) method, which is the main topic of this thesis, follows a different approach which states that we are still able to recover almost perfectly a signal or image from what was previously considered as highly incomplete and inaccurate samples according to Nyquist-Shannon Theorem. This relatively new sampling approach uses the prior knowledge that signals are sparse and the samples random, while traditional ways of sampling are designed for frequency band-limited signals, with no prior knowledge or use of over-complete signal representations [132, 186, 215]. According to CS theory a large class of signals can be economically (or compactly) represented, allowing more flexibility in signal representation, more adaptivity to their morphological content and thus entailing more effectiveness in many processing tasks (restoration, separation, compression and estimation) [41, 45, 75]. CS immediately became the new scientific forefront of the current research due to these characteristics which have high applicability mainly in the areas of signal and image acquisition and processing, medical imaging, error coding and geophysical and astronomical data analysis [100, 132, 186, 215].

2.2.1 The Dirac Delta and Impulse Response functions

For discrete time sampling of signals the dirac delta function (or simply dirac function) can be used as a simple measurement basis (sampling matrix) for a signal. In essence, this is not a function in a strict mathematical definition. It is rather a generalised function or distribution which maps a function, let's say $\phi(t)$, into a real line by producing the value $\phi(0)$ [57, 157, 205]. A general mathematical definition of Dirac delta function in the discrete

time domain is [57, 157, 205]:

$$\delta(t) = \begin{cases} 0, & t \neq 0 \\ \\ \infty, & t = 0 \end{cases} \tag{2.8}$$

which provides three fundamental properties, namely [207]:

1. $\delta(t-c) = 0$, for $\forall t \neq c$,
2. $\sum_{c-\epsilon}^{c+\epsilon} \delta(t-c) = 1$, for $\forall t \neq c$ and $\epsilon > 0$,
3. $\sum_{c-\epsilon}^{c+\epsilon} \phi(t)\delta(t-c) = \phi(c)$, for $\forall t \neq c$ and $\epsilon > 0$.

Then the collection of k-th sample y_k at time t' can be defined as [57, 157, 205]:

$$y_k = \int_{t'-\epsilon}^{t'+\epsilon} \phi(t)\delta(t-t')dt = \phi(t') \tag{2.9}$$

for $k \in M$, with M the collective number of measurements (samples) in time t and $\phi(t)$ a discrete time periodic signal as a function of time t. In all cases we are interested in the values of $\phi(t)$ in terms of $\delta(t)$ and not in the value of $\delta(t)$ itself since $\delta(t)$ is simply a generalised function used for the collection of samples. A slight variation, also used in discrete time sampling, is called impulse response function [57, 157, 205]:

$$h_t = \begin{cases} 0, & t \neq 0 \\ 1, & t = 0 \end{cases} \tag{2.10}$$

Then the output (k-th measurement) y_k at time $t = t'$ is given by:

$$y_k = \sum_{t=-\infty}^{\infty} \phi(t)h_{t-t'} = \phi(t'), \qquad t = t'. \tag{2.11}$$

In effect, impulse function provides a way to analyse signals as one sample at a time. By this way a discrete time signal can be decomposed into a group of components called impulses,

as an approximation of short duration in time. An impulse is a signal composed of all zeros, except a single nonzero point [57, 120].

2.3 Frequency domain

The representation of functions as a superposition of sines and cosines, using for example the Fourier transform, has become a highly efficient and well-known technique with significant impact on the numerical analysis and analytic treatment of signals and images. Fourier transform generally focuses on estimating the spectrum or the spectral density function of a given time series (such as signals) over a range of frequencies. Signals or images are often analysed or modelled in terms of their frequency spectrum since their are often very large in size in time domain (usually $N \gg 10^6$ [132]). This is achieved by frequency domain techniques, such as the Fourier or Wavelet transforms, which are applicable to both continuous and discrete time signals. This change in the domain of representation is usually very convenient so as to decrease the number of the non-zero entries of the transformed signal and therefore to enhance the performance of the sampling procedure. In fact, this transform from the time domain t to the frequency domain ω, is lossless (the recovery error is less than 10^{-15} in Matlab, using FFT transform in images) as the frequency coefficients of the transformed signal represent the contribution of each sine and/or cosine (based on the transform) function of the original signal at each frequency. These transforms can be used and extended for use in a variety of related and sophisticated reconstruction applications for signals. The category of this type of transforms (called Unitary transforms) is widely used in many areas such as electrical engineering, geophysics, marine science and meteorology [99, 186]. All their advantages and their properties will become apparent progressively as we present them in the next sections. At the moment we can say that each Unitary transform has its own characteristic properties in the frequency domain as its suitability highly depends on these aspects as well as on the existence of a proper computational algorithm [99, 132].

2.3.1 Time to Frequency transformation

Most physical and natural phenomena can be modelled as signals which are usually in a continuous time. To represent them in computers we have to sample them first and then process them in the frequency domain. For convenience we usually decompose signals (or images) over a family of functions so as to be well-localised both in time and frequency,

which has a number of applications in the fields of signal processing and harmonic analysis [114, 205]. The key component in such functions, called time-frequency atoms, is the time to frequency transformation, which affects the choice of these atoms and the decomposition offering different properties. These benefits of the transformation can be mainly used for two reasons [99, 136, 175, 186]: for the analysis and processing of a given time signal and for achieving compression. In fact, significant properties which are not seen explicitly in time domain can be enhanced in frequency domain. For example, a simple sinusoid is transformed into an impulse in the frequency domain, meaning that it is consisted of only one frequency component. This transformation is also important for the use of low, high or band-pass filters which work as a threshold for some frequencies and reject some others. Another important aspect is compression applications. transforms, such as the Fourier transform, can concentrate the energy of a signal in only a few impulses or magnitudes. This means that we can represent a signal with much fewer samples in frequency domain than in the conventional time domain. Then these samples can be used for transmission, storing, quantisation and in general for many other processing steps. This is also the basic concept of various audio codecs, such as .flac and .mpeg-4.

As we have seen briefly in Section 2.1.2 (more details in Section 2.3.3), given an N element signal f (in a form of vector), it can be represented as a set of linear combination of N elementary functions Ψ_N ($N \in \mathbb{Z}$) in any domain as follows [99, 136, 175]:

$$\hat{f} = \sum_{i=1}^{N} W_i(f)\Psi_i, \tag{2.12}$$

where $W_N(f)$ is the coefficient vector of f which depicts the signal's behaviour in the domain to which Ψ belongs. For example, in a simple case where the basis Ψ is a combination of cosines and sines, similar to Equation (2.1) where we define the general formulation of signals and the coefficients vector $W_N(f)$ can be the ω_k and v_k values.

In general, in any transformation it is desired that $\hat{f} = f$, where f is assumed to be the signal in its original physical domain (i.e. time domain) and $\hat{f}$ the signal in a transformed domain (usually the frequency domain). In other words, we expect that the transition from one domain to another will be lossless (f will be the same in both transform domains), condition which depends on the proper choice of the elementary functions in Ψ. These elementary

functions can be orthogonal (no correlation among them) or non-orthogonal [99, 136, 175]. The orthogonality property has a major advantage that the energy (Euclidean distance) in the difference between $\hat{f}$ and f is minimum and thus negligible [5]. Moreover, in order to avoid scaling mismatches it is important the elementary functions to also be of unit magnitude (i.e. orthonormal) [99, 136, 175]. Finally, if these elementary functions can completely represent the entire range of the signal in the domain to which they belong, they are also called Basis functions. Orthogonality is also preferred since the coefficients of each elementary function remain unchanged. This is mainly because the weighted basis has no correlation in each other and so there is no affection of the basis by the addition of each other. In Fourier transform complex sinusoids are used for basis with lots of applications, as the elementary functions are periodic and this property can lead to a periodic signal as a linear combination of periodic functions. Before proceeding with the definition and analysis of the transforms we first need to provide a general definition of the linear transformation and then the definitions about when a transform is Unitary and thus Orthogonal.

Definition 1:[99, 114, 136] *Let M quantities $T_0, T_1, \ldots, T_{M-1}$ be linearly and uniformly* [6] *expressed by N other quantities $F_0, F_1, \ldots, F_{N-1}$. This can be written as:*

$$T_j = \sum_{i=0}^{N-1} C_{j,i} F_i, \quad j = 0 : (M-1). \tag{2.13}$$

The transformation of the quantities $F_0, F_1, \ldots, F_{N-1}$, into the quantities $T_0, T_1, \ldots, T_{M-1}$ is called linear transformation, where the coefficients of the transform $C_{j,i}$ form a $M \times N$ matrix C.

Note that every matrix C determines this transform uniquely. In matrix form the previous

[5] According to Parseval's relation, in the ideal case of uniformly sampled signals, the energy of the signal in the time domain remains equal to that in the frequency domain [99, 136, 175] (see Section 2.3.2)

[6] The term "uniformly expressed" means that the difference (expressed in l_1 or l_2 norm) between two elements in C remains the same across all elements $C_{j,i}$. Fourier uniformity, for example, between any two signals is the condition where the phase difference between two Fourier components of the same frequency band remains exactly the same across all the bands (see Section 2.3.2 for details). More general, in signal processing uniformity implies uniform sampling intervals [99, 205].

transformation can be written as $T = CF$:

$$\begin{bmatrix} T_0 \\ T_1 \\ \vdots \\ T_{M-1} \end{bmatrix} = \begin{bmatrix} C_{0,0} & C_{0,1} & \dots & C_{0,N-1} \\ C_{1,0} & C_{1,1} & \dots & C_{1,N-1} \\ \vdots & \vdots & \ddots & \vdots \\ C_{M-1,0} & C_{M-1,1} & \dots & C_{M-1,N-1} \end{bmatrix} \times \begin{bmatrix} F_0 \\ F_1 \\ \vdots \\ F_{N-1} \end{bmatrix}, \tag{2.14}$$

where $F = [F_0, F_1, \dots, F_{N-1}]^T$ and $T = [T_0, T_1, \dots, T_{M-1}]^T$ are column vectors, while C is the transform matrix. This is called a N point discrete transformation and all sequences F on which this transformation is executed, form the domain of this transformation [99, 114]. The numbers defined in the N dimensional sequence F are called orders of the transformation [99, 114].

A simple example is the N point Fourier transform whose matrix C forms the Basis of the transform, as a collection of functions, $C = \exp(2\pi i/N)$. This sequence is sometimes called the kernel or basis of the frequency domain (Fourier transformation) with order equal to N [99, 114]. Note that this is a one-dimensional transform (only one basis used in a form of vector), which converts a one-dimensional vector of N elements (e.g. defined in $\mathbb{R}^N$) from time (or space) domain representation to frequency domain representation. In general, the dimension of a transformation (or more generally in vector spaces S) is the cardinality (i.e. the number of one-dimensional vectors) of the basis used for the transformation [3, 164, 216]. Note in the previous Equation (2.14) we have an N-dimensional transformation since we have N-column vectors in the transform matrix C. For more information on this topic see Appendix A.

Definition 2:[99, 114, 205] *If the Basis functions of an N-point transform (N-point discrete transformation) are defined by a finite set of samples, then such transformation is called discrete. If the Basis (or kernel) of a transformation is composed of real functions, then this transform is real, while if the Basis functions (or kernel) are complex, then the transform is complex.*

Definition 3:[99, 114, 136] *A transform C^{-1} is called an inversion of the transform C if, for every sequence F in the domain of C, the transform C over F is the sequence T of the domain of C^{-1} and the transform C^{-1} over T is the sequence over F. Assuming C^{-1} is*

the inversion of C as an N-point transform we have the following relation:

$$F_j = \sum_{j=0}^{M-1} C_{i,j}^{-1} T_j, \qquad i = 0 : (N-1), \tag{2.15}$$

where $C_{i,j}^{-1}$ are the coefficients representing the inverse of the transform matrix C (assuming C is a square matrix and thus $M = N$).

Definition 4:[99, 114, 164] *An N-point discrete transform C is called Unitary, if the transform matrix is unitary. A real Unitary transform (vector space over $\mathbb{R}$) is called Orthogonal. If C is an Unitary transform (e.g. Fourier, Wavelets, Discrete Cosine), then there exits an inverse transform C^{-1} whose Basis functions satisfy the following properties*:
1) Orthogonality condition (for rows):

$$\sum_{j=0}^{N-1} C_{i,j} C_{k,j}^{*} = \delta_{i,k} = \begin{cases} 1, & \text{if } i = k \\ 0, & \text{if } i \neq k \end{cases} \quad i, k = 0 : (N-1), \tag{2.16}$$

where δ is the Kronecker delta function and C^* is the adjoint of C. Note that column orthogonality must also be satisfied. Orthogonal is a real Unitary matrix. Note that Discrete transform matrices with complex numbers are defined on vector spaces over $\mathbb{C}$ (non-orthogonal) and do not preserve the dot product. Orthogonal matrices are the ones which preserve dot product (inner product in Euclidean space, defined as $\|.\| > 0$, i.e. $\|x, x\| > 0,\ \forall x \neq 0$).
2) Mutuality condition:

$$C_{i,j} = (C_{j,i}^{*})^{-1}, \quad \forall i \in N, j \in M \tag{2.17}$$

If the mutuality condition is satisfied then C is said to be a complete and orthogonal and thus it has an orthonormal set of basis functions. At this point, it is important to note some important definitions from Linear Algebra.

Definition 5:[99, 114, 205, 210] *If a matrix C is real, that is a matrix with real elements, then its adjoint (complex conjugate) C^* coincides with its transpose C^T, i.e. $C^* = C^T$. If C is a square matrix, it is called unitary if $CC^* = I$. If a matrix C is unitary, then its*

inverse C^{-1} coincides with its adjoint C^*, i.e. $C^{-1} = C^*$. Consequently, $\|CU\|_{l_2} = \|UC\|_{l_2}$ for every matrix U given that C is a unitary matrix. If a matrix C is orthogonal, then its inverse C^{-1} coincides with its transpose C^T, i.e. $C^{-1} = C^T$. Finally, if C is a normal matrix then $C^TC = CC^T$.

As a final remark, we have to pinpoint that in many applications, the sparsity of a vector (signal or image) occurs in the frequency domain. In these applications, a necessary transform, such as Fourier transform is an important pre-processing step. Dictionaries (Bases) of Unitary transforms are known to be redundant; thus the search for the vector that matches best the vector is limited to only a small sub-dictionary and hence efficient recovery is possible. However, as we will see, some of the proposed methods in the literature (Matching Pursuit methods) cannot handle the complex case.

2.3.2 The Discrete Fourier transform

Let's assume we have a finite-dimensional signal in time domain $f(t) \in \mathbb{C}^N$ (or $\mathbb{R}^N$) then the classical Discrete Fourier transform (DFT) $\mathcal{F}(f) = \hat{f} : \mathbb{C}^N \to \mathbb{C}^N$ (or $\mathbb{R}^N \to \mathbb{C}^N$) is defined by the following equation [99, 132, 136, 186]:

$$\hat{f}(\omega) = \sum_{t=1}^{N} f(t) e^{(-2\pi i \omega t)/N}, \tag{2.18}$$

where $\omega = 1, ..., N$ and $\hat{f}(\omega)$ is the transformed signal (spectrum). Note that from Euler's relation for complex number notation we have $e^{(-2\pi i\omega t)/N} = \cos((-2\pi\omega t)/N) + i\sin((-2\pi\omega t)/N)$.

The inverse Fourier transform does the opposite; transform data from the frequency domain back into the time domain. So, if we are given the values of the Fourier coefficients $\hat{f}(\omega)$ for all frequencies $\omega \in \mathbb{Z}_N$, then we can obviously reconstruct the signal $f(t)$ via the Fourier inversion formula which is given by the equation [99, 132, 136, 186]:

$$f(t) = \frac{1}{N} \sum_{\omega=1}^{N} \hat{f}(\omega) e^{(2\pi i \omega t)/N} \tag{2.19}$$

for all frequencies $\omega = 1, ..., N$, where $f(t)$ is the signal we want to recover. Since DFT is orthogonal the recovery of $f(t)$ is almost exact. This is based on the Parseval's theorem for

Fourier transform, which is written using inner product notation as [99, 132, 136, 186]:

$$< f(t), f(t) >= \frac{1}{2\pi} < \hat{f}(\omega), \hat{f}(\omega) > \tag{2.20}$$

where the inner product represents the signal's energy. For a time interval (t_0, t_1) the energy of a discrete time signal $f(t)$ can be defined as [114, 136]:

$$E(t_0, t_1) = \sum_{t=t_0}^{t_1} |f(t)|^2 \tag{2.21}$$

and more generally over an infinite time interval:

$$E_\infty = \sum_{t=-\infty}^{+\infty} |f(t)|^2 \tag{2.22}$$

Also, Rayleigh's Theorem states that:

$$\int_{-\infty}^{+\infty} |f(t)|^2 dt = \int_{-\infty}^{+\infty} |\hat{f}(\omega)|^2 d\omega, \tag{2.23}$$

which means that the total energy of a signal is the same in either domain.

If a function $f(t)$ is discontinuous at a point $t = t_1$ then we have to calculate the average value for the inversion formula in (2.19) [114, 136]:

$$f(t_1) = \frac{1}{2}[f(t_1^+) + f(t_1^-)], \tag{2.24}$$

where $f(t_1^+)$ and $f(t_1^-)$ represent the value of the function at point $t = t_1$ as we approach the positive and negative side respectively. A few interesting points have to be discussed here. Rotation, change of size and small quantisation errors or noise in an object (i.e. function) will not affect the phase and the magnitude of Fourier components, which can be defined as [114, 136, 205]:

$$\sqrt{\hat{f}_s^2(\omega) + \hat{f}_c^2(\omega)}, \tag{2.25}$$

which is the magnitude of Fourier transform and

$$\tan^{-1}(\hat{f}_s^2(\omega)/\hat{f}_c^2(\omega)), \tag{2.26}$$

which is the phase of the Fourier transform, with

$$\hat{f}_c(\omega) = \sum_{t=1}^{N} f(t)\cos(-2\pi\omega t/N) \tag{2.27}$$

$$\hat{f}_s(\omega) = \sum_{t=1}^{N} f(t) i \sin(-2\pi\omega t/N) \tag{2.28}$$

This aspect offers Fourier transform many advantages for template matching as it can show how an object is varying (distance between the centre of gravity [7] and points in the periphery of an object) given the object is periodic.

Notice also that both the DFT (defined in Equation (2.18)) and IDFT (defined in Equation (2.19)) involve the same operations except the sign of the exponent and the scale factor. These two functions (DFT and IDFT) are sometimes called Fourier transform pair. In fact there is one-to-one mapping between the spatial and frequency domain. The DFT maps an N (real or complex value) vector $f(t)$ on M complex-value vector $\hat{f}(\omega)$, while the inverse DFT maps the $\hat{f}(\omega)$ on $f(t)$ [132, 186]. Moreover, if $f(t)$ is complex then we need one complex multiplication for the term $f(t)\exp((-2\pi i\omega t)/N)$, which is equivalent to 4 real multiplications and 2 real additions [157, 186]. In general, the computation of $f(t)$ requires N^2 complex multiplications and $N(N-1) \approx N^2$ complex additions no matter what is the approach we have chosen to adopt [157].

Due to this computational cost, in practice, the DFT is computed using the Fast Fourier transform (FFT) algorithm, which is simply a computationally efficient way of obtaining the DFT coefficients based on the symmetries of the basis (matrix Ψ) in the cosine and sine functions so as to reduce the number of multiplications (providing that N is power of

[7] Centre of gravity is a simple geometric property which expresses the average location of an object [182]. For a signal (one dimensional vector) $f(t)$ the centre of gravity c is defined as $c = \sum_{t=1}^{N} f(t)/N$, while for an image $f(i,j)$ (two dimensional vector) the centre of gravity $\mu_{p,q}$ is defined as $\mu_{p,q} = \sum_i \sum_j i^p j^q f(i,j)$, where $p, q > 0$ are positive integers representing the order/power of the centre of gravity (centroids).

2) [99, 114, 132]. For two dimensional objects (i.e. images) FFT makes use of a series of one-dimensional transforms (in a $N \times N$ image we will have $2N$ 1D transforms as separable Fourier) [99, 114]. Finally, it is important to pinpoint that many authors use a slightly different definitions for the aforementioned DFT and IDFT (see [114, 132, 186]). For example, sometimes a constant $1/\sqrt{2\pi}$ is used outside the sum of the Equation (2.18) and then the Equation (2.19) has a symmetric form. Other variations use the constant $1/2\pi$ outside the sum of the equation (2.18) and the absence of minus sign in the exponential, i.e. $\exp((2\pi i\omega t)/N)$ in place of $\exp((-2\pi i\omega t)/N)$. These expressions are also equivalent as the definition of Fourier transform is more customary. In general, if the transform is written as $\hat{f} = A\sum_{-\infty}^{+\infty} f(t)\exp(-ib\omega t)$, then the inversion formula is $f = [|b|/(2\pi A)]\sum_{-\infty}^{+\infty} \hat{f}\exp(ib\omega t)$.

2.3.3 CS method and Fourier transform

Suppose now that the Fourier coefficients $\hat{f}(\omega)$ are known only for a small, limited subset of $\omega \in \Omega \subset 1, ..., N$ of size M, where M are the measurements of the signal taken with $M \ll N$. This means that these Fourier coefficients given are sampled on some partial subset $\Omega \subsetneq \mathbb{Z}_N$ of all frequencies ($\mathbb{Z}_N$: set of N integers). This partial Fourier coefficients matrix can be easily obtained by selecting M rows uniformly at random and re-normalising the columns so that they are unit-normed [39, 42, 43]. In this case the reconstruction of the signal is not so straightforward as the use of conventional methods of interpolating and reconstructing signals will not work. This is expected as there is not enough information to reconstruct the signal exactly in general; the initial signal $F(t)$ has N degrees of freedom (elements) while we are given only $|\Omega| = M \ll N$ [8] observations.

In CS we use a Unitary transform, such as DFT, to enhance the sparsity of an image before down-sampling it. We only change the domain of representation and thus we enhance its sparsity as many natural signals/images have concise representations when expressed in a convenient basis. For simplicity reasons, it is common to treat an $N \times N$ image as a $N := N^2$ signal vector and samples as a vector on the M frequencies ($M \ll N$); principle we will also adopt in this thesis.

Let's assume we have a noiseless signal $F(t)$ as previously, which we expand in an orthonormal basis as a superposition of spikes (e.g. canonical basis in $\mathbb{R}^N$, sinusoids or splines,

[8] The norm $|\Omega|$ denotes the cardinality of Ω.

wavelets or Fourier frequencies) $\Psi = [\Psi_1, \Psi_2, \ldots, \Psi_N]$ as $\hat{F}(\omega) = \sum_{i=1}^{N} F_i(t)\Psi_i$. Then the signal can be represented as a sparse linear combination of atoms in Ψ [132, 186, 215]:

$$\hat{F}_{N\times 1} = \Psi_{N\times N} F_{N\times 1}, \tag{2.29}$$

where F is the expansion coefficients corresponding to basis Ψ, where it is assumed to be sparse (compressible). This basis Ψ is a unitary $N \times N$ matrix of waveforms $\Psi_1, \ldots, \Psi_N$ as columns and $\hat{F}$ is the vector of frequency coefficients with respect to Ψ. So, the N-point DFT is expressed as an N-by-N matrix multiplication, where F is the original input signal and $\hat{F}$ is the DFT of the signal. Then F is sampled or sensed using M sets of N regularly spaced (uniformly spaced) samples. This partial information about $\hat{F}$ (measurements) is collected as $Y_k =< \hat{F}, \Phi_k >, k = 1, 2, \ldots, M$ [42, 75, 186]:

$$Y_{M\times 1} = \Phi_{M\times N}\hat{F}_{N\times 1} = \Phi_{M\times N}\Psi_{N\times N}F_{N\times 1} = C_{M\times N}F_{N\times 1} \tag{2.30}$$

That is, the signal is represented as a linear combination of basis functions of a linear integral transform. We simply correlate the object we wish to acquire with the waveforms Φ, which is the measurement or sampling operator, and Ψ is the sparcifying operator (Unitary transform), which uses cosines and sines as basis functions (i.e. $\Psi_k = \exp(i\omega_k) = \cos(\omega_k) + i\sin(\omega_k), k = 1, 2, \ldots, M$) [11, 39, 75]. In fact, there is no formal difference between Φ and Ψ, which can be combined into the global Sensing basis C. In theory, the former refers to the dictionary of physical spectra (Sensing basis) and the later refers to the dictionary of waveforms (sparsity basis). In practice, C is a partial Fourier matrix obtained by selecting M rows (i.e. measurements) uniformly at random, using Gaussian distribution, and then renormalising the columns so that they are unit-normed (For further details, see Experimental results in images in Section 9.5 and for bibliography see for example in [42, 132, 170, 186, 215]).

Note that the sampling procedure described above can be applied to any orthonormal basis, or more generally given that the columns of C are approximately orthogonal. More generally, the problem can be formed as previously for any set of functions $\Psi(t) = H(t)$ where $H(t)$ represents a Unitary transform (real or complex invertible basis). In practice we can also assume that the collections of samples is achieved with or without a small error, which will not affect considerably the recovery scheme [42, 44, 76]. On the other hand efficient sampling is guaranteed by low coherence between the sparsity system or sparsity basis Φ

and the sensing system or measurement basis Φ. This is achieved by the mutual coherence property [39, 41, 42, 44]:

$$\mu(\Psi, \Phi) = \sqrt{N} \max_{1 \leq i,j \leq N} |\langle \Phi_i, \Psi_j \rangle| \,, \tag{2.31}$$

This result demonstrates how the relationship between the sensing modality (Φ) and the sparsity model (Ψ) affects the number of measurements (M) required to reconstruct a sparse vector (signal) of length N. This serves as a rough characterisation of the degree of similarity between the sparsity and measurement systems. For μ close to its minimum value of 1, each of the measurement vectors (rows of Φ) must be spread-out in the Ψ domain. This relationship is often referred to bibliography as mutual coherence μ. In CS theory, μ as a measure of the largest correlation between any two elements of Φ and Φ implies incoherence. Incoherence means that no element belonging to one basis can be sparsily represented in terms of the other basis [82, 85]. In particular, when the Ψ domain is the Fourier basis (IDFT: $1/N \exp(2\pi i \omega t/N)$) then the sampling scheme in CS should obey $\mu(\Phi, \Psi) = 1$, which means maximal incoherence in any dimension [39, 46]. In case the Ψ domain is the wavelets basis then $\mu(\Phi, \Psi) = \sqrt{2}$ for Haar Wavelets and $\mu(\Phi, \Psi) \in [2.2, 2.9]$ for Daubechies $D4$ and $D8$ wavelets (mainly for image data) [41, 46]. Finally, if the Sensing matrix Φ is drawn from uniform independent random entries (e.g. Gaussian or ± 1 binary entries) then it exhibits high incoherence with any fixed basis Ψ, with high probability $\mu(\Phi, \Psi) = \sqrt{2 \log N}$ [46, 85].

Obviously, the pursue of the best transform domain which leads to the sparsest representation highly depends on the trade-off between the computation time and the size of the dictionary basis [132, 186]. In fact, it has been proven that we can build 2^N different orthonormal bases from a wave-like dictionary [131, 132]. The optimal dictionary can be found through a global minimisation of all vector components (the basis which minimises the entropy of dictionary[9]). This is also ideal in cases where we want to distinguish vector (image) components with same frequency range but different time-frequency structure [131, 132]. The global entropy minimisation problem produces the best match of basis when the image

[9]In general the problem of minimising the entropy of a dictionary can be defined as $H_a = 1/(1-a) \log \sum_{i=1}^{N} 1/N \; W(\Psi_i - c)^a$, where a represents the order of the entropy (for $a = 2$ we have quadratic entropy), $\Psi_i, \; i = 1, 2, \ldots, N$ is the vector column of dictionary or basis Ψ and $c > 0$ is a small constant representing the position where the window function (called Pazzen estimator) $W(\Psi_i - c)$ is centred. This function/estimator is defined as $W(\Psi_i - c) = 1/(\sqrt{\pi\sigma})^d \exp(-(\Psi_i - c)^a/\sigma^a)$ with $d, \sigma > 0$ small numbers as parameters to the estimator. For further details see [107].

or signal is not stationary. Otherwise for highly non-stationary image/signal structures a Matching Pursuit method can be particularly useful though greedy [72, 131]. In this thesis we will mainly use DFT as a more realistic application for sparse compression and recovery, though in Section 9.4.9 other Unitary transforms, such as Discrete Cosine transform, will also be used for comparison purposes. However, even in these cases efficient recovery still requires a unique sparse solution and incoherence for the collection of the under-sampled measurements; vital concepts of the CS framework.

As we have seen the conversion between the spatial (or sometimes time) domain and the wave number representation (or frequency domain) is well-known as Fourier transform. However, this is an example of a more general class of operations, which are called Unitary transforms and include Fourier, Z, Discrete Cosine, Wavelet and Laplace transforms [10]. In fact, Fourier analysis has some very strong links and properties in common with all the aforementioned transforms. A potential future direction of this research can be the examination of the performance of different sparse recovery methods together with different Unitary transforms for compressing and decompressing signals or images of interest. The most common transforms for time-frequency signal decomposition applied in the area of signal (image) processing are briefly discussed in the Appendix B. A more thorough analysis and study of Unitary transforms, their properties and applications can be found at [99, 114, 132, 136, 186, 205, 210].

2.4 Probability models

2.4.1 Randomness in Compressive Sampling

The success of Compressive Sampling lies on the design of the sampling matrix C which is responsible for the compression and recovery of the original vector. This is a very important condition for CS; which requires the sampling matrix, used for the collection of the samples, to be completely random (see later in Section 4.4). For example, if C matrix is a properly

[10] Such transforms are defined by complete sets of exponential (complex or real) functions that are periodical and have many useful properties for studying and processing signals or images (e.g. the inner product between two vectors before the transform is equal to their inner product after the transform). Indeed, a linear or non-linear combination of these functions can be used to represent almost any type of signal and the behaviour of a linear time invariant system in general [99, 114]. They are an important part of image/signal analysis, medical imaging, speech processing, error correction, biomedical engineering and even seismology!

normalised Gaussian matrix with i.i.d entries, or a normalised binary array with i.i.d. entries taking values ± 1 with probability 1/2, then the necessary conditions for many optimisation principles hold for efficient recovery using the l_1 or re-weighted l_2 norms [40, 41, 43]. This is an interesting aspect based on UUP and RIP properties, which will be explored in more details later in Chapter 4 and particularly in Sections 4.3, 4.4 and 4.5. At the moment we can say that RIP/UUP impose the necessary conditions based on the random property of the sampling matrix, so as to be able to recover any type of vector which is sparse enough given that the small number of measurements has been collected randomly. This term of "randomness" is the milestone of the sampling process in the CS theory.

In general, most physical processes in nature involve some sort of a random and stochastic property or process, which can be defined as a statistical phenomenon usually evolved in time based on probabilistic laws [57, 128, 135]. At this point, it is important to define the interconnecting terms of "stochastic" and "random". In general, the term "randomness" refers to a process capable of generating an arbitrary long sequence of outcomes which is expected to be unpredictable and shows no regularity or pattern [120, 135, 147]. A stochastic process can be described as a collection of random variables usually ordered in time and defined at a specific set of time points, which can be either continuous or discrete [57, 120]. On the other hand, mathematically speaking, a random process can be described as a sequence of random variables which are mutually independent and identically distributed [57, 120]. Usually, random variables are normally distributed with zero mean $\mu = 0$ and constant variance σ^2 (e.g. $\sigma^2 = 1$) [57, 120, 128]. As we will see in Sections 5.1, 5.2 a very common purely random process is the noise in sampling signals and images.

The evolution of a random process can be effectively defined and described using a random model. Generally speaking, random models are used in situations where the outcomes have to be random. In essence, all the possible outcomes of a situation that corresponds to a sampling process has to be random in CS. This is achieved conveniently and efficiently using a desired probability density function (pdf). It is important to pinpoint that this section does not provide an extensive coverage of all the necessary material on how to describe and obtain independent random numbers, but it is rather a quick summary of the essential concepts in a form that can be easily understood. For a more detailed treatment on this area the reader can see [57, 120, 140, 192]. Before now introducing how to generate uniformly and normally distributed random numbers as elements in the sampling matrix C, we briefly

need to describe the probability distributions which are highly applicable in CS theory for collecting random measurements:

- Random normalised Gaussian matrices [39, 42]:

$$C_{ij} \sim \mathcal{N}(0, 1/M) \quad \text{or} \quad c_{ij} \sim \mathcal{N}(0, 1) \tag{2.32}$$

- Random normalised Bernoulli or Rademacher matrices [39]:

$$C_{ij} \sim \begin{cases} +1/\sqrt{M} & \text{with probability } 1/2 \\ -1/\sqrt{M} & \text{with probability } 1/2. \end{cases} \tag{2.33}$$

- Random normalised database friendly [40, 215]:

$$C_{ij} \sim \begin{cases} +\sqrt{3/M} & \text{with probability } 1/6 \\ 0 & \text{with probability } 2/3 \\ -\sqrt{3/M} & \text{with probability } 1/6. \end{cases} \tag{2.34}$$

- Random normalised ortho-projections to $\mathbb{R}^M$ of the Fourier or Wavelet Basis [42, 43, 170], which is simply choosing M rows uniformly at random from $N \times N$ orthogonal matrix DFT or Daubechies wavelets (dictionary) and then re-normalising the columns so as that they are unit-normed (based on Banach spaces, see Appendix A). Note that in all cases M represents the number of measurements or samples and C_{ij} is the i-th, j-th element of the sampling matrix C.

In CS theory we want several independent continuous random variables which are independent in essence that their individual probability distribution functions are not correlated [120, 140]. For this reason we need to consider the probabilities of all the possible combinations of these random variables, which is achieved through the use of probability distributions, such as the Normal and Bernoulli distributions, instead of joint probability distributions, for generating random matrices [57, 120, 140]. In this thesis, the term "random matrix" refers to a matrix whose random elements are created based on some property (see Sections 3.3 and 4.5 for further details). Each element of the matrix is a random variable or random quantity

that is generated using a real valued function which actually acts on elements of a sample space [120, 192].

2.4.2 Generating uniformly distributed random numbers

One straightforward and simple way to generate a random number uniformly distributed is to use the Uniform distribution (uniform random number generator). Given a variable x we can simulate that $x \sim U(a, b)$ with $a < b$ very easily [120, 208]:

$$x = a + (b - a)y, \tag{2.35}$$

assuming that $y \sim U(0, 1)$ and $a, b > 0$ are parameters (constants). Note that this is a simple transformation method for generating random numbers using the pdf of the Uniform distribution based on its properties. Several other simple approaches exist based on the nature of the simulation using exponential, scaling and gamma random variates, all based on the calculation of pdfs, but we will not provide any further details here (see [120, 208]). A similar, in principle, but yet more sophisticated technique is the famous multiplicative congruential algorithm introduced by Lehmer et al. in 1951 [120, 192]. This algorithm constitutes the backbone of many state of the art but yet complex algorithms used in nowadays. A sequence of random integers based on this algorithm can be generated using the following equation [120, 128, 192]:

$$x^{(k+1)} = (cx^{(k)} + a)(\mathrm{mod} n) \tag{2.36}$$

where c (multiplier), a (increment) and n (modulus chosen to be very large, say 10^9) are non-negative integers as parameters of the algorithm, while $x^{(k+1)}$ and $x^{(k)}$ represent the new and the current value of a random number respectively. Note also that for the first iteration of the algorithm an initial value ($x^{(0)}$) is needed, which is called seed or starting value [120, 128]. It is notable that most normally distributed random number algorithms are based on this method, though the outcome is pseudo-random as every $x^{(k+1)}$ is completely determined by c, a and n. Also this method can be slightly amended so as to take rational values, $0, 1/n, 2/n, 3/n, \ldots, (n - 1)n$ based on the input parameters. Earlier versions of Matlab programming environment also used a slight variation of this method for generating random numbers [94].

Feedback Shift Register generators constitute an alternative family of random number generators introduced by Tausworthe in 1965, with high applicability in cryptography [120]. This method initially generates a sequence of binary digits $\ell_1, \ell_2, \ldots, \ell_n$ following $\ell_i = (c_i\ell_{i-1} + c_2\ell_{i-2} + \ldots, c_n\ell_{i-n})(\text{mod}2)$, where $c_1, c_2, \ldots, c_n$ are constants either 0 or 1 and $c_n = 1$ [120]. Alternatively, only two c values are non-zero which implies $\ell_i = (\ell_{i-r} + \ell_{i-q})(\text{mod}2)$, where r, q are integers with $0 < r < q$ [120]. The case where we start from ℓ_0 bears some similarities with the previous Lehmer method. Then the sequence of binary integers $v_1, v_2, \ldots, v_k$ is created using n consecutive ℓ_i as a number in base 2, so as $v_1 = \ell_1\ell_2 \ldots \ell_n$ and $v_i = \ell_{(i-1)n+1}\ell_{(i-1)n+2} \ldots \ell_{kn}$ for $i = 2, 3, \ldots, k$ [120]. Then the i-th uniform random number is defined as $r_i = v_i/2^n$ for $i = 1, 2, \ldots, k$. Note that the maximum period of $\{\ell_i\}$ sequence is $2^n - 1$, as we have 2^n different possible states generated [120].

An alternative less direct than the previous methods, is the Uniform Acceptance-Rejection method, which is mainly used for general continuous random numbers. Let's assume we want to simulate a random variable x from a pdf $F(x)$ with finite support on an interval (a, b) and that $F(x) \leq m$, $\forall x \in (a, b)$. In this case we need to choose a function $t(x) \geq F(x)$ $\forall x \in (a, b)$ and then create $r(x) = t(x)/c$, where $c = \int_a^b t(x)dx \geq \int_a^b F(x)dx = m$ [120, 208]. Then we generate s using pdf $r(s)$, generate $\ell \sim U(0, m)$ independently of s and check whether $\ell \leq F(s)/t(s)$; if this condition holds then $x = s$, otherwise reject and follow the same steps again [120, 208]. The accepted values of x have pdf $F(x)$. As a general note, there are several techniques for generating uniformly or normally distributed variables, where every method follows a different pattern and purpose. However, all these techniques can be classified based on their theoretical and mathematical approach into some distinct categories which we briefly describe here. For a more detailed treatment the interested reader is advised to see [120, 192, 208].

2.4.3 Generating normally distributed random numbers

A simple way to generate normally distributed random numbers is to use the polar algorithm [8, 120, 192]. This algorithm generates two random numbers in every iteration and checks whether these numbers fall into a unit circle. The testing procedure is given by the following equation:

$$V^T V \quad \text{with } V = 2\lambda - 1, \tag{2.37}$$

where λ represents a vector of two randomly generated numbers between 0 and 9, while V^T represents the transpose vector of V. Now the generation of the two normally distributed numbers is given as follows [8, 120, 192]:

$$S = \sqrt{-2\log(N)/N}V. \tag{2.38}$$

This method generates a vector of two normally distributed numbers. It is notable that this algorithm though simple in implementation and exact in generating random numbers is a bit computationally expensive since it does not accept all the initial randomly generated numbers. The square root and the logarithm in the generation equation should also be considered as time consuming.

The Box and Miller method developed in 1958 is another simple but yet highly used exact transformation method for generating normal random numbers [120, 192, 208]. It is based on the idea that two i.i.d. normal random variables can be generated using a Uniform distribution for $(0, 1)$ interval ($\sim U(0, 1)$) representing coordinates in a two-dimensional plane [120, 192]. In particular, two random numbers x, y can be generated so as to follow the Normal distribution ($x, y \sim \mathcal{N}(0, 1)$) if we obtain them as $x = \sqrt{-2\ln c_1}\cos(2\pi c_2)$ and $y = \sqrt{-2\ln c_1}\sin(2\pi c_2)$, provided that $c_1, c_2 \sim \mathcal{U}(0, 1)$ are random numbers drawn from the Uniform distribution [120, 192]. Note that the computations of cosine and sine can be usually calculated in one step, while several computationally efficient variations of this method exist [192]. For example, some methods use $c_1 \sim U(0, 2\pi)$ and $c_2 \sim \exp(1/2)$ as two independent uniform variables [208].

A more sophisticated algorithm is based on George Marsaglia, professor at Florida State University, and Wai Wan Tsang, professor at the University of Hong Kong; authors of several books in the area of random number generators. Their method is called Ziggurat algorithm; named after the famous ancient Mesopotamian terraced temple mounds [133, 192]. Developed in the early 1980's, this is a fast yet complicated method for generating a random variable from a given probability density function, for example normal and exponential. It is based on covering the target density function with a set of horizontal equal area rectangles, called "cap" and "tail". A uniform point (x, y) from the randomly chosen rectangle provides the required random variable using two tables, one for keeping integers (K) and one for keeping real values (W). In a highly abstract way the required value x is produced by

generating a random 32-bit integer j with i be the index formed from the rightmost 8 bits of j. If $j < k_i$ (i.e. rejection method to check whether the value falls into the area of the selected rectangle) then return $x = j \times w_i$. It is noteworthy that this method is not based on the Lehmer's algorithm since it does not use any multiplications or divisions between tables. Further details can be found in [133] where the Ziggurat method is extensively described and compared with other methods for normal variables and different table sizes.

Another similar well-known sophisticated class of exact random number generator was firstly introduced by Forsythe in 1972 and later extended by Brent in 1974 [31, 192]. Based on the ideas of John von Neumann, this method, called GRAND, can produce random numbers from different distributions whose pdf follows the form $F(x) = Ke^{-G(x)}$, over a range of $[a, b)$, where $a \leq x < b$, $0 \leq G(x) \leq 1$ and $K > 0$ is a small constant [192]. Based on K value a sequence of random numbers is generated using the Uniform distribution $U[0, 1)$, apart from the initial value which is generated based on initial ranges ($U[a, b)$). Then every value is compared with the previous one; if it is greater it is accepted, while if it is smaller it is rejected; process which continues till some random value is accepted and thus returned as a sample [192]. More details about acceptance regions, boundaries of distributions supported and the production of random values within a range can be found in [31, 192].

Alternative approaches for generating Gaussian random numbers is the Piecewise Linear approximation using Triangular distributions and more general rejection methods. In piecewise linear approximation we divide the Gaussian distribution into a set of K basic triangular distributions $t_1, t_2, \ldots, t_K$, each having the same width $2g$ with centre $g((K + 1)/2 - i)$ and probability p_i [120, 192]. In this case the samples are generated using multiplications and additions probabilistically using the triangular distributions. On the other hand, rejection methods generate random samples using a finite set of points taken uniformly from a finite area defined by a curve using a probability density function [120, 192]. Representative examples in this area are the Polar Rejection and its slight variation Box and Miller method. Note that some methods in these categories suffer from correlation problems between the samples generated and high computational time for the generation of the samples. For further details on these methods and their variations see [120, 163, 192].

2.5 Sparsity, incoherence and compression

2.5.1 Sparsity as compression

The search of a sparse approximation of a vector constitutes an important mathematical problem with a number of practical applications. In general, sparse approximation reduces the amount of space or time needed to store, transmit, analyse or compute huge amounts of high dimensional data, usually image or signal data [186, 215]. Due to this high importance, the area of sparse approximation has been used in many areas of science and technology, such as inpainting, de-noising, feature extraction and gene micro array analysis [132, 186]. Sparse matrices (with only a few non-zero elements) have also widely been used in scientific computation, particularly in large-scale optimisation, structural and circuit analysis, computational fluid dynamics and in general for the numerical solution of partial differential equations [132, 186, 215]. Sparsity also helps in less arithmetic calculations, less complexity in matrix algorithms, less memory requirements and certainly quickly computed results. In recent years, the need for sparse approximations in signal and image processing has been increased attracting much interest and research as an efficient pre-processing step in image and signal analysis (store, transmit and process). Several effective packages written in Fortran, C and Matlab for solving sparse linear systems are available [100]. In this thesis, however, we will only discuss the most well-known ones implemented in Matlab, which is a well-known commercial package most widely used.

Sparse representation of a vector (signal or image) in a particular domain, usually in the frequency domain, is known to enhance compression, restoration and feature extraction due to its attractive theoretical and practical properties [132, 186]. Recently in a number of papers, particularly in [11, 39, 79, 87], researchers have posed an interesting question stating that since a sparse representation of a vector concentrates all the vector's values in only a few entries, it would be beneficial to introduce a novel method for both sampling and compressing a vector at the same time, without keeping all its entries, since some of them are zero, or close to zero in a sparse form. In particular, CS theory assumes that a vector is sparse or compressible in such a way that can be sparsely or concisely represented using a proper basis or a dictionary Ψ (sparse approximation using dictionaries as a domain of representation). Let Ψ denote a dictionary as an orthonormal basis or the standard basis in the traditional

coordinate system. This means that $\Psi \in \mathbb{R}^N$ (or alternatively $\Psi \in \mathbb{C}^N$) is a set of real-valued (or complex valued) vectors Ψ_i of dimension N such that for any pair of these vectors we have $\Psi_i \perp \Psi_j$ with $i \neq j$. In a standard basis (traditional coordinate system) we have a N dimensions vector $\Psi_i = [\Psi_{ij}]$ for $i = 1, \ldots, N$ and $\Psi_{ij} = 1$ iff $i = j$. Note that any given basis can be transformed into the standard one using a Unitary transform. A signal F in a finite dimensional subspace of $\mathbb{R}^N$ (or $\mathbb{C}^N$) can be represented as a vector $F = [F_1, F_2, \ldots, F_N]$, where N is the number of terms or the length of the signal. Using a dictionary Ψ a signal is transformed into a vector of coefficients $W(F)$ as an inner product between F and vectors in Ψ. In fact, any signal F can be defined in a vector format by taking the inner product between the coefficients vector and basis vector as $F_i = \langle W(F)_i, \Psi_i \rangle$, for $i = 1, ..., N$, which can be rewritten by orthogonality of Ψ [11] as:

$$W_i(F) = \langle F, \Psi_i^{-1} \rangle, \quad i = 1, 2, \ldots, N \tag{2.39}$$

or more precisely as:

$$W(F) = \sum_{i=1}^{N} F_i \Psi_i^{-1}, \quad \text{for} \quad i = 1, \ldots, N \tag{2.40}$$

A signal $F \in \mathbb{R}^N$ is said to be K-sparse, of order/cardinality K or has K degrees of freedom if F has less or equal than K non-zero elements, entries or coefficients which are enough for its reconstruction in a domain of representation. To generalise to vectors in CS theory, a vector is said to be compressible or sparse with respect to some dictionary Ψ, when it is well approximated using K terms, as its information is concentrated in only those small in number non-zero coefficients, without knowing their position or their values (only their number).

It is prudent to note that a compressive vector means that it is not exactly sparse but most of its entries are small, close to zero (but not zero) and thus safe to ignore (i.e. its magnitude or coefficients decay following the power law). Now in sparse approximation we seek to represent F in a sparse way (only a few coefficients) by using a proper representation

[11] Due to unitarity of the transform matrix we have $\Psi\Psi^* = \Psi^*\Psi = I$, with Ψ^* the Hermitian or Conjugate Transpose of Ψ. Also, due to orthogonality of the transform matrix (column-row vectors of Ψ are perpendicular to each other) we have $\Psi\Psi^T = \Psi^T\Psi = I$. Therefore, $\Psi^* = \Psi^T = \Psi^{-1}$ and thus we have $F_i = W(F)_i \Psi_i$, or $W(F)_i = \Psi_i^{-1} F_i = \Psi_i^{-1} \Psi_i W(F)_i$.

or dictionary Ψ. More formally, the K sparse approximation of F seeks a vector:

$$\hat{F} = \sum_{i \in K} W_i(F)\Psi_i, \tag{2.41}$$

for some set $|K| = K \ll N$ coefficients. Sometimes, $\hat{F}$ is also written as F_K, which represents the K largest non-zero entries of F. The aim of CS is very similar. Given a K sparse vector F can be approximated in a basis Ψ we seek the sparsest solution $W(F) = F\Psi^{-1}$, by minimising $\|W(F)\|_{l_0} = K$, with $K \ll N$ and $\|.\|_{l_0}$ is the l_0 norm, counting the non-zero entries of vector $W(F)$. That is to recover a vector from only a few measurements drawn from the coefficients which are carefully chosen. This is achieved by expressing a vector in terms of a linear basis where most coefficients are small enough or zero so as they can be safely ignored without significant distortion. Clearly this is a good approximation of the original vector with small recovery error, but not an exact match. Using Parsevals equality, the problem is equivalent as:

$$\min_{\hat{F}} \|\hat{F} - F\|_{l_2} = \sum_{i=1}^{N} \sqrt{(\hat{F}_i - F_i)^2}, \tag{2.42}$$

for the K largest coefficients of F using Ψ for the representation. This is the error of approximation using the Euclidean distance between the approximated $\hat{F}$ and the original F vector (signal). Since the signal is sparse and the basis orthonormal by the Parseval's equality (since $\Psi^* = \Psi^{-1}$) we have:

$$\min_{\hat{W}(F)} \|\hat{W}(F) - W(F)\|_{l_2} = \sum_{i=1}^{N} \sqrt{(\hat{W}_i(F) - W_i(F))^2}, \tag{2.43}$$

where $\hat{W}(F)$ and $W(F)$ is the approximated and the original coefficient respectively. Note that for simplicity reason, in most cases we will assume the sparsity basis is orthonormal. Also, note that the minimisation with respect to either F or $W(F)$ is also referred to in the literature as Analysis versus Synthesis approach. As a consequence, the sparse recovery as an optimisation problem (defined in Section 3.4) can be viewed either as a minimisation along the coefficients $\hat{W}(F)$ (Synthesis approach) or along the signal's values directly $\hat{F}$ (Analysis approach) [51, 132]. If dictionary Ψ is an orthonormal basis, then the Analysis and Synthesis

approaches are equivalent as a minimisation criterion for sparse recovery (Ψ^{-1} exists), which is not the case in over-complete representations [51].

In real life applications we usually have a signal (e.g. image data) which can be represented as a sequence of sinusoidal waveforms but it is not usually sparse. The best way to express and store every type of signal is in terms of a linear basis (a linear combination of coefficients) using a dictionary Ψ which sparsifies it. Examples of such efficient dictionaries (discussed in Section 2.3), are the Discrete Fourier transform, the Wavelets, Discrete Cosine transform etc. The efficiency of this technique lies on the fact that signals are represented as a sum of products between sinusoids and coefficients which makes them sparse. In CS we are interested in recovering the sparsest representation given the dictionary and not in choosing the best possible dictionary, say Ψ, so as to enhance sparsity. Obviously the best transform is the one that leads to the sparsest representation which is highly based on the nature of the signal. For this purpose several variations of transforms have been introduced by appropriate rotation and shift of the basis so as to enhance sparsity based on the signal's structure. A rigorous and detailed treatment on sparse representations and CS theory can be found in [85, 132, 186, 196, 215].

2.5.2 Incoherent measurements

The design of the sampling process is very important in CS theory so as to recover sufficiently which set of coefficients we need to select in the measurement process. In essence, incoherent or pseudo-random measurements have to be chosen in such a way so as to extract the maximum amount of information from the vector using the minimum amount of measurements. Known results in the recent literature indicate that there exists a single measurement matrix such that any compressible vector (signal or image) can be reconstructed from a small number of measurements, with error at most times the worst case error for the class of such vectors [39, 48, 79]. One successful approach is to only consider matrices that can be written as a linear combination of vectors from a dictionary say Ψ, an important aspect of sparse approximation, as we need to find the K largest coefficients from $W(F)$ and measurements need to help us to estimate them. Our goal is to use a transform matrix which compress the signal F to a fewer entries so as to enhance the sampling process.

A straightforward and successful approach introduced by Candès, Donoho and Tao [42, 46, 75] is to randomly choose M rows from the $N \times N$ orthogonal matrix Ψ of the signal and

normalise the columns. An equivalent approach is to randomly choose N unit vectors in M-dimensional space by forming a matrix with random Gaussian entries (between 0 and 1) and organise them into a $M \times N$ matrix (i.e. M measurements of N terms). Then the sampling matrix C is treated as an $M \times N$ matrix with each row representing a sample to collect. Then given a measurements vector $CW(F)$ we are able to find an approximate representation for F. As a sparse approximation principle, the sampling matrix C can be viewed as a transformation, $C = T\Psi$, for some transform matrix T and thus the measurements are collected as $CF = T(\Psi W(F)) = TW(F)$. Then the non-adaptiveness is achieved by finding the K largest coefficients from $W(F)$ using C. Note that although CS theory is a relatively new area of research, it is very active with a voluminous number of publications due to the fact it combines sophisticated mathematics (Linear Algebra, Harmonic Analysis, Uncertainty Principles) with a number of applications (image processing, coding theory, sensor networks, computational biology, geophysics) [132, 186, 215].

2.5.3 A puzzling numerical problem

In many areas of practical interest we need to reconstruct an object, such as a signal or an image from its samples in the frequency domain (e.g. Fourier samples in MRI images). If we already know all the sampled values F represented in a dictionary Ψ, then we can easily (noise free case) reconstruct the object $W(F)$ in it's original domain, based on the conventional recovery methods with a number of applications, using Equation (2.39) (i.e. by inverting Ψ). Here we will adopt a different approach of more scientific interest. Let's assume we want to recover an object in the Fourier (frequency) domain Ψ, without knowing all the Fourier coefficients $\hat{F}(\omega)$. This means that inverse Fourier transform formulae in (2.19) cannot be used as we do not know all the frequencies $\omega \in Z_N$. The problem is recover a discrete time signal $F(t)$ from a partial set of Fourier coefficients $\omega = (\omega_1, \ldots, \omega_M)$ (selected uniformly at random) which belong to some set Ω with ranges over $\{0, 1, \ldots, N-1\}$ of cardinality much less than N; the size of the signal:

$$\hat{F}(\omega) = \sum_{t \in Z_N} F(t) \times e^{(-2\pi i(\omega_1 t_1 + \ldots, + \omega_M t_M))/N}, \tag{2.44}$$

where time $t = (t_1, \ldots, t_M) \subseteq Z_N := \{1, 2, \ldots, N\}$, with measurements $M << N$. Suppose now that $F(t)$ is band-limited, which means that it is supported on a small but a priori

unknown subset K of Z_N in Fourier dictionary ($\Psi = \exp(-2\pi(\omega_1 t_1 + \ldots + \omega_M t_M)/N)$) and thus it can be approximated by only K terms (all but K terms are zero or close to zero to safely ignore). Also assume that the signal can be uniquely determined in a transform domain Ψ (sparsily expressed) as a linear combination of coefficients $W(F_i(t)) = \hat{F}_i(t)$ (represented in its original domain) and the Fourier basis Ψ (uniformly spaced samples):

$$\hat{F}(\omega) = \sum_{i \in K} W(F_i(t))\Psi, \tag{2.45}$$

In contrast traditional methods for signal reconstruction, according to CS theory it is possible to recover the signal $\hat{F}_i(t)$ even without enough frequencies. For examples, a K-sparse $F(t)$ signal can be efficiently reconstructed from only $2K$ random (e.g. Fourier) measurements ω_k, $k = 0, 1, 2, \ldots, 2K$ for any $2K$ frequencies. In, particular for the case N is prime, the signal $F(t)$ can be almost exactly recovered from this small set of observations (frequencies) by solving a straightforward optimisation problem which finds $\hat{F}(t)$, a good (sparse) approximation of $F(t)$ based on the Theorem:

Theorem 2:[39, 42, 75] *It is possible to recover a signal $F(t)$ from its partial Fourier $\hat{F}(\omega)$, provided that its length N is a prime number, its support (sparsity level) $K \leq 1/2|\Omega|$ and $|\sup \hat{F}(\omega)| + |\sup \hat{F}(t)| \geq 2\sqrt{N}$.*

Also in [42, 75] it has been shown experimentally that successful recovery rate is more than 50% for support $K \leq |\Omega|/4$, while for $K \leq |\Omega|/8$ more than 90%, given that $|\Omega| \leq N/4$, $M \geq 2$ and $N \geq 20$. Counter examples where Theorem 2 does not work for arbitrary small sample sizes are also discussed by the same authors. Further details about this problem and its assumptions will be given in Chapter 4. Notice, however that the derived signal $\hat{F}$ is simply an approximation of the original $F(t)$ and can never be exactly the same for two reasons; due to the approximation moving from the time or spatial domain to the frequency or transform domain and also due to the fact that Ω set is partially known through the collected samples.

2.5.4 Key aspects of the CS method in a nutshell

At this point it is crucial to briefly summarise the key aspects and assumptions of CS theory:

- It is possible to reconstruct a signal from incomplete frequency samples (i.e. under-sampled measurements) which are less than those defined as necessary in the Nyquist sampling rule, by solving a non-linear (l_0 norm) optimisation problem.
- For example, if we have a N-term discrete time signal $F(t)$ in a domain ($\mathbb{C}^N$ or $\mathbb{R}^N$) and a randomly chosen set of frequencies $\omega \in \Omega$. Then, it is possible to reconstruct it from partial knowledge of its coefficients (such as its Fourier coefficients) on the set of Ω.
- Signals can be stored in a compressed form (sparse signals) by expressing them in terms of linear basis or more precisely as a linear combination of elementary parts of signals called atoms in the frequency domain. By this way zero or very close to zero coefficients of these signals can be safely ignored without much distortion or in other words with very low recovery error as we have seen.
- Usually a Unitary transform (such as DFT, DCT, Wavelets) is used so as to enhance the sparsity and compressibility of the signal, where the largest coefficients are kept and those close to zero are discarded.
- The CS method can also be applied in signals or images that are almost sparse (e.g. compressible signals) and also in cases where the samples were collected in a noisy environment (small additive random noise), as we will see in a Chapter 5.

Chapter 3

The CS method

Before we define the sparse recovery (approximation) problem as an optimisation problem we firstly need to briefly describe how the Compressive Sampling (CS) technique works in a vector or function (a simple signal). The key steps of CS method are to initially compress and simultaneously sample vector (lossy compression) and then follow the recovery process (decompression) as an optimisation problem. CS constitutes a new way of vector acquisition and recovery by using only a small amount of linear, suitable and non-adaptive measurements, the precise number of which is pertinent to the level of sparsity of the original vector in a given dictionary. An important note is that all the efficient measurement matrices suggested in bibliography follow the same pattern of independent identically distributed uniform values, drawn from a specific probability density function. In all what follows, we will adopt a simple example to present all the main aspects of CS method, including all the necessary assumptions and key steps before defining the sparse recovery problem.

3.1 An example of sparse signal representation

Let's assume a simple, real valued, discrete time, one dimensional signal represented as a function (vector) $F(t) \in \mathbb{R}^N$. We aim to compress it and then recover it by finding it's best sparse approximation which fits the measurement data collected. We also assume that $F(t)$ is of size N (N element vector) but it is supported on a set of size K (K sparse) in a transform domain Ψ. For example the signal can be expanded (represented) as a superposition (linear combination) of spikes (such as canonical form in $\mathbb{R}^N$ using the delta function as basis $\delta(t-i)$),

sinusoids (Discrete Cosine or Sine transforms), Fourier (complex basis), Wavelets, etc. (use of a Unitary transform). This means that $F(t)$ can be approximated in an orthonormal basis $\Psi \in \mathbb{R}^{N\times N}$ as a linear combination in a basis:

$$F(t) = W(F)\Psi = \sum_{t=1}^{N} W_t(F)\Psi_t, \quad \text{for} \quad t = 1, \ldots, N, \tag{3.1}$$

where $W_t(F)$ is the t-th term of the signal's coefficient. Note that $W(F)$ and $F(t)$ represent the same signal, the former represents the signal in its original (time) domain, while the latter represents the same signal in the $\Psi(t)$ domain. Both $W(F)$ and $F(t)$ can also be represented in a vector format:

$$W(F) = [W_{t_1}(F), W_{t_2}(F), W_{t_3}(F), \ldots, W_{t_N}(F)] \tag{3.2}$$

$$F(t) = [F_{t_1}, F_{t_2}, F_{t_3}, \ldots, F_{t_N}] \tag{3.3}$$

Furthermore, note that Ψ is a diagonal matrix with non-zero elements only in its main diagonal and therefore in (3.1) we assume Ψ_t as the t-th element of its main diagonal. In the simple case where Ψ is the canonical basis, its elements are defined as:

$$\Psi_{i,j}(t) = \begin{cases} 1, \text{ if } i = j \\ 0 \text{ otherwise} \end{cases} \tag{3.4}$$

Alternative approaches could be that Ψ is the cosine basis and/or sine basis depending on the transform we apply to the signal (Fourier, Wavelet, etc.). In certain cases in the Experiments Chapter 9 we will explicitly define different bases for Ψ as a sparse approximation for discussion and comparison purposes. Note that we assume that a signal $F(t)$ is sparse in the Ψ domain if its support is very small or concentrated on a small set (compressible), which means $\sup W(F) = N \gg \sup F(t) = K$, where N is the length of signal and sup is the set theoretic support which simply counts the non-zero entries (similarly to l_0 norm). This linear combination between the coefficients and the basis is important for the signal in order to be expressed in a vector format and look sparse under a specific (canonical or not) basis. Although most real-world signals are not sparse, we can choose an appropriate transform

(like Fourier transform) in order to achieve sparsity under a different (appropriate) basis. This is important since the sparse representation efficiently captures the structure inherent to the signal we analyse and compress.

3.2 Partial measurements

Our goal in the sampling process is to collect M measurements which are close to sparsity level K of the signal $F(t)$ for low distortion in the recovery and thus efficient compression during sampling acquisition. Expressed in a different way we want to collect M incoherent (random) measurements as partial information about signal $F(t)$. In CS theory a simple and straightforward way to achieve that is to choose M randomly subsets (elements) of the original signal vector $F(t)$ of N elements. For this purpose we have to generate a random vector $\Phi \in \mathbb{R}^{M\times N}$ with values between 0 and 1, following the Gaussian distribution. Every row of Φ represents a measurement or sample and thus the sampling process is simply a vector multiplication between Φ and $F(t)$. In particular, for the collection of the k-th sample of vector $Y \in \mathbb{R}^M$ we have:

$$Y_k = \langle \Phi_k, F(t) \rangle, \quad k = 1, 2, \ldots, M \tag{3.5}$$

where M represents the number of measurements with $M \ll N$, Y_k is the k-th element of samples vector Y and Φ_k is the k-th row (vector taken from Φ), since the M rows of Φ represent the M measurements. For example, the second row of Φ represents the vector Φ_2 which is the measurement taken for the second sample. Now in vector format the measurements acquisition process can be represented as:

$$Y_{M\times 1} = \Phi_{M\times N} F(t)_{N\times 1}, \tag{3.6}$$

or using the coefficients $W(F)$ and the basis Ψ,

$$Y_{M\times 1} = \Phi_{M\times N} \Psi_{N\times N} W(F)_{N\times 1} \tag{3.7}$$

At this point it is important to mention that we assume the sampling process here is noiseless, or more precisely we assume a noiseless environment for the collection of measurements. However, the CS method also works in a noisy environment with some slight variations

which is discussed in Chapter 5.

In this setup, once the measurements have been collected, the compressed signal can be transmitted, processed or stored for further analysis. A simple scenario could be that the encoder or sender sends the collected (compressed) samples and the Sensing matrix (compressing and sparsifying bases matrices) to the decoder or receiver for the reconstruction.

A more realistic scenario is the fundamental problem in communications, where we want to recover or correct a signal sent from one point to another [176]. A discrete time signal can be a message or sequence of symbols or functions (telegraphy) while a continuous signal can be a continuous sequence of symbols (radio or TV) or even a combination of these two types (transmission of speech) [12] [176]. This reconstruction system will aim to select the best possible solution based on various engineering parameters, such as time, bandwidth, capacity of channel, reliability or even complexity in operations. Again, the information source is the sender from the one end of communication channel, sampling, compressing quantising, encoding and transmitting signals (e.g. climate, military, satellite, seismic data) and the receiver from the other end which performs the inverse operation (correcting, decompressing or reconstructing the distorted signal) [58, 176].

At this point it is important to note that in some cases we may not have the original vector or basis but only the partial measurements expressed as a superposition of a sequence of coefficients in a known dictionary. CS method will also work in this case with the difference that we cannot verify the whole process in terms of recovery error between the original vector and it's best sparse approximation. However, as we will see in Chapter 4 there are certain measurement bounds and properties for the stability of the derived solution obtained from the under-sampled measurements. In general CS theory adopts a different procedure for both signal acquisition and (sparse) recovery in contrast to conventional approaches, dictated by the Nyquist-Shannon theorem of signal reconstruction, which requires uniform sampling at a very high rate and linear recovery by interpolation. CS can recover a signal from highly incomplete and inaccurate measurements drawn randomly from a signal with a low yet arbitrary and unknown support of small size. In essence, the signal is sensed

[12] Note that in this thesis we will only consider discrete type communication systems and thus only discrete time signals. Teletype and telegraphy represent discrete type channels for transmitting information. In general, in a discrete channel we have a sequence of options from a finite set of symbols, which can be transmitted from one point to another, at a certain duration [176].

and compressed simultaneously by a small number of measurements and hence the name Compressed or Compressive Sensing/Sampling of the method.

3.3 Mutual coherence

A key property about the performance of CS is the design of the sampling matrix Φ, which involves the analysis of coherence as a property. In Linear Algebra, mutual coherence is an easy to compute property which defines the absolute value of the inner product between the columns of Φ and rows of Ψ (Φ has less dimensionality than Ψ). This can be defined as [39, 41]:

$$\mu(\Phi, \Psi) = \sqrt{N} \times \max_{1 \leq i,j \leq N} \frac{|\langle \Phi_i, \Psi_j \rangle|}{\|\Phi_i\|_{l_2} \|\Psi_j\|_{l_2}} \tag{3.8}$$

and since $\|\Phi_i\|_{l_2} = \|\Psi_j\|_{l_2} = 1$ (unit columns for Φ and Ψ) we have:

$$\mu(\Phi, \Psi) = \sqrt{N} \times \max_{1 \leq i,j \leq N} |\langle \Phi_i, \Psi_j \rangle| \tag{3.9}$$

where Φ_i and Ψ_j represent the columns and rows of Φ and Ψ respectively. In fact, there is no formal difference between Φ and Ψ, which can be combined into the global Sensing basis $C = \Phi\Psi$, which maps the signal from Ψ to Φ domain (selecting M rows uniformly at random from Ψ).

The $\mu(\Phi, \Psi)$ is simply the maximum correlation between two vectors properly rescaled. It represents the largest entry (maximum) in the product between the two vectors Φ (the measurement basis or system) and Ψ (the sparsity basis or system) properly normalised. It can be seen that the values of $\mu(\Phi, \Psi)$ span in the range of $[\sqrt{\frac{N-M}{M(N-1)}}, 1]$ as a general case [85]. However, since $M \ll N$ the lower bound is approximately $1/\sqrt{M}$. In CS theory we want low coherence ($\mu(\Phi, \Psi) \approx 1$) in order to derive the necessary randomness in the measurements (the lower the coherence, the lower the number of samples needed). There are several sampling matrices Φ which can achieve maximally incoherence in terms of Ψ, such as the Gaussian Distribution with $\mathcal{N}(0, 1/\sqrt{M})$ and Rademacher distribution [13]. In fact,

[13] A Rademacher distribution generates values ± 1 with equal probability 1/2 [39, 85]. This bears some similarities with the sub-Gaussian distribution where a random variable acquires a value based on the pdf $E(e^{xt}) \leq e^{c^2 t^2/2}, \quad \forall t \in \mathbb{R}, \quad c > 0$.

$\mu(\Phi, \Psi)$ determines the necessary number of samples for efficient recovery. If the coherence is equal or very close to 1, this guarantees that the $K + 1 \leq M \leq 2K$ is a sufficient number of measurements for efficient recovery of the signal, with lower bound for the l_0 norm problem and upper bound for the l_1 problem (see details in Chapter 4). In particular, for the l_1 norm problem we want the coherence metric to be almost 1, which requires $M = 2K$ measurements or $M \approx K \log_2(N/M)$ samples [11, 41, 75]. More formally,

Theorem 3:[41, 42] *If a N-element vector F is K-sparse (K non-zero entries) in a domain or basis Ψ then we need to select M measurements uniformly at random in Ψ with $M \gtrsim \mu^2(\Phi, \Psi) K \log_2 N$.*

However, since $\mu(\Phi, \Psi) = 1$ and $M \ll N$ we need only $M \gtrsim K \log_2 N$ measurements. Notice that the dependence on N is logarithmic, but usually the number of measurements M required is approximately 4 times the sparsity level for almost all types of signals [11, 42, 46]. For example, if the length of the signal is $N = 256$, then we need only $M = 30$ data (samples) so as to perfectly recover $K = 15$ necessary coefficients. The importance of this theorem is that we do not actually need to collect more than M samples-measurements while we also cannot sample less than this required amount for efficient recovery using the l_1 norm optimisation principle. For more details on Mutual Coherence see for example [41, 132, 186, 215].

3.4 Recovery as an optimisation problem

Given now the partial information collected (Y) we want to derive the original signal values $F(t)$ or its coefficients $W(F)$ in Ψ domain. Ideally we would want to recover all the N coefficients or elements of $F(t)$, but we have only observed (sampled) a subset ($M \ll N$) randomly collected:

$$Y_{M\times 1} = \Phi_{M\times N} \Psi_{N\times N} W(F)_{N\times 1} \tag{3.10}$$

or by substituting $C = \Phi\Psi$ as a projection $\mathbb{R}^N \to \mathbb{R}^M$ (measurements matrix) we have,

$$Y_{M\times 1} = C_{M\times N} W(F)_{N\times 1}, \tag{3.11}$$

where the coefficients $W(F)$ are of size $K < M \ll N$, which is much less than the number of measurements M and the signal length N. Note that some of these coefficients are zero or small enough in Ψ domain and thus we can safely ignore them without much distortion or error in the recovery. However, this reconstruction requires the generation of all the possible values of the coefficients which satisfy the measurements (fit the data) and then the selection of the "best ones" (sparsest ones), as a subset of the original coefficients (subset of M elements out of N elements). This search is clearly computationally infeasible (For details see [39, 41, 44, 50]). Clearly we cannot solve this problem because the number of unknowns are more than the equations. Such types of problems are called ill-posed or ill-conditioned problems. Ill-posed is a problem whose solutions are not unique, do not exist or they are not stable under perturbations of data (i.e. noise) [39, 42, 46].

Researchers in [43, 79, 87] suggested that the problem of obtaining an accurate reconstruction, among all possible coefficient sequences consistent with the measurements, is equivalent to solving a non-convex optimisation problem:

$$\min_{\hat{W}(F)} \|\hat{W}(F)\|_{l_0} \ s.t. \ C\hat{W}(F) = Y \ (= CW(F)) \tag{3.12}$$

on condition that the initial sampled signal is sparse on a domain of representation and the measurements are incoherent (one consistent solution satisfying both the linear equations and the sparseness constraint). Note that this is an approximation recovery and thus the recovered coefficients $\hat{W}(F)$ (sparsest solution obtained) are not exactly the same as the original ones $W(F)$. Also $\|.\|_{l_0}$ is the strictest measure of sparsity defined as the number of non-zero coefficients ($\|\hat{W}(F)\|_{l_0} = K$). For real-valued data the system (C, Y) is linear (Linear Programming problem) while for complex-valued data the system (C, Y) is a second order cone programming problem [46, 75]. Several sparse recovery methods have been introduced for reconstructing sparse vectors based on their under-sampled measurements (see Chapters 7 and 8). However, this is a combinatorial optimisation problem whose computational complexity grows exponentially as N increases. For this reason it has been suggested in several papers (e.g. in [11, 40, 41, 75]) to relax the l_0 norm with its convex analogue, l_1 norm:

$$\min_{\hat{W}(F)} \|\hat{W}(F)\|_{l_1} \ s.t. \ C\hat{W}(F) = Y \ (= CW(F)). \tag{3.13}$$

In fact, it has been proven that the l_1 norm problem is equivalent to the l_0 norm problem if the Restricted Isometry hypothesis (RIP) is satisfied, which requires every set of columns in C with support less than K approximately behaves as an orthonormal system (i.e. $\|CW(F)\|_{l_2} \approx \|W(F)\|_{l_2}$) [39–41]. Further details about this notion can be found in Section 4.3. With this in mind, the necessary number of samples for efficient reconstruction using the l_1 norm are $M = O(K \log_2(N))$ while for the l_0 norm recovery is possible (in theory) with just $K + 1$ measurements.

As a final remark note that there are two different approaches as a minimisation criterion for l_1 and l_0 minimisation problems. The minimisation of the recovered coefficients $\hat{W}(F)$ is referred in the literature as a Synthesis approach, while the direct minimisation of the signal's values $\hat{F}$ is referred as Analysis approach [51, 132]. The latter is simply a substitution of $\hat{W}(F)$ by $\hat{F}$ in Equation (3.13), namely

$$\min_{\hat{F}} \|\hat{F}\|_{l_1} \ s.t. \ C\hat{F} = Y \ (= CF). \tag{3.14}$$

Both approaches are strongly connected based on the duality properties of their corresponding optimisation problems and they are equivalent in the case that Ψ dictionary is an orthonormal basis, but different for over-complete dictionaries [51]. In general, these two approaches yield different results depending on the design of dictionary used, regularisation function or norm as a penalty term and noise level for the collections of measurements [51]. Different efficient sparse recovery methods have been introduced for different convex and non-convex problems stemming from these two approaches. The interested reader can find further details in [51, 132].

3.5 Stability of the recovery

As discussed already in the previous Sections, the purpose of CS theory is to derive a good approximation $\hat{W}(F)$ of the original compressed signal $W(F)$ (vector), whose coefficients or amplitudes are simply a subsequence of the original ones. Since the collection of measurements is random in nature, there is a probability of failure. For M non-adaptive (random) measurements satisfying low coherence and RIP hypothesis, the signal recovery is almost

exact with probability [39, 41, 75]:

$$1 - \tau_N = 1 - e^{-\gamma N}, \text{ or } 1 - \tau_N = 1 - O(N^{-\lambda}) \tag{3.15}$$

where τ represents the probability of failure, N is the length of the signal and $0 < \gamma, \lambda < 1$ are small constants. The stability of the solution $\hat{W}(F)$ derived by solving (3.13) obeys [39, 46]:

$$\|W(F) - \hat{W}(F)\|_{l_2} \leq \epsilon \frac{\|W(F) - W_K(F)\|_{l_1}}{\sqrt{K}}, \tag{3.16}$$

where $\epsilon \leq 8.77$ is a small constant, $W_K(F)$ is the best K-sparse approximation, while $W(F)$ and $\hat{W}(F)$ represent the original and recovered coefficients of the initial signal $F(t)$. The $\|.\|_{l_2}$ norm represents the Euclidean distance between the coefficient vectors. For further details see Section 4.4.

3.6 Sampling and reconstruction of a simple signal

For a simple numerical example, consider a simple one dimensional signal $F(t) \in \mathbb{R}^N$ as a sequence of discrete time values which is sparse in time domain (its basis is the identity matrix):

$$F(t) = e^{-t^3}, \tag{3.17}$$

which is bandlimited in time $t \in \{0, 1, \ldots, N-1\}$, $(N = 30)$ (the signal's value becomes zero outside this time period). It can be easily seen that the signal represents an exponentially decaying continuous function, which means it is sparse in time domain ($K = 3$ non-zero values) as the signal becomes almost zero after time $t = 2$ (i.e. we assume that $C = \Phi$ in our case since $F(t)$ is sparse in time domain and thus $\Psi = I_N$). By measuring the value of the continuous function $F(t)$ in the given time t we can extract our samples so as to create the discrete time signal in a form of vector which will then compressed. In particular, the continuous time signal can be represented in discrete time as a vector of length $N = 30$; $F(t) = [F(t_0)F(t_1)F(t_2)\ldots F(t_{N-1})]$, where $t_0 = 0$, $t_1 = 1$, $t_2 = 2$, etc. Compressive Sampling requires on the order of $M \approx K \log_2 N = 3 \log_2 8 = 9$ random samples for efficient

recovery, in contrast to ordinary sampling techniques, dictated by the Nyquist-Shannon theorem, which will require $N = 30$ uniform (equally spaced) samples. By generating a sampling matrix $C \in \mathbb{R}^{M \times N}$ we are able to collect the incoherent measurements of the form:

$$Y_k =< F(t), C_k >, \quad k = 1, \ldots, 9; \quad t = 0, 1, \ldots, 29, \tag{3.18}$$

where C_k is a matrix of small random numbers between zero and one generated using the normal Gaussian distribution $\mathcal{N}(0, 1/3)$ and $F(t)$ is the signal in vector format, i.e. $F = [0.0498, 0.0025, 0.0001, 0.0000, \ldots, 0.0000]$, with $F(t_0) = 0.0498, F(t_1) = 0.0025, F(t_2) = 0.0001, \ldots, F(t_{29}) = 0.0000$. The under-determined system of linear equations is formed as:

$$\begin{aligned} F(t_0) * C_{10} + F(t_1) * C_{11} + F(t_2) * C_{12} + \ldots + F(t_{N-1}) * C_{1N-1} = Y_1 \\ F(t_0) * C_{20} + F(t_1) * C_{21} + F(t_2) * C_{22} + \ldots + F(t_{N-1}) * C_{2N-1} = Y_2 \\ \vdots \\ F(t_0) * C_{90} + F(t_1) * C_{91} + F(t_2) * C_{92} + \ldots + F(t_{N-1}) * C_{9N-1} = Y_9 \end{aligned} \tag{3.19}$$

The sparse recovery problem as an optimisation problem is now defined as:

$$\min_{\hat{F}(t)} \|\hat{F}(t)\|_{l_1} \; s.t. \; C\hat{F}(t) = Y \; (= CF(t)) \tag{3.20}$$

The results of the recovery of this simple signal are presented in the following Figure (3.1) and Figure (3.2). CVX package (see Chapter 8 for details) was used for the implementation of both the l_1 and l_2 optimisation principles. It can be clearly seen that the l_1 norm based optimisation principle works well as a sparse recovery approach without noise in the sampling (compressing) process. Quantitatively speaking note that the l_1 norm based optimisation principle correctly identifies all the non-zero components of the signal $F(t)$ and correctly sets all the others to zero. Clearly the conventional l_2 norm based optimisation principle which minimises the residuals, i.e. $\|C\hat{F}(t) - Y\|_{l_2}$ exhibits low quality estimate of the signal, as it tends to ignore the true value of the zero elements of the signal. In fact, the l_1-norm is a better metric than the l_2-norm since it is able to find an optimal solution closer to the coordinate axes of $\mathbb{R}^N$ where is likely to have more zeros or closer to zero values. However, note that due to the nature of the CS technique (i.e. lossy compression-sampling matrix C)

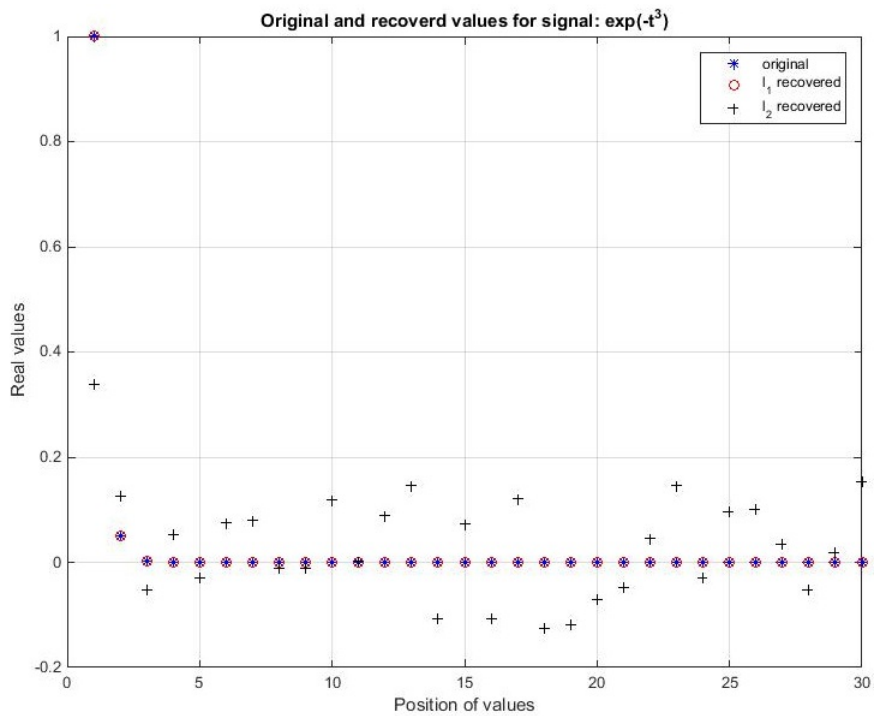

Figure 3.1: Estimation of a simple one-dimensional sparse signal $F(t) = e^{-t^3}$, consisted of 30 elements (in vector representation), with C a 9 by 30 sampling matrix with independent Normal distribution entries. The blue star indicates the original (true) values of the signal, the red circle the estimate using the l_1 norm based optimisation principle, while the black cross the estimate using the l_2 norm based optimisation principle. Observe that l_1 norm based problem efficiently recovers the signal's values for small samples size (9 samples) in contrast to the l_2 norm based problem which fails (example1.m).

the l_1 norm based estimation procedure is not exactly accurate in essence that the estimation $\hat{F}(t)$ of signal $F(t)$ from $C\hat{F}(t) = Y$ is not exact, but a good approximation with a recovery error proportional to $O(10^{-9})$ (some constant of proportionality depending on sparsity level K of $F(t)$ and the size of the measurement matrix C, see Chapter 4 for details). Note also that the necessary number of measurements (dimensions of matrix C) highly depends on the structural content (sparsity) of the signal rather than its length (number of elements).

At this point, we can briefly pinpoint the differences between CS and Traditional Sampling techniques. The key aspects of Traditional Sampling techniques:

1. A signal has a known and connected set of coefficients of size N in a transform domain.

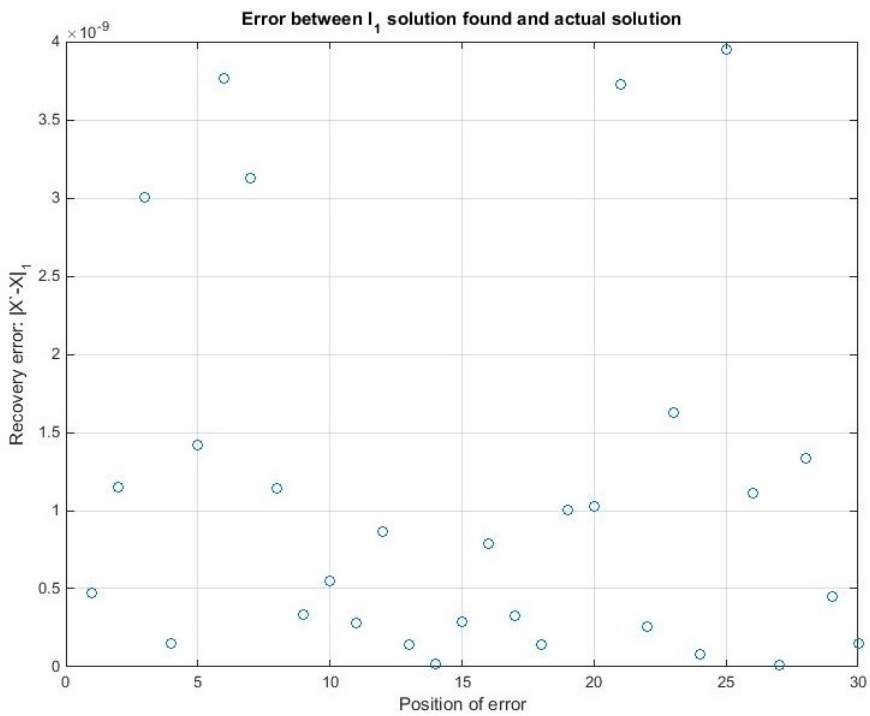

Figure 3.2: Absolute value of the recovery error $\|F(t) - \hat{F}(t)\|_{l_1}$ between a simple one-dimensional sparse signal $F(t) = e^{-t^3}$, consisted of 30 elements (in vector representation) and an estimate $\hat{F}(t)$ using the l_1 norm based optimisation principle, with C a 9 by 30 sampling matrix with independent Normal distribution entries. Observe that although the error is small, due to the efficiency of the sparse recovery approach, it is not absolutely zero due to the nature of CS technique; it is a lossy compression process of a sparse vector (example1.m).

2. Uniform sampling at high rates is required for exact recovery from N equally spaced domain specific samples.

3. This is a linear (lossless) recovery using one of the well-known interpolation methods (see Appendix B).

CS technique, on the other hand, is a non-linear sampling method:

1. A signal has an arbitrary and unknown set of coefficients of size N in a transform domain.

2. Almost exact recovery is possible, with high probability, from $M \approx K \log_2 N$ randomly chosen measurements (select M sample locations at random).

3. This is a non-linear recovery by solving an optimisation problem.

3.7 CS as a sparse image recovery method

The CS method as a recent revolutionary sparse recovery technique can be applied not only on signals but on images as well. A typical high resolution image, captured by an ordinary machine, such as a camera, can be efficiently compressed into a substantial smaller in size image without much loss. This recovery of large digital data sets is possible even in the case where the available measurements are incomplete and insufficient according to the famous Nyquist-Shannon criterion [42, 75, 170]. Straightforward and precise the Nyquist-Shannon theorem states that it is possible to reconstruct an image from its samples if the sampling interval is chosen to be in size such that it is less than half of the smallest interesting detail (pixel) in the image. Consequently, if we follow an under-sampling procedure, the image will be distorted (i.e. aliasing effect), as there will not be enough samples (pixels) for the reconstruction and thus the individual components of the image will overlap [111, 170, 182]. CS as a novel sampling paradigm allows us to go beyond this limit by adopting two important concepts: transform of an image into an appropriate basis so as to enhance its sparsity and then keep randomly only the most important coefficients. These assumptions are enough to process any type of image, compress it and then sparsily represent it, using non-adaptive linear projections (measurements) while keeping the image structure [132, 186].

3.7.1 General Image representation

In order to perform some sort of an image analysis, an image, captured by a digital camera, has to be represented by a concrete and concise form of representation [14]. Common types of image presentation include matrices, chains, pyramids, quadtrees and other relational or hierarchical data structures. In this thesis for simplicity and efficiency reasons we will adopt only the matrix representation of an image which can be of .bmp, .pgm, .tiff, .gif, or .jpeg format. This is the most well-known and traditional data structure as a simple image representation for processing [111, 182, 186].

In general, images can be represented as two dimensional arrays of points, which represent the special distribution of the irradiance at a plane and whose dimensions depend solely on the dimensions of the image [111, 182, 186]. For example, a very common 2 Megapixel (e.g. 1920×1080 [15]) digital image is represented as a matrix of the same dimensions 1920×1080, which contains 1920 samples per line and 1080 samples per column; all together 2073600 samples. A grey level image is a 2D rectilinear array of pixels or pels (i.e. abbreviation of picture elements), each one of which represents a sample from the image or a sample from a continuous function as we will see. Every pixel which is an element of this matrix is located on a rectangular grid and has a set of integer values that correspond to the position of the matrix (x, y coordinates) and the brightness value (light intensity), which is the z coordinate [111, 132, 182]. Sometimes time t of the captured element of the matrix is also another common property. In essence, a pixel represents a particular property, namely the location of irradiance or colour intensity at the corresponding grid position [111, 182, 186].

It might be easier to see every element of the matrix as a continuous function of two ($f(x, y)$) or three ($f(x, y, z)$) variables, where (x, y) represents the coordinates of a pixel in a matrix or plane and the third variable z in $f(x, y, z)$ represents time or the type of color intensity (i.e. Red, Green or Blue for color images or the light intensity of gray in gray level/scale images) [111, 132, 182]. Every set of these three or two coordinates also represents a pixel, which is the minimum information stored and thus the collection and sum of these pixels represents the sampling process (as every pixel is actually a sample itself). A more thorough analysis of a structure and representation of images can be found

[14] In essence an image analysis represents an attempt to find a relation between input image(s) and previously established models of the observed world [111, 182].

[15] Actually this is the popular typical High Definition size of images.

in [158], where the PGM format is discussed. Portable Gray Map (PGM) is the simplest form of a greyscale image file format which has been designed as an easy image format to learn and write programs for processing. General analysis and processing techniques on images can be found in [58, 111, 132, 182, 186].

There are many different types of colour used in the representation of images, such as Red-Green-Blue, Cyan-Magenta-Yellow-Black and Hue-Saturation-Intensity, with the most famous one the RGB model [111, 132, 182]. However, in all cases, despite the model used, the last or third variable (z) of each pixel represents the colour, whose values highly depend on the standard followed. The standard followed represents the number of variations of the values of each colour. For example, in the 8 bit representation we have $2^8 = 256$ values for each colour intensity (different intensities), between $0 \ldots 255$, while in the 16 bit standard (2^{16} values) the colour intensity values are $0 \ldots 65535$, which can be values of red, yellow, saturation or grey (shades of grey) for colour and grey level digital images respectively. In binary representation of images, there are only 2 different values, white (the strongest) represented by 1 and black (the weakest) represented by 0. In general, the smallest value of the light intesity is always 0 and represents black, while all the other values represent the tones or shades of grey. Note also, that a simple and straightforward way to change a colour image to a black and white or grey level/scale image, among many other ways, is to take the average of all the three parameters or levels of colour into one [111, 178, 182]. One of the most common ways used in software, such as Adobe Photoshop, is to use weights in this average namely 30% Red, 59% Green and 11% Blue.

3.7.2 Sampling images

Digitisation is an important step of an image before it can be suitable for further processing and analysis using digital computing methods. In essence, an image must be digitised spatially and in amplitude, which means we have to determine its number of pixels and its different levels of light intensity. The quantisation rate determines the number of different discrete values of the set of points used for the image representation (light intensity of pixels). The sampling rate determines the spatial resolution of the digitised image, which is the sampling rate of the image's values at a discrete set of points, represented in a vector format (number of pixels). Our starting point is to sample our image using a discrete set of sampling points in the plane. In general, the sampling process can be modelled as an

input-output system $F_s = r(F)$, where F is a single image or an image sequence observed, r is the linear or non-linear image processor such as sampling or compression and F_s is the image we derive after this processing step. If we now restrict our analysis in rectangular grids, the sampling process is a sampling function $r(x, y)$, which can be written as [111, 182]:

$$r(x, y) = [m\Delta x, n\Delta y], \quad (m, n) \in \mathbb{Z}^2, \tag{3.21}$$

where $x = m\Delta x$ and $y = n\Delta y$ are the sampled image points (pixels) based on the image dimensions, say $M \times N$, $m = 1, 2, \ldots, M$ and $n = 1, 2, \ldots, N$. The distances Δx and Δy are the sampling intervals which separate two neighboring sampling points. Then, the sampling operation is the product of the continuous image function $f(x, y)$ and the sampling function $r(x, y)$, can be represented as [111, 182]:

$$f_s(x, y) = f(x, y) \sum_{n=1}^{N} \sum_{m=1}^{M} \delta(x - m\Delta x, y - n\Delta y), \tag{3.22}$$

where $f_s(x, y)$ is the sampled image function. Usually, in ordinary sampling schemes, the sampling interval has to be chosen in size so as it is less than half of the smallest area of interest, in terms of detail, for efficient recovery. However, it is also possible to consider infinite sampling grid, which is periodic with periods Δx and Δy, in the frequency domain [182]. In this case the Fourier transform of the sampled image is the sum of periodical repeated Fourier transforms $F(u, v)$ of the image. Note that here we are not interested in images which are sparse by themselves but are sparse in a known transform domain (Unitary transform). In essence, we assume that images can be expanded on a fixed orthonormal basis (sparse representation on a transform domain), meaning that only a small number of transform coefficients are non-zero (sparsity enhancement) [132, 186, 215].

A continuous image $f(x, y)$ is normally approximated by equally spaced samples based on the desired sampling rate [111, 182]. Therefore the crucial question arises as to what the sampling rate should be so as to have no losses on the image representation. Intuitively, the complete and accurate representation of an image is determined by a strict mathematical sampling Theorem, known as Nyquist-Shannon sampling criterion. According to this Theorem, a careful sampling is necessary to reproduce the image efficiently, otherwise considerable distortions will occur in reconstruction. Since every pixel is actually a single sample which

carries light intensity information, a considerable amount of pixels has to be collected for efficient reconstruction of the image:

Theorem 4 (Nyquist-Shannon):[111, 182, 212] *It is possible to efficiently reconstruct an image if the sampling interval between the intensity measurements (pixels collected) is less than half the period of the highest frequency, called cut-off frequency f_c. A function (image) which is bandlimited in a Fourier spectrum (domain) has no sinusoidal components with frequency higher that f_c. This means that the spatial sampling rate must be no larger than $\Delta x = 1/2f_c$, so as to preserve the distance between two resolvable adjacent points in the focal plane of the sampled image (two sample points per cycle so as to preserve spacing between the centre and the edge points).*

Consequently, a sampling grid of a certain density which follows this pattern is enough for any reconstruction (interpolation) method to recover an image which has the same topological properties and shape as the original (sampled) one. Note that if $\Delta x < 1/2f_c$ then the sample spacing is larger than the one needed for efficient recovery and thus the image is said to be under-sampled. When the sampling space is too large, the pixels collected are too far in relation to the size of the image and therefore some details of the image are lost in the sampling process.

The CS method has been introduced lately as an alternative measurement system and sampling architecture in real-world implementations, namely images. It shows that it is possible to sense and compress an image through non-adaptive random measurements which do not follow the explicit knowledge of the aforementioned Nyquist-Shannon theorem. At a first glance, improper sampling appears to cause an enormous loss of information which may result to a reduction in image resolution (quality) and thus serious sampling artifacts, called aliasing phenomenon. Similar problem appears in signals, where the sampling distance is smaller than the signal's wavelength [111, 182]. However, CS guarantees efficient recovery by performing a convolution operation between an image and a random pulse or matrix. Important factors on this universal sampling technique is the sparse image representation and the incoherent (random) measurements. In essence, sparse image representation has been the key area of research for over the past 20 years with the most notable product being the Fourier and Wavelet transforms, which are also used in CS theory for enhancing

sparsity of images. For further details on measurement bounds, stability of the recovery and assumptions for accurate reconstruction see Chapter 4.

Consider now a general linear measurement process which collects $M \ll N$ samples as an inner product between the vectorised image F (of length N) and Sensing vectors $(\Phi)_{1\leq k\leq M}$ [42, 132, 186]:

$$Y_k =< F, \Phi_k >, \quad k = 1, 2, \ldots, M \tag{3.23}$$

or in matrix form,

$$Y_{M\times 1} = \Phi_{M\times N} F_{N\times 1} = \Phi_{M\times N}\Psi_{N\times N} W(F)_{N\times 1} = C_{M\times N} W(F)_{N\times 1}, \tag{3.24}$$

where Ψ is the sparsifying transform (e.g. Fourier transform), Φ is the Sensing or Sampling matrix (vectors ϕ_k are actually rows of matrix Φ) and $W(F)$ is the coefficients vector. Simply put, initially the image is transformed by taking its 2D spatial Fourier transform. After that a random mask or matrix is applied, where the acquired image convolves (multiplied) with a known pseudo-random pulse (random sampling matrix). This is the standard random convolution or measurement architecture for compressive imaging [39, 46]. The initial acquisition of the image can be implemented with a physical device, the Fourier transform is simply a mask (transformation) applied on the image, while the measurements are collected randomly, using an ordinary method of sampling images [42, 43]. Compressive imaging has a very high applicability in tomographic images as the data acquired from the scanning process are usually values of the Fourier transform of the desired image [33, 132, 186]. Examples in this category include magnetic-resonance imaging, x-ray transmission tomography, positron-emission tomography (PET) and single-photon emission tomography (SPECT) [33, 186]. These types of images are usually approximately locally constant with sparse discrete partial derivatives [33, 186].

A slightly different approach was proposed by Romberg for compressing and improving coarse resolution images [169, 170]. Coarse resolution images are mainly used to represent satellite data for ecological surveys or remote sensing projects and usually exhibit lower spatial resolution compared to other frequently used types of images [58, 106]. For example, in land cover, much of the details are lost with coarse resolution imagery, such as small fragments or boundaries. These limitations have lead to the introduction of different method-

ologies which are based on geostatistics and fractal analysis as a way to improve the quality from satellite imagery and thus make them more consistent with finer resolution imagery and photography [58, 106].

During the last few years there has been an increasing interest in designing algorithms for increasing the accuracy of information, while decreasing the size, from daily, coarse-resolution satellite images [58, 106, 169]. For a better and more efficient explanation of the algorithm proposed by Romberg in [169, 170], let's assume a 256×256 coarse resolution image. Initially the image is divided into a 64×64 grid of 4×4 (i.e. 16 pixels) blocks, where its values are summed over each of these blocks. This step allows us to represent an image acquired by averaging over relatively large regions or pixels. Then the resolution of the image is enhanced by taking another measurement of coarse pixels. In this case we get the same 4096 coarse-grid low-resolution measurements, as before, but then we perform a convolution (multiplication) of the image (obtained from these measurements) with a real-valued, known, random pulse (sampling matrix) by randomly modulating (encoding) the phases (spatial or temporal/quantisation enhancement of pixels) and summing again over the same 16 pixel regions. Using the two previous measurement processes we obtain a sum of 8192 measurements (in total) from which the image is reconstructed using the TV based optimisation problem defined in (3.26).

3.7.3 Sparse recovery in images

Following the collection of CS measurements $Y = CF = \Phi\Psi W(F)$ the decoding step is the following optimisation problem as a sparse recovery approach [39, 42, 87]:

$$\min_{\hat{F}} \|\Psi^* \hat{F}\|_{l_1} \quad s.t. \quad \Phi\hat{F} = Y, \tag{3.25}$$

Φ is again a linear operator (sampling matrix) which maps the image F to a set of M measurements, Ψ^* is the adjoint of the sparsifying matrix Ψ representing the inverse Unitary transform, such as IDFT, so as $\hat{W}(F) = \Psi^*\hat{F}$ ($\hat{F} \approx F$ due the approximation nature of the method). This optimisation problem (minimisation of the absolute value of the sum of magnitudes) searches for the sparsest image representation in the domain Ψ which fits the measurements Y and it is equivalent to minimising the l_1 norm in a domain of image representation, such as the Wavelet and Fourier domain.

An interesting variation of the l_1 based optimisation principle discussed in (3.25) is the image reconstruction using its Total Variation property (TV) in the spatial domain [43, 169, 170]:

$$\min_{\hat{F}} \quad TV(\hat{F}), \quad s.t. \quad C\hat{F} = Y \tag{3.26}$$

$$TV(F) = \sum_{i,j=1}^{N} \sqrt{(F_{i+1,j} - F_{i,j})^2 + (F_{i,j+1} - F_{i,j})^2} \tag{3.27}$$

where TV stands for the Total Variation of the image, $\hat{F}$ is the recovered image, $C = \Phi\Psi$ is the sampling matrix, Y are the collected samples and the $F_{i,j}$ is the light intensity (pixels) of the image at position (i, j). The total variation norm is in essence the l_1 norm for the gradient or directional change in the intensity or color of the image, which is mainly used for extracting information from images (assuming sparsity on the gradient). Moreover, in the case of complex valued images (i.e. a transform in a complex domain has been applied) the minimum TV problem in (3.26) can be recast as a special convex programming problem known as second order cone programming problem [39, 170].

A very similar approach is discussed in [186], where the purpose is to reconstruct a single image from six consecutive (small timeshift between the acquisition of two images) raster (low resolution RGB images) compressed and possibly noisy images (low quality high-speed scanning) obtained by telescope Herschel. Following the CS approach the sparse recovery of the final average image is obtained as a l_1 norm based minimisation problem [186]:

$$\min_{\hat{W}(F)} \sum_{i=1}^{P} \frac{1}{2} \|Y_P - \Phi_P S_P(\Psi \hat{W}(F))\|_{l_2} + \lambda \|\hat{W}(F)\|_{l_1}, \tag{3.28}$$

where $P = 6$ is the number of raster images collected, Φ_P is the sampling matrix used for each image collected, S_P is an operator used for shifting the image and mapping all of them together for composing the final image (for instrumental noise and model imperfections), while Ψ is a sparse dictionary used for image representation (e.g. Fourier transform) and $\lambda > 0$ is a small regularisation parameter based on the nature of the problem in order to enhance sparsity of the l_1 norm decoder. Note that in this case we have noisy measurements

and thus our linear measurements model is slightly different, $Y_P = \Phi F_P + \epsilon$ for all raster images $P = 1, \ldots, 6$ with $\epsilon > 0$ a small unknown error (expressed as additive noise) in the sampling process (representing instrumental noise and model imperfections) [186]. Also note that the most important feature in CS method applied to images (and large scale problems in general) is the complete randomness in the sampling process, which is very important here as the known observations are acquired from an unknown original image and the noise level is also unknown (though assumed to be small).

Another similar approach has been introduced in [173], where the aim is to detect minerals on a series of images of the Syrtis Major volcanic complex, taken from planet Mars by Mars Express (space exploration mission conducted by ESA). This is achieved by processing a series of images, taken by a sensor which measures the solar light reflected and scattered back from the surface and from the atmosphere. The sampling or collection of pixels can be modelled in a form of a two dimensional vector Y of size $PP \times LL$, where PP is the number of pixels for the image acquired in LL different mineral spectral bands (surface reflectance values). Every spectrum band can be expressed as a linear combination of RR mineral spectra bands forming a matrix X of PP rows (representing the number of pixels) and RR columns (representing the mineral spectral bands), while matrix A is the blending operator which contains RR rows of the spectral bands expanded in LL columns representing the pixel components. Then, the problem of the so-called "linear spectral mixing model" can be formulated as a minimisation of the least squares criterion in order to find X, given Y and A [173]:

$$\min_{\hat{X}} \|Y - XA\|_{l_2} + T\|X\|_{l_1}, \quad s.t. \sum_{j=1}^{RR} X_{ij} = 1 \ and \ X_{ij} \geq 0, \forall i \in PP, \forall j \in RR \tag{3.29}$$

where T is the non-negative, regularisation term which balances between quality of data fitting and prior information or confidence for the sparsity of the solution. Note the second condition which requires that the spectral band values should sum to one for each pixel and the non-negativity of the values of X.

Another major approach to the fundamental problem of image processing (also used in CS theory) is the use of functional spaces, such as Banach, Sobolev and Hilbert (see Appendix A) [2, 132]. Usually, this adoption of functional spaces for image analysis and processing is

measured by an energy $E[F]$ of an object, such as image. Classical Fourier and Frequency based approaches assume that images are drawn from $L^2(\Omega)$ and $E[F] = \|F\|_{l_2} = \|\mathcal{F}(F)\|_{L^2}$, where $\mathcal{F}(F)$ is the Fourier transform [2, 30]. The linear filtering theory and more modern approaches assume that images belong to Hilbert space $L^2(\mathbb{R})$ and their image information (pixels intensity) is measured by $E[F] = \|\nabla F\|_{l_2}$. To process images even better a famous approach is the total variation (TV) model or measure described in (3.26).

An pertinent approach for image sparse restoration is to model with the regularising term of higher order derivatives [53, 121]. This approach is sometimes favoured over the TV regularisation as it provides very efficient recovery while preserving all the properties of TV norm, such as image sharpness, convexity, homogeneity and rotation [121]. In fact, there is a growing interest in literature for applying a higher order derivative as a regularisation technique for biomedical images, where image interpretation requires the uttermost detail, and in general in the areas of image analysis and restoration, such as inpainting, zooming and de-noising. The majority of the proposed regularisers involve second-order differential operators to better fit smoother areas in images based on the Euler's elastica image model (spline model) [53, 121]:

$$E[F] = \int_\Omega \left(1 + \left(\nabla\left(\frac{\nabla F}{\|\nabla F\|_{l_1}}\right)\right)^2\right) \|\nabla F\|_{l_1} dF, \tag{3.30}$$

which is a variation of the famous Euler's elastica spline model which minimises the energy functional of the form [53]:

$$e[\gamma] = \int_\gamma x(a + bk^2)dx, \tag{3.31}$$

where k is the curvature of the curve γ, x is the arc length, and $a, b > 0$ some parameters. In the case of image analysis and restoration problems, the most representative regularisers are [121]:

1. The l_1 norm of the Laplacian operator for $\Omega \in \mathbb{R}^2$:

$$E[L] = \int_\Omega \|\Delta F\|_{l_1} dF, \tag{3.32}$$

where $\Delta F = \theta^2 F/\theta_i^2 + \theta^2 F/\theta_j^2$ and $i, j \in \mathbb{Z}^2$ are the positions of the pixels of the image F (Cartesian coordinates in a plane).

2. The Frobenius norm of the Hessian for $\Omega \in \mathbb{R}^2$:

$$E[F] = \int_\Omega \sqrt{\theta^2 F^2/\theta_i^2 + 2\theta^2 F^2/\theta_i\theta_j + \theta^2 F^2/\theta_j^2} dF \tag{3.33}$$

3. The affine TV functional for $\Omega \in \mathbb{R}^2$:

$$E[TV] = \int_\Omega \left(\sqrt{\theta^2 F^2/\theta_i^2 + \theta^2 F^2/\theta_i\theta_j} + \sqrt{\theta^2 F^2/\theta_j\theta_i + \theta^2 F^2/\theta_j^2} \right) dF \tag{3.34}$$

4. The regulariser based on the l_q norm of derivatives:

$$E[l_q] = \int_\Omega \|\Delta^p F\|_{l_q}, \tag{3.35}$$

where $\Delta^p F = \theta^p F/\theta_i^p + \theta^p F/\theta_j^p$ is the p-th order derivative and q determines the order of the norm used for penalisation (usually $q = 1$ or $q = 2$).

However, note that only TV regularisation method will be used in image recovery experiments and only for comparison purposes with other competing l_1/l_2 norm based sparse recovery methods.

Chapter 4

Sparse approximation as a problem

The aim of sparse approximation problem is to solve a matrix equation $CX = Y$, or more precisely to find (reconstruct, estimate, approximate or interpolate) X from given limited measurements Y as a linear problem. Here, X is the (unknown) K-sparse vector (signal) we are looking for (or the coefficients of this signal), Y is the known measured vector (i.e. the samples) and C is the known Measurement or Sensing matrix. In a general case, Y can represent partial observations or measurements (real or complex) sampled or compressed by a linear operator C (or mapping function from N to M, with $M \ll N$). For example, C can represent a sampled complex exponential sinusoid model of a partial set of Fourier frequencies $\omega = [\omega_1, \ldots, \omega_M]$, where ω could be $[1, e^{-j\omega}, e^{-2j\omega}, \ldots, e^{-j(M-1)\omega}]$ (in vector format, see Equation (2.44). These sampled values might be generated within a specific range (e.g. $(-\pi, +\pi)$) which may or may not be uniformly spaced and the chosen partial frequencies might not match exactly the (basis selection process) harmonics of original signal X (i.e. presence of noise). We aim however to pick the best possible values for X which are closest to the original signal.

Note, that since vector X has N terms or entries and the vector Y has M terms or entries with $M \ll N$, the sparse recovery problem, as a system of linear equations, is under-determined (also referred as ill-posed or ill-conditioned). This is a class of problems which often does not fulfil Hadamard's postulates of well-posedness [11, 39, 42, 46, 75]. According to Hadamard (1902) any well-posed problem [16] must fulfil the following three properties

[16] These types of problems usually aim to recover a function from a certain number of possibly noisy

[113, 184, 194]:a) a solution exists for all admissible data (existence), b) the solution is unique for all admissible data (uniqueness) and c) the solution depends continuously on the data (stability). If any of these properties is violated then the problem is ill-posed. In the context of data, generally, we refer to the measured inputs or states. Many classical mathematical problems are considered to belong to the class of ill-posed problems with many important applications, including the areas of engineering, physics and finance.

Based on the core principles of CS theory, it is possible to recover sparse signals efficiently from what appears to be highly incomplete sets of linear measurements using the constrained l_0-minimisation problem as a signal reconstruction problem. This novel method for sparse signal recovery uses an objective function together with the constraints so as to define the problem more explicitly and particularly to minimise the number of non-zero variables, i.e. to derive K non-zero entries (out of N) from the much fewer (M) measurements. The whole process is actually a search for the distinguished (non-zero) elements which bears some resemblance with the famous Sum of the Subsets problem and the Interpolation problem in general. It also reminds us the famous search of the counterfeit coin from a sack of coins where using a scale we can derive the weight of genuine coins so as to check the coins and discard the fake one. The l_0-norm based minimisation problem can be defined as follows:

$$\min_{\hat{X}} \|\hat{X}_{N\times 1}\|_{l_0} \quad s.t. \quad Y_{M\times 1} = C_{M\times N}\hat{X}_{N\times 1}, \tag{4.1}$$

Since the signal X we want to reconstruct is K-sparse and it can be formulated as a linear combination of the matrix vector product between a Basis Ψ and a vector W ($X = \Psi W$), we can rewrite the same problem equivalently and more precisely as follows:

$$\begin{aligned} &\min_{\hat{W}} \sum_{i=1}^{N} \|\hat{W}_i\|_{l_0} = \sharp\{n : \hat{W}_n \neq 0\} = K \\ &\qquad s.t. \quad Y_{M\times 1} = \Phi_{M\times N}\Psi_{N\times N}\hat{W}_{N\times 1}, \end{aligned} \tag{4.2}$$

measurements which are thought to be unsolvable as an approximate solution derived from the equation was considered as being of little practical use for most of the problems applied [113, 194]. Methods, such as the traditional least-squares and maximum likelihood provide solutions that may not be unique and subject to large changes caused by a small change on data (samples). However, the requirements set for ill-posedness are now considered as minimal since much stronger requirements, such as measurement error and model uncertainty, connected to the observations can also be included [113, 194].

where $X = \Psi W$, Y represents the under-sampled measurements and Φ is the incoherent measurements model. This represents a pseudo-random classical scheme of sampling based on a probability distribution (i.i.d. random waveforms), such as a random Gaussian (Normal), Binary (Bernoulli) Distribution or even randomly selected Fourier samples. X is the concise K-sparse vector (signal) of length N, approximated by K terms instead of the N terms that initially has. This optimization principle represents the decompression technique we have to follow in order to recover the vector (signal) from its collected samples. Note that this problem has been proven to be NP-Hard as we will see in the following Section 4.1.

4.1 The Complexity of the sparse recovery problem

This Section discusses the complexity of the sparse recovery problem and the complexity of optimal approximation in general; well-known problems arising in Compressed Sensing. In particular, we show that M-approximation problem, the l_0 and l_2 approximation problems are all NP-Hard (i.e. Non-deterministic Polynomial) and thus any straightforward greedy recovery algorithm is necessarily computationally infeasible.

Following the analysis of Subsection 2.3.1 we assume a linear redundant dictionary of waveforms (such as a Fourier matrix), represented by elementary functions as Ψ_M ($M \in \mathbb{Z}$) in any domain. The aim is to approximate a N element vector $X \in \mathbb{R}^N$ by a set of a linear combination of $M \ll N$ such elementary functions.

Definition 6: [69, 175] *Let $X \in \mathbb{R}^N$ be a N element vector approximated by a set of elementary functions Ψ_M, where $M \ll N \in \mathbb{Z}$. The (ϵ, M)-approximation is an expansion*

$$\hat{X} = \sum_{i=1}^{M} W_i(X)\Psi_i, \tag{4.3}$$

where we seek to minimise the approximation error

$$\|\hat{X} - X\|_{l_2} \leq \epsilon, \tag{4.4}$$

where $W_N(X)$ is the coefficient vector of X, representing the vector's behaviour in the domain to which Ψ belongs (e.g. Fourier transform), while $\hat{X}$ is the M-optimal approximation

of X (in its original domain) which minimises the approximation error.

In general, in any transform it is desired that $\hat{X} = X$. If the transform is orthogonal (Ψ columns are unit-length, see Section 4.2) then we can obtain an M-optimal approximation by computing all the inner products $|\langle \Psi, X \rangle|$ and then sorting them so as to find the M-optimal approximation as [69]

$$\hat{X} = \sum_{i=1}^{M} \langle X, \Psi_i \rangle \Psi_i \tag{4.5}$$

However, the problem of optimally approximating a vector using a linear redundant dictionary of waveforms (such as Fourier matrix) is NP-Hard, as finding the optimal dictionary waveform that best matches the sparse vector's structure is also NP-Hard. In fact, the following theorem shows that finding M-optimal approximations is an NP-Hard problem.

Theorem 5: [69] *Let H be a N-dimensional vector space (See Appendix A for details) and Ψ_N be the set of all dictionaries (elementary functions) for this domain H that contain $O(N^\kappa)$ vectors, where $\kappa \geq 1$. Also let $0 < \bar{\alpha}_1 < \bar{\alpha}_2 < 1$ and $M \leq N$ such that $\bar{\alpha}_1 N \leq M \leq \bar{\alpha}_2$. Then the (ϵ, M)-approximation problem of finding an optimal approximation for any vector $X \in \mathbb{R}^N$, given $\epsilon > 0$, $\langle X, \Psi_N \rangle \in H$ is NP-Hard.*

Note that the previous theorem does not imply that the (ϵ, M)-approximation problem is intractable for specific dictionaries. Indeed for orthonormal dictionaries the problem can be solved in polynomial time [69]. However, the (ϵ, M)-approximation problem is NP-Hard for any given dictionary in a vector space; it is as hard as the exact cover by 3-sets problem, which is known to be NP-Hard, by reduction. For details about the Set Covering problem in general see Appendix B.

Definition 7:[69] *Let X be a set containing $N = 3M$ elements and let C be a collection of 3-element subsets of X. The exact cover by 3-sets problem is to decide whether C contains an exact cover for X; to determine whether every member of X occurs in exactly one member of a sub-collection contained in C.*

Lemma 1:[69] *Any instance (X, C) of an exact cover by 3-sets problem of size $N = 3M$ and $\bar{\alpha}_1 = \bar{\alpha}_2 = 1/3$ can be transformed in polynomial time into an equivalent instance of (ϵ, M)-approximation problem with a dictionary of size $O(N^3)$ in a N-dimensional vector space.*

Note that the complexity analysis of the (ϵ, M)-approximation problem described previously can also be generalised for any arbitrary numbers $0 < \bar{\alpha}_1 < \bar{\alpha}_2 < 1$ and dictionaries of size $O(N^\kappa)$, for $\kappa > 1$, given an instance of exact cover by n on a $3n$-dimensional vector space H. Indeed, consider now the sparse approximate problem with $C \in \mathbb{R}^{M \times N}$ (M rows and N columns with $M \ll N$), vector $Y \in \mathbb{R}^M$, unknown vector $X \in \mathbb{R}^N$ and $\epsilon > 0$ a small constant. The aim, as it has already been previously described, is to find the sparsest vector, over all such vectors X which satisfies the property $\|CX - Y\|_{l_2} \leq \epsilon$. Two important aspects are of particular interest to mention [28, 69, 146]: a) the cost of evaluating X increases with the number of non-zero entries in the solution and hence it is desirable to have as few non-zero entries as possible, with a small recovery error in the interpolation process. b) The interpolant with the fewest non-zero entries is expected to approximate the whole function that generated the samples. Especially under the presence of noise the best practice is to select the sparsest yet approximate interpolant for X. We will now prove that the sparse recovery problem is NP-hard on a (Turing) machine model of infinite precision.

Theorem 6: [146] *The sparse recovery problem is NP-Hard, by reduction to the Set Covering problem.*

Proof [146]: The proof is by reduction from the problem of exact cover by 3 sets. Let's assume an instance of a set S and a collection A of 3-element subsets of S. We want to show whether A contains an exact cover for S, i.e. a sub-collection of A such that every element of S occurs exactly once in the sub-collection. We will transform an instance of exact cover by 3-sets to an instance of the sparse recovery problem. Assume an instance of exact cover by 3-sets, $S = s_1, s_2, \ldots, s_M$, and $A = a_1, a_2, \ldots, a_N$. Assume also that M is a multiple of 3 and Y is a vector $[1, 1, 1, \ldots, 1]$ of M ones. The matrix C will have N column vectors, one for each set in A. In particular, for each set $a_i \in A$, $i = 1, 2, \ldots, N$, the corresponding column vector will have entries $[Z_1, Z_2, \ldots, Z_M]$, where $Z_i = 1 \; \forall i \in \{1, ..., M\}$ iff $s_i \in a_i$ and $Z_i = 0$ otherwise. Finally, assume that $\epsilon = 1/2$. The given sparse recovery problem has a solution with $M/3$ or fewer entries if and only if the given instance of exact cover by 3-sets has a

solution. Indeed, consider a vector $X = [X_1, X_2, \ldots, X_N]$, where $X_i = 1 \ \forall i \in \{1, ..., N\}$ if the i-th set in A is included in the solution of the instance of the exact cover by 3-sets problem and $X_i = 0$ otherwise. Then, $CX = Y$ exactly and hence the exact cover instance has a solution X with at most $M/3$ non-zero entries. Since $\|CX - Y\|_{l_2} \leq \epsilon = 1/2$, each entry of CX must be between $1/2$ and $3/2$. Since each column of C has only 3 non-zero entries, X must have at least $M/3$ entries. Thus X has exactly $M/3$ entries. Now consider the sub-collection of A, consisting of those sets a_i such that the i-th entry of X is non-zero. It is clear that this sub-collection is an exact cover for S.

Following the previous complexity analysis of the generalised least-squares loss function calculation, namely $\|CX - Y\|_{l_2}$ we will now extend this concept to introduce sparsity constraints and penalties so as to provide the complexity of the l_0 norm based sparse recovery problem in a more general form. Indeed, the l_0 norm based problem can be expressed by the following three non-equivalent (due to non-convexity) optimisation principles, where the constrained forms are used when prior knowledge of sparsity of the solution or the noise level are known [151, 183].

Theorem 7: [151, 183] *The sparse recovery problem expressed by the following optimisation principles is NP-Hard:*

$$\min_{\hat{X}} \|\hat{X}\|_{l_0} \quad s.t. \quad C\hat{X} = Y, \tag{4.6}$$

$$\min_{\hat{X}} \|C\hat{X} - Y\|_{l_2} \quad s.t. \quad \|\hat{X}\|_{l_0} = K, \tag{4.7}$$

$$\min_{\hat{X}} \frac{1}{2}\|C\hat{X} - Y\|_{l_2} + T\|\hat{X}\|_{l_0}, \tag{4.8}$$

where T is a small regularisation parameter between data fidelity and sparsity, while K is the sparsity level of vector $X \in \mathbb{R}^N$.

Note that in order to solve directly the l_0 norm based problem, one must find all the possible K out of N nonzero components in X, which is intractable as the search space is expo-

nentially large. However, the same problem has many basic feasible (non-unique) solutions and thus its local minima can be easily found [6, 145, 151]. For this reason, sometimes this problem is stated as "both hard and easy", in a sense that its decision problem is in P (polynomial), but its optimisation problem is in NP [92, 142, 161, 213]. In other words finding a global minimum is NP-Hard but finding a local minimiser is achieved in polynomial time [92, 142, 161, 213].

Furthermore, results in [92, 126, 141] have shown that an optimal basic feasible solution for l_1 minimisation problem is not always the sparsest solution as in some cases not all basic feasible solutions have a good sparse structure under l_1 norm. Therefore, although it is very computationally expensive to reconstruct the original vector exactly, it is possible to reconstruct a good approximation of it (using an approximate polynomial time algorithm). However, decision and optimisation problems are fundamentally different from a complexity point of view and an approximate solution for a decision problem is particularly useless. A more detailed analysis on this issue together with some naive approaches of solving the l_0 norm based problem are discussed in Section 6.5. For a more detailed treatment on the l_0 complexity, the hardness of finding optimal solutions and the conditions for relaxation of the l_0 norm see [39–41, 75]. Details in [50, 69, 92, 145, 146] provide further insights on the complexity of the sparse recovery problem and other relevant NP-hard problems, together with some general solution approaches.

4.2 The Nullspace property

Suppose $X \in \mathbb{R}^N$ is the vector we want to sample and $Y \in \mathbb{R}^M$ is the number of samples to collect with $M \ll N$, using a matrix $C : \mathbb{R}^N \to \mathbb{R}^M$. Also consider the recovery matrix $\Delta : \mathbb{R}^M \to \mathbb{R}^N$ and thus our approximation is defined as $\Delta(C(X)) = X_K$, where X_K is a K-sparse approximation of X, found by solving the l_1 optimisation problem $\min \|X\|_{l_1}\ s.t.\ CX = Y$, so as $CX_K = Y$ [68, 215]. We want as few measurements as possible and thus we will also assume efficient sampling is obtained so that the rows of C are all linearly independent. We are interested in the case where the null space of sampling matrix C, $Null(C) = \{X : CX = 0\}$, does not imply invertible mapping, $\Delta(C(X)) \neq X$ [68, 215]. Note that Δ is not necessary a linear map and therefore for any sample Y we will thus have a class of vectors that would obtain the same set of measurements, $F(Y) = \{X : CX = Y\} \subseteq \mathbb{R}^N$. Then the recovery of X_K is possible if and only if there is no vector ν in the null space (ker-

nel) of matrix C so that $x_k+\nu$ is interior to (or intersects the boundary of) the l_1 ball (space) defined by $\|X_K\|_{l_1}$ [68, 138, 215]. In fact assume that $X_K \in Null(C)$ and $\nu \in Null(C)$, then $CX_K = C\nu$, which by linearity gives $C(X_K - \nu) = 0$, or $X_K = \nu$. Otherwise $X_K, \nu \in F(Y)$, which means $X_K - \nu \in Null(C)$ or $X_K \in \nu + Null(C)$, which violates our hypothesis. Further details can be found in [68, 138, 163]. The Nullspace property (NSP) now can be summarised as follows:

Theorem 8:[68, 138, 215] *a)If C, Δ pair is optimal on X for K, then C satisfies NSP for $2K$ support on X by a constant. b)If C satisfies NSP for X and $2K$ then there exists Δ so as C has the optimal X for support K (X_K).*

In essence, NSP checks how any X_K, which is K-sparse, efficiently affects the null space of C. NSP is mainly connected with the l_1 norm based minimisation. In is not a property that requires sparse approximation and in fact the null space results are much weaker than the associated RIP based results. However, NSP and RIP prove a "strong equivalence", that the matrix can be used for compressed sensing of any K-sparse vector [68]. Note there is also another version of NSP, less restrictive that ensures a matrix can be used for Compressed Sensing for all but a small fraction of K-sparse vectors. In practice this can be achieved when selecting a K-sparse vector at random. For further discussion on NSP see [68, 74, 215].As we will see in the next section UUP and RIP conditions are just generalisations of rectangular orthogonal matrices with respect to their columns. As a quick reminder recall the definition of rectangular orthogonal matrix:

Definition 8:[189, 215] *If $C \in \mathbb{R}^{M\times N}$ is an orthogonal matrix, then the column vectors of $C = [C_1, C_2, \ldots, C_N]$ are orthogonal which means that they have unit length ($\|C_i\|_{l_2} = 1, \forall i = 1, 2, \ldots, N$) and are orthogonal to each other ($< C_i, C_j >= 0$ for $i \neq j$). Also if $(B_1, B_2, \ldots, B_N)$ is a N-dimensional vector $B \in \mathbb{R}^N$ then the linear combination (i.e. encoding) of C and B, $B \to \sum_{i=1}^N C_i B_i$ is an isometry since $\|\sum_{i=1}^N C_i B_i\|_{l_2} = \sum_{i=1}^N \|B_i\|_{l_2}$. This implies that the encoding can be inverted as we can derive the coefficients vector B uniquely from $\sum_{i=1}^N C_i B_i$, while small fluctuations will not affect B.*

In fact, these notions are not new, they were firstly referred as "neighborliness" in the convex polytope community. David L. Donoho and Emmanuel Candès analysed these concepts

in detail back in 2005 (see [78] and [87] respectively). For completeness in our discussion we also present here the connection between nullity and rank through the following theorem:

Theorem 9 (Rank-Nullity Theorem):[68, 138, 163] *For any invertible matrix $C \in \mathbb{R}^{M\times N}$ its rank plus nullity is N (number of columns).*

For any matrix $C \in \mathbb{R}^{M\times N}$, its rank is equal to the number of linearly independent rows ($|\text{rank}(C)| < \min\{M, N\}$) and the dimension of nullity is $\text{null}(C) \in \{0, N\}$. Note that if $M = N$ and thus C is square, then $\text{rank}(C) = N$, C is invertible and $CX = Y$ has a unique solution for any Y, since $\varnothing$ is only mapped to $\varnothing$ (the solution space is exactly defined as we do not have not have any non-zero values mapped to $\varnothing$ by C).

4.3 The Restricted Isometry Property

NSP is a weaker assumption than UUP/RIP used for establishing guarantees for sparse approximations in the l_1 norm based minimisation problems. Unlike with NSP, UUP/RIP establish stronger conditions, as more general robust principles, applied to many sparsity based algorithms. UUP/RIP also consider noisy cases, where measurements can be contaminated with noise or be corrupted by some quantisation error. In essence RIP implies NSP, as a sufficient condition to guarantee that NSP is satisfied, but not the other way round [68]. Before defining RIP, we firstly need to assume that X already represents the coefficients of the signal in the appropriate basis; this means that we pick an approximation to X in the set $\sum_K$ ($X \in \sum_K$ with K non-zero entries) as a K-sparse vector. We have [40, 43, 68, 180]:

$$\sum_K = \{X \in \mathbb{R}^N : \sharp\,\text{sup}(X) \leq K\}, \tag{4.9}$$

where $\text{sup}(X)$ is the support of a set of indices i for which $X_i \neq 0$ and $\sharp$ is the number of elements in a set. The best approximation can be achieved by using the norm $\|.\|_{l_p}$ and it is described as the best K-term approximation as [68, 69, 215]:

$$\epsilon_K(X)_p := \inf_{z\in\sum_K} \|X - X_K\|_{l_p}, \tag{4.10}$$

where $p = 0, 1, 2$ defines the type of the norm and X_K the best K-term approximation of X. Using this process we assume that the derived X_K varies from the original X, which requires first to compute all of the basis coefficients. Note that finding in general the best K-term approximation as a problem is NP-Hard, though for specific bases (e.g. orthonormal dictionaries) the problem can be solved in polynomial time [69]. In other words, the problem of optimally approximating a function with a linear expansion over a redundant dictionary of waveforms is NP-hard. Further details can be found in [69].

Recently, in [40, 41, 43, 75, 87] Candès, Donoho and Tao argued that since we retain only a few of these coefficients in the end, perhaps it can be possible to actually compute only a few non-adaptive linear measurements in the first place and still retain the necessary information about X in order to build a compressed representation. For this purpose they introduced the concept of CS, together with the initial notion of the Uniform Uncertainty Principle (UUP), which they afterwards refined it by the Restricted Isometry Property (RIP) for the Sensing matrix C. Both of these properties are very important for the stability of CS decoder as these properties characterise matrices which are nearly orthonormal, when working on sparse vectors. Note that there are not any large matrices with bounded isometry constants known, but many random matrices have been shown to remain bounded. In particular, it has been shown that with exponentially high probability, random Gaussian, Bernoulli, and partial Fourier matrices satisfy the UUP and RIP properties with the required number of measurements nearly linear in terms of sparsity level [39–41, 43].

The UUP was introduced by Candès and Tao in 2005 and states that the measurement matrix C has to obey the following property [48, 87]:

$$(1 - \delta_K)\|W\|_{l_2} \leq \|C_l W\|_{l_2} \leq (1 + \delta_K)\|W\|_{l_2}, \tag{4.11}$$

for all subsets $l \subset \{1, 2, \ldots, M\}$ with $|l| \leq K$, C_l denotes a $|l| \times N$ submatrix consisting of columns of C indexed by the elements in l, δ_K (for the support K of W) is a small constant based on C and W_j, $j \in \{1, 2, \ldots, M\}$ represents the coefficients of a matrix X represented by a dictionary. This property in other words states that every submatrix C_l with cardinality less than the sparsity level K must be an orthonormal system [68]. It has also been proven that $\delta_K + \delta_{2K} + \delta_{3K} < 1$ based on a K sparse vector with support size $|l| \leq K$ [87]. Note also that it is NP-Hard to verify that UUP holds for large K as there is exponentially large

number of different submatrices to consider and test [189]. The RIP property now follows:

Theorem 10:[34, 40, 41, 180] *A matrix C satisfies the RIP of order K if there exists $\delta_K \in (0,1)$ constant such that:*

$$(1-\delta_K)\|X\|_{l_2} \leq \|CX\|_{l_2} \leq (1+\delta_K)\|X\|_{l_2}, \tag{4.12}$$

holds for all $X \in \sum_K$ and $K = 1, 2, \ldots,$.

A vector X is said to be K-sparse if it has K non-zero entries. Note also that if C satisfies the RIP of order K with constant δ_K, then for any $K' < K$ we automatically have that C satisfies the RIP of order K' with constant $\delta_{K'} \leq \delta_K$. Moreover, in [149] it is shown that if C satisfies the RIP of order K with a sufficiently small constant, then it will also automatically satisfy the RIP of order γK for a certain small γ, with a deterioration in δ constant. In particular we have the following Lemma:

Lemma 2:[149] *Suppose that C satisfies the RIP of order K with constant δ_K. Let γ be a positive integer. Then C satisfies the RIP of order $K' = \gamma\lfloor\frac{K}{2}\rfloor$, with constant $\delta_{K'} < \gamma\delta_K$ and $\lfloor . \rfloor$ denotes the floor operator or function.*

Notice that apart from the cases of $\gamma = 1, 2$ when $\gamma \geq 3$ and $K \geq 4$, the Lemma allows us to extend the RIP of order K to higher orders. However, δ_K must be sufficiently small in order for the resulting bound to be applicable [68]. Note that for simplicity reasons in the analysis we assume symmetric bounds around 1 in the previous definitions. In practice, we can consider arbitrary bounds as [68]:

$$\alpha\|X\|_{l_2} \leq \|CX\|_{l_2} \leq \beta\|X\|_{l_2}, \tag{4.13}$$

where $0 < \alpha \leq \beta < \infty$. Given any such bound we can easily scale C to satisfy the necessary symmetry by multiplying the bounds with a proper ratio.

The original work of Candès in [42] considers a symmetric RIP constant $\delta_K = \max(\delta_K^{\max}, \delta_K^{\min})$, who proved that a small enough value for δ_K is enough to efficiently recover all K-sparse vectors. In fact it has been shown that the sufficient condition for sparse recovery (lower

bound) is $\delta_K \leq \sqrt{2} - 1$ [40], while in [91] recovery is sufficient when $\delta_K < 0.4531$. Recently, in [34] it has been shown that stable recovery can be achieved even under the presence of noise on condition that $\delta_K < 0.307$, which cannot be substantially improved in a noiseless case. In addition to that the upper bound of 0.5 for δ_K cannot be increased in general so as to guarantee stable recovery of K-sparse signals [34].

It is important at this point to note that RIP conditions are in general hard to verify for any general sensing matrix C but easy to verify for specific matrices [34, 40, 189]. In fact, RIP is a deterministic property applied to Sensing matrices in general, but it is known to work with specific probability density function matrices. There is yet no paper ensuring that RIP will hold for any given matrix and in general finding values of δ_K is NP-Hard (but easy to verify them) [12, 42, 189]. In general, it has been shown in [12, 40, 186] that RIP holds for specific Sensing matrices $C = \Phi\Psi$, where Ψ is a sparsifying transform basis and Φ the measurement matrix randomly generated from a favourable distribution (see next Section 4.4). The matrix ensemble (i.e. random/stochastic matrix) Φ which are known to obey UUP and RIP conditions are for example random normalised Gaussian, Bernoulli and DFT matrices [12, 189]. Many matrix ensembles are also proven to have the NSP property; see [77] by Donoho and Tanner. However, for large sparsity levels, there are only probabilistic ways of proving that the previous isometry conditions hold with a small probability of failure [12, 189]. In fact, there is not any fast (e.g. sub-exponential) algorithm known to test these conditions, while the obvious way of searching all submatrices in a given class and test each one for UUP/RIP conditions is practically infeasible [12, 189]. Indeed, it has been proven in [44, 48] that columns of Sensing matrices can be as large as $M \exp(\alpha M/K)$, for some absolute small constant $\alpha > 0$, which makes it easy to construct (since in high dimensional space the random selection of two orthogonal vectors is easy) but hard to check all the submatrices for isometry.

However, for relatively small $K \approx \sqrt{M}$ it is easy to obtain Sensing matrices [44]. DeVore in [70] showed that it is possible to polynomially construct Sensing matrices with entries $\pm 1/\sqrt{M}$ using deterministic methods given that $K \leq \mu\sqrt{M} \log N / \log(N/M)$, N equal to an arbitrary large power of M and $\mu > 0$ a small constant. At this point it is important to mention that RIP is just a sufficient condition to guarantee the performance of the design of the Sensing matrix. Perhaps there might exist specifically constructed matrices that have the same performance in compression and yet one may satisfy RIP conditions while the

other not. Alternatively, there may be matrices for which RIP can be easily verified and still likely holds for them. Yet another interesting issue is the concept of sparse recovery under oversampling ratios (overdetermined linear system) where RIP or UUP are only conjectured to hold [189]. Only recently some robust results on compressible signals (rather than sparse) under the presence of noise have appeared. For a more detailed discussion on these areas see [100, 189]. It is yet to be seen in the scientific community any universal results where RIP and UUP will work for all types of sparse and compressible signals under noisy data instead of generic sparse data, using either probabilistic or exact algorithms for producing rigorous results. Towards this aspect it would also be particularly beneficial in large problem sizes to derive improved bounds on the RIP constants for different ensembles and how these bounds can be used to satisfy RIP or UUP conditions.

In general, the results derived from RIP and its generalisation UUP are very important as they ensure efficient recovery of all sparse vectors from undersampled measurements. Both of these properties are designed to guarantee the uniqueness of the solution we want by showing that C is an approximate isometry for all vectors restricted to be K-sparse (i.e. $\|CX\|_{l_2} \approx \|X\|_{l_2}$ and thus the l_0 norm can be relaxed by the l_1 norm with similar results). They provide sufficient guarantees that the distance between the measurements of two vectors $Y = CX$ and $Y' = CX'$ are proportional to the distance between the original vectors X and X' respectively. In essence, they show that two sparse vectors with small measurements error cannot be the same as all submatrices of C with size $M \times K$ are close to an isometry. This important outcome, namely that CX and X have about the same norm, can be seen through a simple yet representative example. Imagine for example that X has a dimension of 10^{+50} with only a few K non-zero elements in it. Computing the norm of this high dimensional vector might take some time. Instead, as C satisfies RIP condition with support K we can compute the norm of CX which is equivalent. Matrix C can act as a universal compressor for any sparse vector without much loss and thus using a sparse recovery method we can then find the original vector and project back to that. Note, however, that it is NP-hard to compute exactly the RIP constant δ_K for matrices C of high dimension.

One way to achieve that would be to find all the K sparse vectors of unit norm, compute the norm $\|CX\|_{l_2}$ and then take the maximum values away from 1 [40]. Relaxation methods to evaluate approaches and find RIP constants deterministically are in their infancy [12, 40].

On the other hand, sparse recovery methods in general are designed under the assumption that C satisfies RIP [41, 209]. This is important as C^TC acts like an isometry when applied to sparse vectors. In fact some l_1 norm based solvers and greedy methods make estimations of a new solution based on C^TY (e.g. L_1 Magic). In this context we can use data from Y to approximately estimate X (back-projection). Alternatively some greedy algorithms replace C^T with the pseudo-inverse $C^T(CC^T)^{-1}$ which can be viewed as a special case of preconditioning data in Y [142, 213]. The reconstruction accuracy can also commonly improved by enforcing non-negativity in X while projecting back ($C^T(CC^T + \delta I)^{-1}Y$), $\delta > 0$, i.e. Regularised Least Squares) [28, 163].

4.4 Stability of the solution

Let's assume that $\hat{X}$ is the best sparse approximation we could obtain if the locations and amplitudes of the k-largest entries of X were known. Assume also that $\delta_{2K} < \sqrt{2} - 1$. Then the solution $\hat{X}$ obeys the following rules for some constant ϵ_1 (for l_1 minimisation) [40, 41, 43]:

$$\|\hat{X} - X\|_{l_1} \leq \epsilon_1 \|X - X_K\|_{l_1} \tag{4.14}$$

and

$$\|\hat{X} - X\|_{l_2} \leq \epsilon_1 K^{-1/2} \|X - X_K\|_{l_1}, \tag{4.15}$$

for small values of ϵ_1 depending on the RIP constant. For example for $\delta_{4K} = 1/5$, $\epsilon_1 \leq 8.77$ [39], while if $\delta_{3K} \leq (\sqrt{2}-1)^2/3$, then $\epsilon_1 = \frac{2\sqrt{2}+2-(2\sqrt{2}-2)\delta_{3K}}{\sqrt{2}-1-(\sqrt{2}+1)\delta_{3K}}$ [61] (for a more detailed analysis in K-term approximation and optimal bounds see [61]). X_k represents the real best K-sparse approximation of X. If X is K-sparse then the recovery is almost exact, which means that if $\delta_{2K} < 1$, the l_1 norm based problem has a unique K-sparse solution. Also for $\delta_{2K} < \sqrt{2} - 1$, the solution to the l_1 norm based problem is equivalent to that obtained by the l_0 norm based problem. In other words, the convex relaxation (l_1 minimisation) derives the sparsest solution on sparse vectors with high probability [40, 41, 43]. Alternatively, if we assume that $\delta_{2K} = 1$, then $2K$ columns of C Sensing matrix may be linearly dependent and thus there is a $2K$-sparse vector h so as $Ch = 0$. Then we can decompose $h = X - X'$, where both X

and X' are K-sparse. But now we have $CX = CX'$ which means that $X = X'$. For further details on this topic see [40, 41, 43, 68, 87, 180]. Note an alternative definition of the stability of the solution for the l_0 norm based problem can be defined as:

Theorem 11:[39, 48, 75] *Let T and Ω be subsets of $\{0, 1, \ldots, N-1\}$ for N a prime number and length of a vector X which is supported on $\Gamma = T \cup \Omega$ with $CX = Y$ such that $|T| + |\omega| < N/2$. Then the solution to l_0 norm based problem in Equation (4.1) is unique and equal to original vector X.*

Consequently, if there exist two distinct vectors X_1, X_2 obeying $|\sup(X_1)|, |\sup(X_2)| < N/2 + 1$, then $CX_1 = CX_2$ and thus $X_1 = X_2$. Finally in terms of best K-term approximation using a particular dictionary we have the following results for the stability of the solution, recovered $\hat{X}$ from original vector X in terms of measurements M:

1. Fourier Dictionary [167]: $\|X - \hat{X}\| \leq M^{-1/2}$.
2. Wavelets Dictionary [104, 167]: $\|X - \hat{X}\| \leq M^{-1}$.
3. Curvelets Dictionary [104, 167]: $\|X - \hat{X}\| \leq M^{-2}$.

For the noisy case now, let's assume again that $\delta_{2K} < \sqrt{2} - 1$ (RIP constant δ) and $\|e\|_{l_2} \leq \lambda$, where e represents the noise in the sampling process and $0 < \lambda < 1$ a small constant. Then the solution of l_1 norm based problem obeys the following rule (general model) [42, 43, 87]:

$$\|\hat{X} - X\|_{l_2} \leq \epsilon_1 K^{-1/2} \|X - X_K\|_{l_1} + \epsilon_2 \lambda, \tag{4.16}$$

with the same constant ϵ_1 (for the approximation error) as before and some ϵ_2 as another small constant (for the measurement error). Both of these constants are very small $0 < \epsilon_1, \epsilon_2 < 10$. For example, if $\delta_{2K} = 1/4$, then $\epsilon_1 \leq 5.5$ and $\epsilon_2 \leq 6$ [39], while if $\delta_{2k} = 1/4$, it has been proven in [40] that the error of recovery for (4.16) is less than $4.2K^{-1/2}\|X - X_K\|_{l_1} + 8.5\lambda$. Another interesting outcome that derives from the previous theorems is as that $| < CX, CX' > | \leq \delta_{s+s'} \|X\|_{l_2} \|X'\|_{l_2}$, holds $\forall X, X'$ supported on disjoint subsets $T, T' \subseteq \{1, 2, \ldots, N\}$ with $|T| \leq s$ and $|T'| \leq s'$.

In general, if matrix C satisfies the RIP then this is sufficient for a variety of algorithms to be able to successfully recover a sparse vector (image or signal) from noisy or non-noisy

measurements. In fact, this means that a small amount of additive noise to the measurements will not considerably affect the sparse recovery. More precisely this can be alternatively defined as [68, 215]:

$$\|\Delta(CX + e) - X\|_{l_2} \leq \epsilon \|e\|_{l_2}, \tag{4.17}$$

for a small constant $\epsilon > 0$, $\Delta : \mathbb{R}^M \to \mathbb{R}^N$ the recovery algorithm and $C : \mathbb{R}^N \to \mathbb{R}^M$ the Sensing matrix. A slightly different result is given in [209] where noise is assumed to be independent and identically distributed (i.i.d.) Gaussian with zero mean and variance σ^2 (i.e. white Gaussian noise). This case is of particular interest for sensing, reconstruction and target detection in the area of hyperspectral imaging. Hyperspectral images consist of spatial maps of light intensity variation of spectral bands or wavelengths (i.e. measurement of the spectrum of light transmitted or reflected in a scene) [209]. This has high applicability in identifying material properties of a scene as chemical elements have unique spectral signature.

As noted by the authors in [209] it is possible to derive a slightly tighter result without replacing Equation (4.17) for small quantity of noise $\|e\|_{l_2}$ with $E[\|e\|_{l_2}] = \sqrt{M}\sigma$, which will increase the collection of necessary measurements. Specifically, robust sparse recovery can be achieved as an estimate of the form:

$$E[\|\hat{X} - X\|_{l_2}] \leq \epsilon_1' \sqrt{\frac{K \log N}{\beta}} \sigma + \epsilon_2' \frac{\|X - X_K\|_{l_1}}{\sqrt{K}} \tag{4.18}$$

where K represents the number of non-zeros entries or significant coefficients of X, ϵ_1', ϵ_2' are some small absolute constants and $0 < \sigma, \beta < 1$ represent small constants so as C/β satisfies RIP. Note that now we assume that the standard RIP assumption of $\|CX\|_{l_2} \approx \|X\|_{l_2}$ has been replaced with a a more relaxed assumption, namely that $\|CX\|_{l_2} \approx \beta\|X\|_{l_2}$. Obviously, either an increase in β or (an equivalent) decrease in σ will improve the estimation of X. In fact, it has been shown in [50] that for any Sensing matrix C which might not satisfy RIP but with the same l_2 norm, any sparse recovery algorithm cannot improve the estimate of X more than a constant factor. This means that for any fixed set of random linear measurements, under the presence of noise, no recovery system can achieve better than standard sparse recovery methods. A more thorough analysis about sparse models, non-negative sensing matrices and imperfect system models can be found at [209].

An alternative recovery guarantee for bounded noisy cases which also considers mutual coherence $\mu(C)$ of Sensing matrix C is given in [76, 85]:

$$\|X - \hat{X}\|_{l_2} \leq \frac{\|e\|_{l_2} + \epsilon}{\sqrt{1 - \mu(C)(4K - 1)}}, \tag{4.19}$$

where support (sparsity level) $K \leq (1/\mu(C) + 1)/4$ and $\epsilon > \|e\|_{l_2}$ is a small constant. Note that this stability of the solution is for the l_1 norm based problem. Also in [198] the solution of the l_1 norm problem is shown to be unique with probability at least $1 - 2\delta_K - (K/2)^{-5}$ (δ_K RIP constant) given that $X \in \sum_K$, with sparsity K drawn at random from $\binom{N}{K}$ available options with i.i.d. entries generated and $\sqrt{18 \log(N/\delta_K) \log(K/2 + 1)\epsilon} + 2K/N\|C\|_{l_2} \leq \exp(-1/4)(1 - 2^{-1/2})$, for some small constant $\epsilon \geq 1$, $0 < \delta_K < 1/2$, $Y = CX$ and $K \leq \log(N/\delta_K)/(8\mu^2(C))$. The value $\mu(C)$ represents the coherence which has to be small and $\|C\|_{l_2}$ represents the spectral (Euclidean) norm of the Sensing matrix C.

Another class of solution guarantees is provided for greedy algorithms and particularly by OMP methods. For the OMP method with stopping criterion $\|X - \hat{X}\|_{l_2} \leq \|e\|_{l_2}$ and $\|e\|_{l_2} \leq L((1 - \mu(C))(2K - 1))/2$ with $L > 0$ a positive lower bound based on the elements of X, the stability of the solution is given by [76, 85]:

$$\|X - \hat{X}\|_{l_2} \leq \frac{\|e\|_{l_2}}{\sqrt{1 - \mu(C)(K - 1)}} \tag{4.20}$$

In [67] it is shown that OMP can efficiently recover a signal X in exactly K iterations given that $X \in \sum_K$ sampled using $Y = CX$ with C satisfying RIP constant $\delta_{K+1} < 1/3\sqrt{K}$. In fact in [217] it has been shown that OMP has error recovery bound of $\|X - \hat{X}\|_{l_2} \leq \epsilon\|e\|_{l_2}$, after $30K$ iterations under noise level e ($Y = CX + e$) and a small constant $\epsilon > 0$, given that the sampling process of $X \in \sum_K$ follows RIP constant $\delta_{31K} < 1/3$. In general, the authors in [85] have shown that for every $X \in \sum_K$ and $Y = CX$, sparse recovery methods (OMP, AIHT and BP methods) can recover almost exactly X from Y given that C follows RIP with constant $\delta_{\epsilon K}$ where ϵ and δ are method specific. Note also that recovery is actually proportional to the noise level in all the above cases.

4.5 Design of CS matrices

The design of Sensing matrices is very important in CS theory so as to guarantee the uniqueness of solution from the given under-sampled measurements. In general, ill-posed inverse problems do not have unique sparse solutions [96, 142]. The number of sparse solutions can range from $\binom{N-1}{M} + 1$ to $\binom{N}{M}$ [28, 96, 142]. However, the design of the Sensing matrix can guarantee the uniqueness of the sparsest solution based on the following definition and theorem (Unique Representation Property - URP).

Definition 9:[56, 96] *A system is said to have a unique sparse solution if any collection of $M = 2K$ columns of C are linearly independent. This guarantees that any basis is uniquely represented by the columns of C and thus X is the unique solution of $CX = Y$ having support/sparsity $X \in \sum_K$ ($\|X\|_{l_0} = K$) according to the following theorem.*

Theorem 12: [56, 96] *If a linear system $CX = Y$ satisfies URP and has $K < M/2$ dimensional solution (sparsity) then there can be no other solution with dimension less than $R = M - K + 1$.*

This property particularly holds for Gaussian Sensing matrices C given that $M \geq 2K$. This means that the maximally sparse solution has dimension less than $M/2$ or in other words the solution is guaranteed to be unique when the measurements M exceed the signal dimension (length) by a factor of 2. Note however that the sampling process does not necessarily need to be uniform. A very relevant theorem which proves the uniqueness of the sparsest solution is based on spark.

Definition 10:[79, 85] *The spark(C) of a given matrix C is the smallest number of columns of C that are linearly dependent.*

Note that $M \times N$ matrices C with i.i.d. random entries generated by a continuous distribution have spark(C) $= M + 1$ with high probability, while as M and N grow the coherence converges to $\mu(C) = 2\sqrt{\log N/M}$ (for distributions with zero mean and finite variance, such as random matrices from Gaussian, Rademacher, etc.)[79, 85]. The following theorem guar-

antees the uniqueness of the sparsest solution using spark.

Theorem 13: **[79, 85]** *If spark*$(C) > 2K$ *then for every measurement vector* $Y \in \mathbb{R}^M$ *there exists exactly one vector* $X \in \sum_K$ *such that* $Y = CX$.

This means that spark$(C) \in [2, M+1]$ so that this theorem holds for measurements $M \geq 2K$ [85]. Note however, that finding spark of a matrix has high computational complexity which is the reason that mutual coherence $\mu(C)$ property is chosen instead [85]. Note that spark and mutual coherence can be related as follows:

Lemma 3:[**79, 85**] *For any matrix* C, *spark*$(C) \geq 1+\frac{1}{\mu(C)}$, *where* $\mu(C) = \max_{1\leq i\neq j\leq N} \frac{|<C_i,C_j>|}{\|C_i\|_{l_2}\|C_j\|_{l_2}}$, *is the mutual coherence of Sensing matrix* C.

Note that since C is orthonormal, $\|C_i\|_{l_2} = \|C_j\|_{l_2} = 1$. Combining now the previous Theorem and Lemma we can also define the following condition for C so as to guarantee uniqueness of the sparsest solution.

Theorem 14: **[79, 85, 197]** *If* $K < \frac{1}{2}(1+\frac{1}{\mu(C)})$ *then for every measurements vector* $Y \in \mathbb{R}^M$ *there exists exactly one vector* $X \in \sum_K$ *such that* $Y = CX$.

This provides the theoretical upper bound on the sparsity level in terms of coherence ($K = O(\sqrt{M})$ while the lower bound of $\mu(C)$ is $\Omega(1/\sqrt{M})$) on condition that Y is produced without error (noiseless environment) [85]. We can also connect RIP with the mutual coherence as follows:

Lemma 4:[**36, 46**] *If* C *has unit-norm columns and coherence* $\mu(C)$ *then* C *has RIP constant* $\delta_K \leq (K-1)\mu(C)$, *which means that if RIP constant* δ_{2K} *is satisfied for* C *matrix and thus spark*$(C) > 2K$, *then any vector* $X \in \sum_K$ *can be uniquely recovered.*

Note that for each integer $S = 1, 2, \ldots$ we can define a slightly different constant δ_{SK} of the Sensing matrix C as the smallest number such that RIP holds for all K-sparse vectors (i.e. K is the order of RIP). Based on this assumption we say that a matrix C obeys the RIP

of order K if δ_K is not too close to one. Under this condition C matrix can guarantee that all K-sparse vectors cannot be in the null space of C, as C must be a distance preserving transformation for all K-sparse vectors sampled and bounded by some constant δ_{SK}, known as the RIP constant (i.e. all subsets of C columns taken from C are in fact nearly orthogonal) [39, 43, 46]. In the other case of $S = 2$ we assume that δ_{2K} is sufficiently less than one. If this RIP constant property does not hold then it is probable for a K-sparse vector to be in the null space of C and thus it may be impossible to efficiently recover it.

4.6 Measurement bounds

If we ignore the δ parameter used in RIP we can then establish the lower bound needed for the number of measurements M, using K as the level of sparsity and N as the length of the measured vector. We firstly need to provide the following Lemma:

Lemma 5:[41, 68, 87] *Let K and N satisfying $K < N/2$ be given. Then there exists a set $X \subset \sum_K$ such that for any $x \in X$ we have $\|x\|_{l_2} =\leq \sqrt{K}$ and for any $x, z \in X$ with $x \neq z$,*

$$\|x - z\|_{l_2} \geq \sqrt{K/2}, \tag{4.21}$$

and

$$\log |X| \geq K/2 \log(N/K). \tag{4.22}$$

Based on this Lemma we can now define the bound on the required number of measurements to satisfy the RIP property.

Theorem 15: [40, 41, 43] *Let C be an $M \times N$ matrix that satisfies the RIP property with constant δ and order $2K$, $\delta_{2K} \in (0, 1/2]$. Then*

$$M \geq O(\epsilon K \log(N/K)), \tag{4.23}$$

with probability of failure $O(N^{-\beta})$ (for a small constant $\beta > 0$) and $\epsilon = 1/2 \log(\sqrt{24}+1) \approx 0.28$. Note also that the restriction $\delta_{2K} \in (0, 1/2]$ is merely for simplicity reasons and

we can establish bounds for any arbitrary value of $\delta_{2K} < 1$. A more detailed analysis in optimal/improved bounds can be found in [68], which investigates different bounds for measurements. For a pair of bases Φ and Ψ (e.g. bio-orthogonality, see Appendix B) following RIP property and a Sensing matrix $C = R\Psi\Phi$, where R extracts M elements randomly, Candès and Tao showed that a small recovery error holds with overwhelming probability if the following measurements bound is satisfied [39, 44]:

$$M \geq O(\epsilon K(\log N)^5) \tag{4.24}$$

Rudelson and Vershynin in [171] used geometric functional analysis and probability in Banach spaces in order to prove that it is possible to have a smaller recovery error with a bit smaller probability of failure if the measurement bounds satisfy:

$$M \geq O(\epsilon K(\log N)^4), \tag{4.25}$$

Note however, that no formal result has been established in a general case of incoherent measurements, without (K, δ)RIP conditions, for $M = O(\epsilon K(\log N))$, case which is only conjectured to hold.

It has also been shown in [12, 85] that Sensing matrices which are random and satisfy (K, δ)RIP with high probability require $M = O(K \log(N/K)/\delta^2)$ measurements. DeVore in [70] showed that it is also possible to create Sensing matrices deterministically which satisfy again (K, δ)RIP for $K = O(\sqrt{M} \log(M)/\log(N/M))$ and $M \approx O(K^2 \log N)$ which is more restrictive for real life applications. In more general cases where C is not known whether it satisfies RIP/UUP properties, the number of measurements needed for efficient recover is [41, 83, 85]:

$$M \geq \ell K N \mu^2(\Phi, \Psi) \log(N/\delta), \tag{4.26}$$

or alternatively

$$M \geq \ell' \log^2(N/\delta), \tag{4.27}$$

with probability of success $1-\delta$, for small $\delta < 0$ and $0 < \ell, \ell' < 1$ some small constants. Then

$W(X)$ (signal coefficients) is the solution to $\Phi\Psi W(X) = Y$, assuming that $W(X)\Psi = X$ ($X \in \sum_K$ in Ψ) has support $\Omega \subset \{1, 2, \ldots, N\}$ ($|\Omega| = K$) and $\Phi^{M\times N}$ is used to draw from $W(X)$ uniform and random measurements Y as a subset $A \subset \{1, 2, \ldots, N\}$ with $|A| = M$. Note, μ is the mutual coherence of the column rows of the orthonormal bases Φ_i, Ψ_j, with $\mu(\Phi, \Psi) = \max_{1<i\neq j<N} | < \Phi_i, \Psi_j > |$. Since $\mu(\Phi, \Psi) \in [\sqrt{N}, 1]$, the measurements M number ranges from $O(K \log(N))$ to $O(N)$.

4.7 Alternative recovery problems

As we have already discussed in this Section sparse recovery is achieved by solving the l_0-norm based problem defined in Equation (4.1). This is a minimisation problem of non-zero coefficients of the sparse signal (defined as vector) we want to recover or decompress. An alternative approach is to solve the l_0 norm based unconstrained optimization problem, than trying to solve the constrained version of the same problem, which has been effectively used in several similar signal recovery problems; see for example [48, 83, 141, 159, 161]. This equivalent problem formulation without constraints is given as [48, 87, 142]:

$$\min_{\hat{X}} \ \|Y_{M\times 1} - C_{M\times N}\hat{X}_{N\times 1}\|_{l_0}, \tag{4.28}$$

where Y is the samples vector, C is the Sensing matrix and $\hat{X}$ is the unknown sparse vector. This problem formulation has some similarities with the error correction and coding theory problems [39, 48, 87].

It has also been proven that the problem in (4.28) yields the same conditions for optimality as its counterpart in (4.1) and derives the same stationary points as solving the equivalent constrained problem [6, 141]. Its major advantage is the elimination of the constraints which usually grow very quickly in size, aspect which makes the problem much easier and quicker to solve. However, due to the ill-posed nature of the problem and the Null Space of the Sensing matrix C, any miscalculations that might occur in the estimation of X can be propagated in the solution set, as we have numerous local minima within the feasibility set. In particular, an estimate of X can satisfy $C\hat{X} = Y$, but might yield high error $\|\hat{X} - X\|_{l_2}$, which can be propagated in the next step of the l_0 heuristic proposed in Section 6.6.

Note also that the l_0 norm defined in sparse recovery problems in (4.28) and (4.1) is not

a real norm (not continuous) and thus a combinatorial search is still required in order to find the optimal solution among lots of local minima. In fact, both of these problems are defined as NP-Hard [6, 11, 39, 46]. Therefore, mathematicians and researchers of the CS method initially proposed the l_1-norm problem for efficient sparse recovery. This is a convex (linear) optimization problem which can be solved numerically very efficiently using one of many linear optimization methods such as the well known Simplex method or other interior point methods. However, there are some properties that have to be considered in order to apply the l_1-norm minimisation problem as a good approximation of the l_0-norm minimisation problem. This is achieved with the help of RIP/UUP principles which are already defined in Section 4.3.

4.7.1 The l_1 min problem

It is notable that in the most literature the l_1 convex (approximate) relaxation problem is solved instead of the l_0 problem which in principle requires a combinatorial search. The l_1 minimisation problem can be defined as follows [11, 42, 75, 119]:

$$\min_{\hat{X}} \|\hat{X}_{N\times 1}\|_{l_1} \quad s.t. \quad Y_{M\times 1} = C_{M\times N}\hat{X}_{N\times 1}, \tag{4.29}$$

where, $\hat{X}$ is the unknown sparse vector, Y are the measurements or samples and C is the Sensing Matrix used for the collection of these samples. For real-valued C and $\hat{X}$ this is a linear programming problem (LP), which can be solved by gradient based methods or Simplex variations. For complex valued C and $\hat{X}$ this is a second-order cone programming problem (SOCP) which can be solved by gradient based methods [28, 132, 215]. This l_1 minimisation problem is a linear problem and it is equivalent to the l_0 minimisation problem if the Restricted Isometry Property, previously discussed, holds. Then, we can reconstruct a given signal or image, exactly, with overwhelming probability, as it is given in Sections 3.5 and 4.4, using $M \approx K \log N$ measurements or samples [39, 46]. In general, many researchers follow this approach as a sparse approximation scheme because it has both linear constrains and objective function and thus it can be easily solved by many state-of-art algorithms including the method of the famous American mathematician George Dantzig, commonly known as Simplex method.

Other well-known variations of the l_1 norm based problem are as follows:

1. The quadratic relaxation, least squares approximation with l_1 regularisation or Least Absolute Shrinkage and Selection Operator (LASSO) [28, 29, 193]:

$$\min_{\hat{X}} \|\hat{X}\|_{l_1} \quad s.t. \quad \|C\hat{X} - Y\|_{l_2} \leq \lambda \tag{4.30}$$

or alternatively:

$$\min_{\hat{X}} \|C\hat{X} - Y\|_{l_2} \quad s.t. \quad \|\hat{X}\|_{l_1} \leq \lambda, \tag{4.31}$$

where the presence of l_1 enforces the components of $\hat{X}$ to become close to zero (i.e. encourages sparsity) and $\lambda > 0$ is a non-negative real parameter based on the feasible nature of the problem. Note that this is a non-linear problem.

2. The Dantzig selector or residual correlation constrains problem which is a linear programming problem [39, 49]:

$$\min_{\hat{X}} \|\hat{X}\|_{l_1} \quad s.t. \quad \|C^T(C\hat{X} - Y)\|_{\infty} \leq \lambda, \tag{4.32}$$

which solves the sparse recovery problem by projecting back to the vector of residuals $(C^T(C\hat{X} - Y))$, with $\lambda > 0$ a non-negative real parameter based on the feasible nature of the problem. Note that this is an efficient estimation within the predicted noise level (assuming noise $0 \leq \|e\|_{l_2} \leq 1$), but does not correlate well with the Sensing matrix C [39].

3. The l_1 analysis in an over-complete framework (basis C is over-complete) can be defined as [51, 86, 215]:

$$\min_{\hat{X}} \|C^T\hat{X}\|_{l_1} \quad s.t. \quad \|C\hat{X} - Y\|_{l_2} \leq \lambda, \tag{4.33}$$

which is a minimisation of an over-complete dictionary as a sum of dictionary subsets (e.g. Wavelets, DCT and DFT). This type of problem commonly arises in image/signal reconstruction and restoration where C represents the observation operator and redundant dictionary or basis and X is the vector of representation coefficients of an unknown signal/image.

4.7.2 The l_2 min problem

The Least Mean Squares (LMS) algorithm or l_2-norm minimisation method is one of the most popular recursive parameter estimation methods and thus the most obvious approach someone might assume for sparse recovery. However, in its standard form it does not consider any special characteristics of the measurements and sampled vector as a parameterized model. If such model is sparse in some domain (for example, it has sparse impulse or frequency response), the method is not adapted properly for this underlying sparsity and is solely based on the number of measurements which are highly incomplete and thus achieving poor results. It has been shown in literature that this approach, as a process of sparse recovery cannot find an efficient minimum l_2-norm solution [42, 43, 75] (see also example in Section 3.6). This is expected due to the ill-posedness of the problem and its numerous local minima [119, 203].

A more efficient approach is the weighted least squares (weighted norm approximation) or regression estimation method as an optimisation principle which tries to minimise the weighted difference between the observations vector Y and the estimation model CX, defined as follows [28, 56, 188]:

$$\min_{\hat{X}} \ \|W(Y_{M\times 1} - C_{M\times N}\hat{X}_{N\times 1})\|_{l_2}, \tag{4.34}$$

where $W \in \mathbb{R}^{M\times M}$ is a weighting matrix, which is usually diagonal so as to give relatively large weights to the small residuals ($R = C\hat{X} - Y$) of the model parameter estimation and thus leading to optimal solutions. Unlike ordinary least squares approach each term is weighted so as to determine the significance of each estimation X over the observations Y in the data set.

At this point two important issues have to be pinpointed. Firstly, that recently in the literature this weighted approach has received considerable attention in CS theory as an alternative to l_1 norm problem with good quality parameter estimates. Further details on this technique and relevant solution methods are discussed in Sections 7.2 and 7.4. Secondly, if the Sensing matrix C can be inverted and has dimensions $N \times N$ (same as the vector -

signal) the unweighted least squares problem has an analytical solution [28, 163, 188]:

$$X = (C^T C)^{-1} C^T Y \tag{4.35}$$

The latter outcome is crucial, as we will see, since based on this result we can create a pseudo-inverse of the matrix C instead, so as to generate an efficient initial feasible solution for the proposed heuristic described in Section 6. Alternatively, the objective function in Equation (4.34) can be replaced by a global smoothness function $s_p(f)$ used in many interpolating methods in images [33, 163]:

$$s_p(f) := \frac{1}{p} \sum_{i,j} \|\nabla_{i,j} f\|^{p/2} = \frac{1}{p} \sum_{i,j} [w_{ij}[f(j) - f(i)]^2]^{p/2}, \tag{4.36}$$

where $f(j) - f(i)$ represents the difference of light intensity of pixel j and i respectively in rows and columns for all pixels of an image (i.e. minimise the expectation error between all the pixels of the image). The weight w_{ij} for pixels i, j is a small positive number based on the image (e.g. the average of pixel intensity), while p parameter defined the norm used in the calculations. For $p = 1$ we have the TV norm defined in Equation (3.26), while for $p = 2$ we have:

$$s_2(f) := \frac{1}{2} \sum_{i,j}^{N} w_{ij}[f(j) - f(i)]^2 \tag{4.37}$$

A similar approach where we want to choose the subset of K potential regressors (columns of C) and the associated values of $\hat{X}$ with $K \ll M \ll N$ can be expressed as [28, 29, 193]:

$$\min_{\hat{X}} \ \|C\hat{X} - Y\|_{l_2} \quad s.t. \quad \sup(\hat{X}) \leq K, \tag{4.38}$$

where $\sup(\hat{X})$ measures the support/sparsity of a vector which is simply the number of non-zero entries. This is a useful optimisation principle with many variations, as a measure of error between the possibly noisy samples Y and the estimate $\hat{X}$. However, this is very difficult to optimise so as to find the sparsest solution. In fact, this is NP-hard [28], which is expected as we need to check every possible combination of pattern in $\hat{X}$ with sparsity K. This check of all the sub-matrices of C that correspond to all the possible sparsity patterns

is $\binom{N}{K} = \frac{N!}{K!(N-K)!}$ [28].

On a practical aspect, another similar and very straightforward approach is the classic regularisation least squares minimisation with penalised criterion, using both the quadratic error term and the sparsity regularisation term defined as [25, 26, 196, 218]:

$$\min_{\hat{X}} \frac{1}{2}\|C\hat{X} - Y\|_{l_2} + T\xi(\hat{X}), \tag{4.39}$$

where C is a dense matrix of corresponding dimension and $\|.\|_{l_2}$ denotes the standard Euclidean (l_2) norm. The term $\|C\hat{X} - Y\|_{l_2}$ is the data fitting term while the T is the non-negative, regularisation term which balances between quality of data fitting and prior information or confidence for the sparsity of the solution. Many regularisation functions $\xi(\hat{X})$ have been designed and proposed in literature, including the l_1 norm $\xi(\hat{X}) = \|\hat{X}\|_{l_1}$, the l_0 norm $\xi(\hat{X}) = \|\hat{X}\|_{l_0}$, or even a strictly convex differentiable penalty function on $\mathbb{R}^+$. Typical examples of such smooth and convex functions, which can accelerate convergence, are $\xi(\hat{X}) = \|X\|_{l_p}$, for $0 < p < 1$ and $\xi(\hat{X}) = \|\upsilon + \hat{X}\|_{l_2}$, for a small constant value $\upsilon \geq 0$, which produces similar results to l_1 norm [25, 26, 59, 218].

This optimisation approach, as a variant of the previous approaches has been successfully applied on many different applications in signal and image processing, such as de-noising, inpainting, de-blurring, source separation and sparse recovery [218]. In fact, due to its popularity and high applicability a modified regularised least squares method has also been introduced, so as to reduce the computational cost of the solution from Equation (4.39) [25, 26, 218]:

$$\min_{\hat{X}} \frac{1}{2}\|U^T U\hat{X} - U^T Y\|_{l_2} + T\xi(\hat{X}), \tag{4.40}$$

where U matrix is computed directly from a dictionary Ψ (e.g. Fourier) by randomly taking N rows and $2K+1$ columns, while $U^T U$ can be computed by randomly taking a $N \times 4K+2$ matrix from Ψ, with K the sparsity level of $\hat{X}$ and N the length of $\hat{X}$ [25, 26]. Note also that the relation $UU^T = 2KI_N$ holds for Fourier matrices, with I_N the identity matrix of N elements, while some scalar functions can also be used for the calculations when U is unitary (since the l_2 norm is unitary invariant) [218].

4.7.3 The l_0 approximation min problem

Recently the l_0 norm based problem (Equation (4.1) of this chapter) has attracted considerable research as an alternative to l_1 and l_2 minimisation problems. Smoothed (approximated) l_0 norm minimisation problems have been studied in the optimization/signal processing society as an efficient and alternative way of sparse recovery. A class of smooth regularisation functions $f(X, \sigma)$ have been proposed for this purpose as a strategy to directly approximate the l_0 norm. Two of the main families of such functions are very well-known [60, 110, 142, 151, 183]:

- Convex, continuously differentiable functions, asymptotically linear functions with quadratic behaviour near zero. Typical example in this category are the functions $f(X, \sigma) = \sqrt{X^2 + \sigma^2}$ and $f(X, \sigma) = \sigma \ln(1 + \|X\|/\sigma)$, for every $X \in \mathbb{R}^N$ and parameter $\sigma \in \mathbb{R}^+$.

- Asymptotically constant functions with a quadratic behaviour near zero. Typical examples are truncated quadratic functions $f(X, \sigma) = X^2/(2\sigma^2 + X^2)$ and $f(X, \sigma) = \exp(-X^2/2\sigma^2)$, for every $X \in \mathbb{R}^N$ and parameter $\sigma \in \mathbb{R}^+$, used by the l_0 heuristic (discussed in Chapter 6) and the variation of Steepest Ascent method (discussed in Section 7.5).

In this case the l_0 norm based problem discussed in Equation (4.1) can be reformulated as a penalised regularised optimisation problem as [60, 151]:

$$\min_{\hat{X}} \|C\hat{X} - Y\|_{l_2} + h(\hat{X}, \sigma), \tag{4.41}$$

where $C \in \mathbb{R}^{M \times N}$ is the non-zero sampling matrix, $Y \in \mathbb{R}^M$ is a possibly corrupted (noisy) samples vector, while $h : \mathbb{R}^N \to \mathbb{R}$ is a regularisation function with $\sigma \in \mathbb{R}^+$ a positive parameter (with σ depending on the level of noise) of the following form [60, 151]:

$$h(\hat{X}, \sigma) = f(\hat{X}, \sigma) + \|V(\hat{X})\|_{l_2}, \tag{4.42}$$

where $f(\hat{X}, \sigma)$ is a class of smooth regularisation functions discussed earlier while $V \in \mathbb{R}^{P \times N}$ is a regularisation vector with either $P = 1$ or $P = 2$ (depending on the problem - one or

two dimensions), which is usually the discrete gradient (horizontal or vertical), similar to total variation ($TV(\hat{X})$) defined in Section 8.1:

$$V(\hat{X}) = \sum_{i=1}^{N} \sqrt{(D_i\hat{X})^2} = \sum_{i=1}^{N} \|D_i\hat{X}\|_{l_2} \tag{4.43}$$

$$D_i\hat{X} = \begin{cases} \hat{X}_{i+1} - \hat{X}_i & i < N \\ 0 & i = N \end{cases} \tag{4.44}$$

Note that this efficient minimisation principle has also been used in other similar to sparse recovery problems, namely image de-noising, segmentation, de-blurring and reconstruction. A more detailed treatment, including numerical experiments and asymptotic analysis of this optimisation principle can be found in [60, 151].

Another slightly different approach which solves a Lagrangian approximation of the original l_0 norm based constrained problem, defined in Equation 4.1, has been introduced in [110]. Efficient sparse vector recovery is now achieved by using the following Lagrangian function defined as [110]:

$$\arg \min_{\hat{X}} f(\hat{X}, \sigma) + \nu'(C\hat{X} - Y), \tag{4.45}$$

where $C \in \mathbb{R}^{M\times N}$ is the sampling matrix, $Y \in \mathbb{R}^M$ is the samples vector, $f(\hat{X}, \sigma)$ is an asymptotically constant family of functions with a quadratic behaviour near zero (using parameter σ), defined earlier, while $\nu' \in \mathbb{R}^{M\times 1}$ is the vector of Lagrange multipliers. Due to the non-linearity of the problem in (4.45) the authors have proposed a function which can be used for the evaluation of solutions of the same problem defined as [110]:

$$F(\hat{X}, \rho') = \rho' g(\hat{X}) + (1 - \rho')\hat{X}, \tag{4.46}$$

where $0 < \rho' \leq 1$ is a small parameter, while the mapping function $g : \mathbb{R}^N \to \mathbb{R}^N$ is defined as [110]:

$$g(\hat{X}) = W^{-1}(\hat{X})C[CW^{-1}(\hat{X})C]^{-1}Y, \tag{4.47}$$

so as $Cg(X) = Y$ and $g(X) \neq X$, while $W(\hat{X}) = \text{diag}\{f(\hat{X}_1, \sigma) \dots f(\hat{X}_N, \sigma)\}$ and X_i the i-th element of vector $X \in \mathbb{R}^N$. Further details about this optimisation principle, including a proposed solution algorithm and performance analysis, can be found in [110].

Within the same framework of the approximating l_0 norm based regularised least squares problem, researchers in [183] have presented a similar non-convex continuous penalty approximation function. The l_0 norm is replaced by a continuous, smoother penalty $l_2 - l_0$ function which is much easier to minimise as follows [183]:

$$\min_{\hat{X}} \frac{1}{2}\|C\hat{X} - Y\|_{l_2} + \bar{\phi}(C, \hat{X}, \ell), \tag{4.48}$$

where $C \in \mathbb{R}^{M \times N}$ is the non-zero sampling matrix, $Y \in \mathbb{R}^M$ is a possibly corrupted (noisy) samples vector and $\bar{\phi}(C, \hat{X}, \ell)$ is a continuous exact l_0 approximation penalty function defined as:

$$\bar{\phi}(C, \hat{X}, \ell) = \sum_{i=1}^{N} \ell - \frac{\|C_i\|_{l_2}}{2} \left(\|X_i\|_{l_1} - \frac{\sqrt{2\ell}}{\|C_i\|_{l_2}} \right)^2 \mathbb{1}_{\left\{ \|X_i\|_{l_1} \leq \frac{\sqrt{2\ell}}{\|C_i\|_{l_2}} \right\}}, \tag{4.49}$$

where $\ell > 0$ is a small positive parameter, X_i and C_i is the i-th element of X and i-th column of C respectively, while $\mathbb{1}$ denotes an indicator function defined as:

$$\mathbb{1}_{\left\{ \|X_i\|_{l_1} \leq \frac{\sqrt{2\ell}}{\|C_i\|_{l_2}} \right\}} = \begin{cases} 1 & \text{if } \|X_i\|_{l_1} \leq \frac{\sqrt{2\ell}}{\|C_i\|_{l_2}} \\ 0 & \text{otherwise} \end{cases} \tag{4.50}$$

It has been proven in [183] that this principle of sparse recovery produces sparse solutions as global minimisers which can be easily obtained. It has also been proven in [183] that the set of global minimisers of problem (4.48) is non-empty and is also included in the set of global minimisers of the l_0 norm problem 4.1. Conversely, being a local minimiser of problem (4.48) is a necessary condition to be a global optimum of the l_0 norm based sparse recovery problem. For a more detailed analysis of the global minimisers of the aforementioned optimisation principle, including theoretical guarantees, see [183].

Chapter 5

Noisy or perturbed environments

5.1 Noise in signals

In practice every sampling process in signals or images has to deal with the problem of noise, since every sample acquired in real world is susceptible to noise, even when sampled at quite small time intervals. This white noise, as it is called, can be approximated usually as a purely random process in discrete time intervals. The model that gives an adequate description to this case is as follows [32, 205]:

$$OBSERVATION/SAMPLE \quad = \quad SIGNAL \quad + \quad NOISE \tag{5.1}$$

where, *SIGNAL* is a sinusoidal signal with known frequencies and amplitudes while *NOISE* is a purely random error term. The key step for this equation to hold is that the signal is a linear combination of a set of variables called atoms, which is sometimes called state vector at time t. By this way the signal is sparse and thus compressible while the measurements are purely randomly generated [39, 40, 43, 76]. Moreover, noise is also assumed as a purely random process which can be defined as a sequence of independent and identically distributed (i.i.d.) random variables [40, 43, 76]. Sometimes these variables are uncorrelated to each other rather than independent, as these variables usually follow the normal (Gaussian) distribution where the zero correlation implies independence [57, 120]. This section aims to introduce the concept of noise occurred in sampling and the CS theory as an important step of handling or eliminating it. As we will see CS method is very efficient in noisy environments as a method

for signal processing and reconstruction, which provides optimal estimates of signals and images [39, 43, 76, 105].

5.1.1 Problem definition

So far we have discussed how the CS method works in noise-free environments. This section describes a more recent approach for noisy or disturbed environments. A great deal of scientific research has focused in the last 10 years for solving the noiseless model or system of equations:

$$Y_{M\times 1} = C_{M\times N} X_{N\times 1} \tag{5.2}$$

where Y is the observation or samples vector, C is the full-rank measurement matrix or system model (with $M \leq N$) and X is the original sparse (or almost sparse) signal represented under an appropriate fixed basis Ψ.

Recently researchers have included an additive noise (e) in this model so as it can be more realistic with real life applications. The noisy or perturbed model or system of equations can now be defined as follows [39, 43, 76, 105]:

$$\hat{Y}_{M\times 1} = C_{M\times N} X_{N\times 1} + e_{M\times 1}, \tag{5.3}$$

where e is the noise, typically modeling simple errors in the sampling process which are stochastic and not correlated with signal vector X. This for instance might represent the digitisation (quantification) of CX, storage or transmission errors (noisy channel), malfunctions during the transmission, background/environment noise, or even unavoidable noise from sampling mechanisms.

Most of the time for simplicity, this noise is assumed to be independent and identically distributed based on the Gaussian distribution. For example, a well-known randomly generated noise model is the additive white Gaussian noise $e \sim \mathcal{N}(0, \sigma^2)$ [28, 163]. Beyond additive noise is the multiplicative stochastic noise ($\hat{Y} = C \times X \times \epsilon$) where ϵ may follow Gamma distribution, Poisson distribution or a combination of them. However, we will not consider such cases in this thesis as this may require more sophisticated complexity based regularisation and Bayesian estimation methods which fall out of the scope of this thesis.

Instead, we will only consider a bounded-norm noise model generated randomly; fact which is also considered widely in literature for comparison purposes for measuring the performance of the sparse recovery methods. See for example, [13, 42, 45, 76, 141].

5.1.2 Solution formulation

In order to recover this vector X which represents a digital image or signal from incomplete and noisy (contaminated) observations or measurements we have to solve the following relaxed l_1 minimisation problem [42, 43, 76, 105]:

$$\min_{\hat{X}} \ \|\hat{X}^T_{1\times N}\|_{l_1} \quad s.t.\ Y_{M\times 1} = \|C_{M\times N}\hat{X}_{N\times 1}\|_{l_2} \leq \epsilon, \tag{5.4}$$

or alternatively in an over-complete dictionary Ψ,

$$\min_{W(\hat{X})} \ \|W(\hat{X})\|_{l_1} \quad s.t. \|Y - \Phi\Psi W(\hat{X})\|_{l_2} \leq \epsilon \tag{5.5}$$

where $CX = \Phi\Psi W(X)$, $W(X) = \Psi^T X$ (due to orthogonality of Ψ) and ϵ is a small constant which represents the size of the error term e (upper bound on the size of the noisy distribution). Again if C obeys the UUP or RIP (with unit-normed columns) and if the vector X is (approximately) sparse, then the solution $\hat{X}$ of the previous optimisation problem is within the following noise level (i.e. the recovery error is of the same order as the observation error)[43, 76, 105]:

$$\|\hat{X} - X\|_{l_2} \leq C\epsilon \tag{5.6}$$

It is noteworthy that if C is a Gaussian random matrix, the recovery is stable and occurs for almost all different combinations of the elements in C, provided that the number of non-zero elements of the original vector X is approximately the same order as the number of observations Y. In case of few Fourier samples of vector X, the recovery is also stable for almost any set of coefficients, provided that the number of non-zeros is of order $N/(\log M)^6$ [43]. Of course, in case of noiseless measurements or samples, the error term ϵ vanishes and the recovery is almost exact as already discussed in Sections 4.7.1 and 4.7.2.

5.2 Noise in images

Image noise is the outcome of errors in the digital image acquisition process that results in pixel values that do not represent the real light intensities of an image scene (degradation of image values) [111, 182]. Noise can be produced by the sensor or circuitry of a scanner/digital camera, as an undesirable by-product during an image capture that adds unnecessary information, or during transmission or processing which may be dependent or independent of actual image content [111, 182, 195]. This type of information changes both the structure and the model of the image. Darker areas are more affected in contrast to brighter areas, resulting in a higher overall SNR (signal to noise ratio, for details see Section 9.3) [182, 210]. These fluctuations in color and light intensity (which can vary significantly from one model to another) can considerably degrade the image quality, which can be accessed by two different methods; subjective and objective. Subjective methods usually apply a list of quality criteria to award appropriate marks for the image quality, while objective methods measure image quality by comparison with a already known reference image using simple distance measures, such as the Euclidean distance, maximal absolute difference and correlation between the pixels of the two images [182, 204]. In Section 9.3, Chapter 9 we will briefly describe the recovery error metrics used in Experiments.

In general, digital images are prone to a variety of types of noise, depending on how they were created. This includes: Gaussian noise (additive noise), salt-and-pepper noise (analog-to-digital converter errors), impulse noise (individual noisy pixels), shot noise (fluctuations in light intensity using Poisson distribution), quantisation noise (quantisation in the pixels of a sensed image to a number of discrete levels), multiplicative noise (signal magnitude noise due to degeneration, which is maximal within the same line of image) and general sensor related noise (noise in sensors from digital cameras) [111, 182, 210, 212]. In this thesis we will consider the first two types of noise (and a combination of them) which are the most common ones. Notice also that noise fluctuations can also vary in both their magnitude and spatial frequency, and that complete elimination of a relatively high amount of noise could result in unnatural looking images. In this thesis, we will not consider spatial frequency for noise, which has the same intensity throughout the image (all noise frequencies are present which simplifies the calculations). A typical data model in vector format for blurring images

and then corrupting with random (Gaussian) noise can be defined as follows [111, 182, 212]:

$$X' = kX + e, \tag{5.7}$$

where e is an additive Gaussian noise, sometimes called white (idealised) noise, k is a linear blurring operator and lowpass ($K \approx 1$) and X the original image. This is an image independent degradation which can often be used for testing and comparison purposes. Other useful yet more sophisticated algorithms for noise generation can be found in [111, 182, 210, 212].

Chapter 6

Heuristics for the l_0-norm problem

6.1 Problem definition (revisited)

As we have seen so far, the vector sparse recovery problem can be written in a general form as follows:

$$\textit{Find a vector } X \quad s.t. \quad CX = Y \tag{6.1}$$

where X is the unknown coefficients or the signal vector and Y the samples vector. Matrix $C = \Phi\Psi$, Φ is the measurement or sampling operator (matrix) and Ψ is the sparcifying operator, such as Fourier spikes or Wavelets Basis [132, 186, 215]. In general, C represents the Sensing matrix with or without the Base Ψ. It is usually a random matrix with values between 0 and 1, generated randomly following the Gaussian distribution. In fact, there is no formal difference between Φ and Ψ. In theory, the former refers to the dictionary of physical spectra and the later refers to the dictionary of signal/image waveforms. In practice, Ψ is a partial Fourier/Wavelet matrix obtained by selecting M rows (i.e. measurements) uniformly at random, using for example a Gaussian distribution, and then re-normalising the columns so that they are unit-normed (see for example [42, 75]).

CS framework is very advantageous in signals/images which are sparse (have only a few non-zero entries) in a known basis provided that the measurements collected are incoherent (i.e. random) [39, 41, 46]. Since we are interested in sparsely represented highly under-sampled signal/images, the linear system describing the measurements is under-determined

and therefore has infinitely many solutions [39, 42, 75]. Furthermore, in most cases the matrix C is singular (its rank is less than n), its inverse does not exist and therefore the equation $X = C^{-1}Y$ does not have a solution [39, 47, 75]. This problem is also NP-Hard which constitutes impossible to apply a straightforward greedy algorithm so as to go through all possible solutions on the basis of provided information without worrying about the effect they may have in the future [11, 39, 119]. Moreover, we cannot apply an alternative optimization method such as the Simplex method or the Least Squares method since this system of linear equations is ill-posed or ill-conditioned; there are more unknowns (N) than equations (M) [39, 46, 47]. For the same reason we cannot apply an interpolation method for a stable recovery. This instance of an under-determined system of linear equations constitutes a linear inverse ill-posed problem [39, 46, 47].

In order to deal with this problem the researchers Candès, Donoho and Tao introduced the concept of l_0 norm as we have seen. In this case the problem is defined as:

$$\min_{\hat{X}} \|\hat{X}\|_{l_0} \quad s.t. \quad C\hat{X} = Y \tag{6.2}$$

where Y is the samples vector, C is the sensing matrix or the product of the sensing matrix with the basis matrix and $\hat{X}$ is the estimate of the unknown vector X. The norm $\|.\|_{l_0}$ counts the number of non-zero components and thus $\|X\|_{l_0} = K$, where K is the sparsity level (number of non-zero entries) of vector $\hat{X}$. We can easily see that the constraints are linear while the objective function is non-linear. Therefore, it is not possible to use a linear programming method, such as the famous Simplex method, to solve this problem.

6.2 Problem reformulation

We know that the l_0 norm of a vector $X = [X_1, X_2, \ldots, X_N]$ measures the number of non-zero entries of this vector, which is:

$$f(X_i) = 1 \;\text{ if }\; X_i \neq 0 \;\text{ and }\; f(X_i) = 0 \;\text{ otherwise} \tag{6.3}$$

which should be done for every element X_i of vector X. As we can see there are discontinuities of the l_0 norm caused by this function $f(X_i) = \|X_i\|_{l_0}$. Moreover, the l_0 norm based problem in Equation (6.2) is NP-Hard (reduction to Set Covering problem) [11, 39, 46, 75]. A common

approach to overcome these difficulties of the combinatorial search would be to replace the problem by its convex relaxation and particularly by substituting the l_1 norm for the l_0 pseudo-norm (see for example [39, 42, 48]). In this thesis, we follow a different approach. The main idea is to approximate the l_0 norm by a smooth continuous function which is easier to process by the l_0 heuristic and is also less sensitive than l_1 norm to noise [6, 142, 161, 199]:

$$\|X\|_{l_0} \approx f(\|X\|, \sigma) = \sum_{i=1}^{N} 1 - f(\|X_i\|, \sigma) = N - \sum_{i=1}^{N} \exp(-\frac{|X_i|^2}{2\sigma^2}), \tag{6.4}$$

where X_i is the i-th element of vector X of length N and σ is a sequence index parameter. Note that X_i can also represent the real and imaginary part of X, in case $X \in \mathbb{C}^N$ (See Experiments, Chapter 9, for details). The continuous function $f(\|X_i\|, \sigma)$, which belongs to the Gaussian family of functions, smoothly approximates the original l_0 norm and can also be easily differentiated. The σ determines the smoothness and thus the quality of the approximation. It is a decreasing sequence of constants $[\sigma_1, \sigma_2, \ldots, \sigma_j]$ for every iteration of the algorithm so as to minimise the smoothed l_0 norm using a general iterative approximate algorithm.

In general, this l_0-norm function can be viewed as a smooth approximation of the l_0 norm which interpolates the function space between the l_1 and l_0, across the parameter $\sigma \in [0, +\infty)$, so as to simulate the l_p norm for $p \in [0, 1]$ [141, 142]. When $\sigma > 0$, $f(\|X\|, \sigma)$ is continuous and strictly concave which can asymptotically approximate the Kronecker delta function (the shape of the approximation). Also, for some subset of $\mathbb{R}^+$ the function is also inherently subadditive and thus true metric for deriving a unique solution for the sparse recovery problem [108, 142, 199]. Therefore, by starting from any point in a convex feasible region and moving on the trajectory of a function, which is concave on the same region, we can obtain quickly the global minimiser of that region. Note that in cases where a function is concave-continuous only in the vicinity of a global minimum, a recovery method is able to converge to a global minimum, provided that it starts from any point sufficiently closed to the global maximum and there is a suitable decreasing sequence for σ. The geometric decreasing sequence for σ can guarantee the local concavity of the function, while the proper selection of the internal loop of the recovery method can force the asymptotic functional to the global minimum, obtaining the sparsest solution.

In general, the properties of the function $f(\|X\|, \sigma)$ can be summarised as follows [108, 141, 142, 199]:

$$\lim_{\sigma \to 0} f(\|X\|, \sigma) = \begin{cases} 1, & \text{if } X = 0 \\ 0, & \text{if } X \neq 0 \end{cases} \tag{6.5}$$

or alternatively for $\sigma > 0$ (locality),

$$f(\|X\|, \sigma) \approx \begin{cases} 1, & \text{if } |X| \ll \sigma \\ 0, & \text{if } |X| \gg \sigma. \end{cases} \tag{6.6}$$

Also we want in general, $\forall X \in \mathbb{R}$ that $0 \leq f(\|X\|, \sigma) \leq 1$ (non-negativety) and $f(\|X\|, \sigma) = f(\sigma, \|X\|)$ (symmetry). Note that in the previous definitions we define $|X|$ as the absolute value of an element of a vector X.

Consequently, we have $\lim_{\sigma \to 0} f(\|X\|, \sigma) = 1 - f(X)$ and thus by defining $f(\|X\|, \sigma) = \sum_{i=1}^{N} f(\|X_i\|, \sigma)$, we have:

$$\lim_{\sigma \to 0} f(\|X_i\|, \sigma) = \sum_{i=1}^{N} (1 - f(X_i)) = N - \|X\|_{l_0} \tag{6.7}$$

and thus, $\|X\|_{l_0} \approx N - f(\|X_i\|, \sigma)$. In the design of the function we also want for sufficient large values of σ, the minimiser of $f(\|X\|, \sigma)$ subject to $CX = Y$ to be the minimum l_2 norm solution of $CX = Y$, i.e. the solution given by the pseudo-inverse (see Section 6.6.1). It is also important to note that after finding a global minimum of $f(\|X\|, \sigma)$ for some value of σ, we need to choose its next (decreasing) value (i.e. its geometric sequence) so as to guarantee the locally concave search area. Note that in general, we want to design a function that tends to 1 or 0 at infinity. A counter example, which does not follow this principle, is the function: $f(x) = x^2 \exp(-x^8 \sin^2 x)$ which is square integrable without being bounded [141].

Among the advantages of this approach are the robustness of the l_0 norm to noisy samples (by approximation), the number of measurements required, which is much smaller than the ones required by its convex analog (l_1 norm), and less restrictions in the design of Sensing matrices C [92, 141, 161]. In fact, RIP constraints demanding randomised CS matrices with i.i.d. entries from a standard probability distribution are often not feasible or too restrictive

for real life applications, mainly due to the cost of multiplication in high dimensional data [6, 141]. Instead, using the l_0 norm we do not have this restriction in the sampling process and thus we can potentially use different probability density functions as long as we follow the incoherence property (see Experiments with different Sensing matrices in Chapter 9). In other words, we just need to select an orthonormal basis (Φ) which is incoherent with the sparsity basis (Ψ) so as to obtain our measurements. Alternatively we can select uniformly at random a subset of coefficients in a chosen basis Ψ. The optimisation problem now is easier to compute and thus improving the performance of the l_0 heuristic. This approach of using the l_0 norm has attracted considerable research recently with computational experiments suggesting that replacing l_p norm ($p = 1$) with $p < 1$ can achieve equally fast recovery with much fewer measurements, while keeping robustness to noise and low vector sparsity [92, 141, 161, 199, 213]. Then the sparse recovery problem can be reformed as [6, 141, 142]:

$$\min_{\hat{X}} \quad f(\|\hat{X}\|, \sigma) \approx (N - \sum_{i=1}^{N} \exp(-\frac{(\hat{X}_i)^2}{2\sigma^2})), \quad \text{s.t.} \quad Y = C\hat{X} \tag{6.8}$$

where $\hat{X}_i$ is the i-th element of the estimate vector $\hat{X}$ and $f(\|\hat{X}\|, \sigma)$ is a smooth non-linear objective function which approximates the l_0 norm subject to the same constraints as initially defined in Equation (6.2). The σ parameter is a decreasing sequence of values that represents the trade-off between accuracy and the smoothness of the approximation. The smaller the σ, the better an approximation of the $f(\|\hat{X}\|, \sigma)$ to the l_0 norm, while the larger the σ, the smoother an approximation (but worse approximation to l_0 norm). For small values of σ, $f(\|\hat{X}\|, \sigma)$ contains a lot of local minima and thus it is difficult to minimise it. Therefore, we need to set this parameter initially very large so as to make the objective function convex and then gradually decrease it according to the value of the objective function using a variation of a PSO algorithm.

Notice that in Equation (6.8) N largest terms of the original vector X are recovered though some of them are set to 0 (since the vector is sparse). Also, despite the approximation approaches mentioned, efficient recovery of X still requires an under-determined system of linear equations which satisfy the unique sparsest solution (i.e. following the URP and coherence properties). Moreover, although we have a smoother objective function, the complexity of the problem remains non-linear as the new objective function is still non-linear.

However, the objective function now is more useful and easier to compute as it is also differentiable. Note that if X is the coefficients vector ($W(X)$) of a dictionary Ψ, we initially need to derive $W(X)$ from Y and then find the K-sparse compressed vector as $\hat{X}_K = \Psi^{-1}W(X)$, where Ψ^{-1} is the inverse of the transform domain used. This similar smoothed l_0 minimisation can reduce the required samples and achieve the expected reconstruction quality with a given base or dictionary Ψ. In this thesis, we will initially use some elementary and generic vectors for testing and improving the effectiveness of the proposed method (e.g.randomly generated $1D$ discrete time signal $f(t)$ using cosines/sines) and then progress to more complex signals and images, where CS has high applicability. Finally it is also important to remark here that the purpose of this thesis is to test the efficiency of this smooth l_0 norm and its optimisation approach and possibly extend or improve it wherever this is possible. Alternative decreasing sequences of values for the parameter σ, will also be considered for this purpose in the experiments, Chapter 9.

6.3 Convergence conditions for sparse recovery

Although the use of l_p $(0 \leq p < 1)$ semi-norms has been initially dismissed, as convergence conditions to a global minimum cannot be guaranteed, their use through approximating function has recently attracted a lot of attention. This can be attributed due to the fact that RIP based conditions are generally very restrictive in most real cases (impractical for basis design matrices) to hold [142, 199, 213]. In fact, the use of a function (e.g. the function defined in (6.4)) to approximate the l_0 norm problem defined in Equation (6.2) can achieve significantly better sparse recovery solutions under much lower theoretical measurement bounds, as the approximated l_0 norm metric exhibits better desired threshold bounds on any non-zero entry of a computed solution [55, 183, 199]. This is possible as unlike the l_0 norm, the gradient of its approximating function is non-zero, concave and positive over $\mathbb{R}^+$, easily calculated and decreased in practice through standard decent or Newtonian methods (See the previous section for details). Computational results have also shown that sparse recovery based on this norm, can be achieved equally fast with the l_1-norm and it can be more robust to noise and signal non-sparsity [6, 141, 161].

Researchers in [55, 199] proposed the use of l_p $(0 \leq p < 1)$ semi-norms for sparse recovery and showed asymptotic convergence of the l_p norm problems using measurement bounds

from RIP. Consider the generalised sparse recovery problem as follows:

$$\min_{\hat{X}} \|\hat{X}\|_{l_p} \quad s.t. \quad C\hat{X} = Y, \tag{6.9}$$

where $0 \leq p < 1$, then a a sufficient condition for a K sparse vector X is given by [55, 199]:

$$\delta_{2K} < \left[1 + \sqrt{\frac{2M}{K}} \left(\frac{K}{M}\right)^{1/p}\right]^{-1}, \tag{6.10}$$

for any $K \leq M$, where M denotes the number of measurements and $\delta_{2K} \in (0, \sqrt{2} - 1)$ (for details see Section 4.3). Obviously, smaller values for p offer sparse recovery from fewer measurements at the cost of increasing numerical complexity, but the choice of the best value for p providing the optimal tradeoff between the quality of sparse recovery and the number of required measurements remains open [55, 199]. The key motivation for using the l_0 norm through approximation is the lowest possible measurement bound using the minimum possible assumptions in measurements collection (only incoherence). Consequently, the function $f(\|X\|, \sigma)$ defined in (6.4) has been chosen for the approximation of l_0 norm due to its properties (its gradient is positive definite, concave and zero-symmetric [108, 142]) and its popularity for comparison with other methods (sparse method discussed in Section 7.5 [141, 142]). In fact, this function is a simple and efficient measure of sparsity as an approximation of l_0 norm, which can converge to an optimal solution if certain conditions are satisfied. The following theorem provides these conditions for the convergence of the function $f(\|X\|, \sigma)$, based on URP (Definition 9 defined back in Section 4.5).

Theorem 16:[60, 142] *Consider a family of univariate differentiable functions $f(\|X\|, \sigma)$ with parameter $\sigma \in \mathbb{R}^+$ satisfying the conditions:*
1) $\lim_{\sigma \to 0} f(\|X\|, \sigma) = 0$ for all $X > 0$.
2) $f(0, \sigma) = 1$, for all $\sigma \in \mathbb{R}^+$.
3) $0 \leq f(\|X\|, \sigma) \leq 1$ for all $X \geq 0$ and $\sigma \in \mathbb{R}^+$.
4) For every positive constant α and $\beta = 1/N$ there exists $\sigma_0 \in \mathbb{R}^+$ that satisfies $X > \alpha$ and $f(\|X\|, \sigma) < \beta$ for all $\sigma < \sigma_0$.
Now assume all $M \times M$ submatrices of the Sensing matrix $C_{M \times N}$ are invertible based on URP. Also assume $f(\|X\|, \sigma) = \sum_{i=1}^{N} f(\|X_i\|, \sigma)$ with $CX = Y$ and $X^* \in X$ is the unique

sparse solution with $X^\sigma = [X_1^\sigma, X_2^\sigma, \ldots, X_N^\sigma]$ *be the minimiser of* $f(\|X\|, \sigma)$*. Then* $\underset{\sigma\to 0}{X^\sigma} = X^*$*.* Following this theorem and based on URP we also have the following Lemma for $f(\|X\|, \sigma)$.

Lemma 6:[142] *Assume a Sensing matrix* $C = [C_1, C_2, \ldots, C_M] \in \mathbb{R}^{M\times N}$ *(*C_i *is the i-th row) which has all* $M \times M$ *invertible submatrices based on URP. Also if* $N - M$ *elements of* $X \in Null(C)$ *converge to zero then all these elements and* X *will also converge to zero (Recall* $Null(C) = \{X : CX = 0\}$*).*

Some slightly tighter but more generalised conditions for the convergence of an asymptotically constant family of functions with a quadratic behaviour near 0 (including the function $f(\|X\|, \sigma)$) is given below.

Theorem 17:[60] *Assume a Sensing matrix* $C = [C_1, C_2, \ldots, C_M] \in \mathbb{R}^{M\times N}$ *(rows of* C*) with null space defined as* $Null(C) = \{X : CX = 0\}$ *and a family of univariate differentiable functions* $f(\|X\|, \sigma)$ *with decreasing sequence of numbers* $(\sigma_i)_{i\in\mathbb{N}} \in \mathbb{R}^+$ *converging to* 0 *and null space* $Null(f) = \{X : f(\|X\|, \sigma) = 0\}$*. Also assume that:*

1. $\forall\sigma > 0$*,* $f(\|X\|, \sigma)$ *is continuous and concave and takes non-negative values,*
2. $Null(C) \cap Null(f) = \{0\}$*,*
3. $\forall(\sigma_1, \sigma_2) \in (0, +\infty)^2$ *and* $\sigma_1 \leq \sigma_2$ *then* $\forall X\mathbb{R}^N, \quad f(\|X\|, \sigma_1) \geq f(\|X\|, \sigma_2)$*,*
4. $\exists\bar{\omega} \in \mathbb{R}^+$ *such that* $\forall X \in \mathbb{R}^N, \sigma \in \mathbb{R}^+ \quad 0 \leq \nabla f(\|X\|, \sigma) \leq \bar{\omega}$*,*
5. $\exists\lambda \in \mathbb{R}^+$ *such that* $\forall X \in \mathbb{R}^N, \quad \lim_{\sigma\to 0} = \lambda \begin{cases} 0 & \text{if } X = 0 \\ 1 & \text{otherwise} \end{cases}$.

Then,

1. $\lim_{i\to+\infty} \inf \ \{f(\|X\|, \sigma_i)\} = \inf \ \{f(\|X\|, \sigma_0)\}$*,*
2. *If* $\forall i \in \mathbb{N}, \quad X^{\sigma_i}$ *is a minimiser of* $f(\|X\|, \sigma_i)$*, then the sequence* $(X^{\sigma_i})_{i\in\mathbb{N}}$ *is bounded and all its points are minimisers of* $f(\|X\|, \sigma_0)$*,*
3. *If* $f(\|X\|, \sigma_0)$ *has a unique minimiser* X^**, then* $\lim_{i\to+\infty}(X^{\sigma_i}) = X^*$*.*

Note that based on the previous theorem, we can derive that: a) a continuous function $f(\|X\|, \sigma)$ is also level-bounded and b) the set of minimisers of $f(\|X\|, \sigma)$ is non-empty and compact.

Definition 11: [60, 101] *A function $f((\|X\|, \sigma)) \in \mathbb{R}^N \times \mathbb{R}^+ \to \mathbb{R}$ is called level-bounded in X locally uniformly in σ if for each $\sigma' \in \mathbb{R}^+$ and $\bar{\lambda} \in \mathbb{R}$, there exists a neighbourhood $V_{\sigma'}$ of σ' along with a bounded set $S \subset \mathbb{R}$ such that $\{X \in \mathbb{R}^N : f(\|X\|, \sigma) \leq \bar{\lambda}\} \subset S$, $\forall \sigma \in V_{\sigma'}$.*

At this point and before closing this section we briefly provide a counter example, discussed in [126] to demonstrate that the l_p ($p \in (0, 1]$ arbitrarily chosen) norm based regularisation approach may fail to recover a sparse solution (in contrast to the ordinary linear constrained l_p norm based approach which is not affected). In cases where no prior knowledge of the noise or the sparsity of the solution is known the following l_2/l_p norm based penalised form can be used [126, 183]:

$$F(\hat{X}) := \min_{\hat{X}} \frac{1}{2}\|C\hat{X} - Y\|_{l_2} + T\|\hat{X}\|_{l_p}, \tag{6.11}$$

where Y is the samples vector, C is the Sensing matrix and $\hat{X}$ is the unknown sparse vector. The parameter T is used as a tradeoff between data quality and sparsity. Note that this problem is a variation of the Equation (4.39) discussed in Section 4.7.2, used in TwIST package [17, 18] (See Section 8.4 for details), but due its non-convexity this problem is not equivalent to its constrained l_0, l_1, l_2 norm based counterparts [51, 60, 183] (for more details on these optimisation principles see Sections 4.7.1 and 4.7.2). For the same reason of non-convexity any sparse recovery method, including the l_0 heuristic, based on the constrained l_0, l_1, l_2 norm based optimisation principles is able to successfully recover the sparse vector of the linear system provided in the following proposition.

Proposition 1: [126] *Assume a linear system of equation of the form $C\hat{X} = Y$, constructed as $Y = Y_1 + Y_2$ and $C = [Y_1, Y_2, aI_N, aI_N] \in \mathbb{R}^{N\times(2N+2)}$, where I_N denotes a $N \times N$ identity matrix, $a = |\langle(Y_1, Y_2)\rangle|_{l_p}$ (l_p norm of inner product) and any $Y_1, Y_2 \in \mathbb{R}^N$. It can be easily seen that the system has an optimal solution $X^* = [1, 1, 0 \ldots, 0]^T \in \mathbb{R}^{2N+2}$. Assume now a solution to the same system as $\bar{X} = [0, 0, Y_1/a, Y_2/a]^T \in \mathbb{R}^{2N+2}$. Applying Equa-*

tion (6.11) for these two solutions we have $F(X^*) = 2^{1/p}T$ *and* $F(\bar{X}) = T$*, which implies that* $F(X^*) > F(\bar{X})$ *and thus* X^* *cannot be recovered by solving the problem in Equation (6.11). Furthermore, the relative error in this case is* $(F(X^*) - F(\bar{X}))/F(\bar{X}) = 2^{1/p} - 1 \geq 1$.

It can be easily seen that due to the ill-posedness of the linear system $C\hat{X} = Y$ (null-space $Null(C) = \{X : CX = 0\}$), a better estimation of the data misfit between the samples Y and the estimate $\hat{X}$ is the back-projection based on samples $\|C^T Y - \hat{X}\|_{l_2}$ instead of the ordinary data misfit given by $\|C\hat{X} - Y\|_{l_2}$. This is the reason that the variation of the former (back-projection) has been used in the generation of new solution in the l_0 heuristic.

6.4 Particle Swarm Optimization algorithms

During the last few years the interest in swarm based techniques raised steadily. Compared to traditional search and optimization procedures, such as calculus-based and enumerative strategies, swarm based techniques are robust, globally oriented and generally more straightforward to apply in situations where there is little or no a-priori (prior) knowledge about the problem to solve [112, 206, 214]. This family of algorithms does not require a sophisticated initial solution to proceed and improve or any derivative information or estimate of this initial solution, since they are stochastic in nature and thus capable of searching the solution space with very good likelihood of finding the global optimum. As the PSO was developed as an optimisation tool, it has a number of variants. In this thesis, an extended and refined variation of the Particle Swarm Optimisation (PSO) algorithm, using the concepts of neighbourhood and gradient based methods, has been proposed as an efficient way to solve the l_0 norm based optimisation problem.

The Particle Swarm Optimisation (PSO) algorithm is a population based evolutionary algorithm which was developed by Kennedy and Eberhart in 1995, based on the swarm behavior of fishes and birds [112, 206, 214]. Many algorithms, such as Ant Colony Optimisation and Virtual Ant algorithms, use the behaviour of the so-called swarm intelligence. Swarm intelligence is much simpler as an optimisation principle despite its similarities with other optimisation methods. It does not use complex mutation/crossover operations, used in Genetic Algorithms, complex Temperature parameters, used in Simulated Annealing, or pheromone parameters used in Ant Colony algorithms. Instead, swarm methods use real random numbers and global communication (i.e. best minimum or maximum value) among

swarms, which is easier and straightforward to implement without other complex parameters. Under this concept, the objective function of the problem is associated with the movement of swarms (i.e. the path of swarms) and the search for the best value of the objective function is achieved by adjusting the location of the swarms. A typical solution generation for a particle can be given as follows [112, 206, 214]:

$$X_{t+1} = (1-\beta)X_t + \beta f(X_t) + \alpha(\epsilon - 0.5), \tag{6.12}$$

where t is the current iteration, X_{t+1} and X_t is the new and the current solution of a particle, $f(X_t)$ is the objective function value based on the current solution, $\epsilon \sim U[0,1]$ is a small random number generated using the Uniform distribution, $\alpha \in [0.1, 0.4]$ (stochastic step) and $\beta \in [0.1, 0.7]$ (deterministic step) are some small randomly generated numbers using the Normal distribution which are based on the nature of the optimisation problem.

In general, swarm based methods begin by initialising and associating a random swarm of a finite number of swarms (in a finite dimensional space) with feasible candidate solutions. Then the swarms search the state space of the objective function by adjusting the velocity of individual swarms, carrying a solution, as the piecewise path is formed by a quasi-stochastic manner using their neighboring swarm's experience [112, 214]. Each particle's movement has two main components [112, 206, 214]: a stochastic and a deterministic one for the generation of a new solution. Each swarm is attracted towards the position of the current global best and its own best location in history, while at the same time it has a tendency to move randomly. When a swarm finds a location that is better than any previous locations, it updates it accordingly as the new current best. At each iteration, there is a current best location for each particle that forms the current best path for this iteration (for all the swarms). The purpose of the algorithm is to find the global best solution among all current best solutions represented by the swarms' locations till no further improvement is found, based on the path created by the swarms and the given certain number of iterations. Usually, the PSO methodology is used as a concept for the optimization of nonlinear functions due to their complexity and difficulty of being solved by other traditional stochastic or deterministic methods (More details can be found at [112] and [214]).

6.5 Naive solution approaches

Let's assume a system of N linear equations with M unknowns which can be written in the following algebraic equation:

$$CX = Y, \tag{6.13}$$

where $C \in \mathbb{R}^{M \times N}$ is the coefficients matrix and $Y \in \mathbb{R}^M$ is the right-hand side vector. In general, the aim is to find the solution $X \in \mathbb{R}^N$ (if exists) given the constant terms Y. If $M > N$ then we have an overdetermined system of linear equations and a least squares solution is found by solving $\min \|CX - Y\|_{l_2}$ [28, 163, 207]. In case matrix C is normal (it is full rank, square with non-zero determinant and thus $M = N$) and $\text{rank}(Y) = \text{rank}(C) = N$ (implying that $Y \in \mathcal{R}(C)$ as Y belongs to the real subspace of C, see Appendix B) then it is invertible and as a result the system has only one unique solution which is easy to obtain. This solution is given as follows [28, 163, 207]:

$$X = C^{-1}Y, \tag{6.14}$$

which is established by multiplying both of the members with the inverted matrix C^{-1}, since CC^{-1} gives the identity matrix I_N. The inversion of the matrix C can be achieved by using several methods. The well-known Cramer formula, for an invertible matrix C with columns $C_1, \ldots, C_M$ gives the solution [28, 163, 207]:

$$X_i = \frac{\det(C_1, \ldots, C_{i-1}, Y, C_{i+1}, \ldots, C_M)}{\det C} \tag{6.15}$$

In CS we are interested in the case where $M \ll N$. In this case the Cramer method is unsuitable due to the cost in execution time which is prohibitive and also because C matrix is not invertible. The famous direct method of the Gaussian elimination which reduces the problem to a solution of linear system with a triangular matrix cannot be applied as well since the matrix C is not square (invertible or not).

An alternative way for solving Equation (6.13) is the LU decomposition or factorisation. It consists of factorising the matrix C into a product of two triangular matrices, which are easier to invert than the original matrix. If L is the lower triangular and U is the upper

triangular then $C = LU$ [28, 163]. It is in fact the same as the Gaussian elimination, where the solution of the linear system $CX = Y$ is equivalent to the simple solution of two triangular systems $LW = Y$ and $UX = W$ [163]. A similar method is the Cholesky method which is only applicable to real symmetric, positive definite matrices. It consists of factorising the matrix C in the form $C = BB^T$, where B is a unique real lower triangular matrix, such that all its diagonal elements are positive [207]. Similarly, variations of gradient methods can only be applied in real symmetric positive definite matrices.

However, matrix C is singular [17], which means its inverse does not exist and thus all the previous methods cannot be applied. Note that the high sparsity in X can only improve the operations count and the storage necessary to solve the linear system. It cannot affect in any way the singularity of matrix C. Moreover, note that the inversion of matrices and matrix multiplication have the same asymptotic complexity, which means that if there exists an algorithm for one of these operations (e.g. $O(N^q), q \geq 2$ number of operations and N the length of the vector), then an algorithm for the other operations can be constructed with the same complexity (e.g. $O(N^q), q \geq 2$) [3].

In order to find a solution, if any, of the linear system in Equation (6.13) we need to use alternative techniques of optimisation, such as an interpolation [18] using the Least Squares method. These methods are mainly based on the idea of using a generalised and under conditions inversion of the matrix C satisfying some certain properties. An optimal interpolator in $\mathbb{R}^N$ for the under-determined linear system of equations can be defined as the minimum of the l_2 norm of the error vector [28, 39, 215]:

$$\arg\min_{\hat{X}} \sum_{i=1}^{M} \sum_{j=1}^{N} \|Y_i - C_{ij}\hat{X}_j\|_{l_2} + \mu \sum_{i=1}^{N} \|\hat{X}_i\|_{l_2}, \tag{6.16}$$

where μ is a regularisation parameter which controls the smoothness and thus reduces the over-fitting.

[17] In general, if the linear transformation $C : \mathbb{R}^N \to \mathbb{R}^M$ is singular (its rank is less than N) and $Y \notin \mathcal{R}(C)$ (not included in the real subspace of C) the equation $CX = Y$ is unsolvable since there is no feasible X from a chosen subset of C [28, 163]. However, the design of C guarantees that there is an optimal solution to $CX = Y$ and it's the sparsest one. See URP in Section 4.5, Definition (9).

[18] In general, interpolation is an approximate representation of a function using values of this function at some certain finite set of points in a finite dimensional space. Several techniques have been proposed in the literature as Interpolating methods, such as splines, Gaussian, linear, polynomial, etc. Throughout this thesis we will assume only the most representative types of interpolation. See Appendix B for details.

This approach though simple in space $\mathbb{R}$, provides poor quality of bounded solutions around the measurement points. In fact, $\hat{X}$ cannot be interpolated efficiently by (6.16) and its quality is severely degraded especially in high dimensions due to a large number of local minima [39, 215]. The solution accuracy highly depends on the availability of points in Y, which is limited especially in the high dimensionality (due to the low dimensional/rank matrix C used for compression), as an excessive large data set of measurements Y is needed. This conventional approach of involving all the dictionary vectors in the solution does not generally allow us to derive categorically or physically meaningful solution, as the solution contains less than M elements without knowing their exact position [28, 39, 215]. For example, consider the problem of positioning M data points randomly in the hypercube [19] $[0,1]^N$, repeatedly for a very large N and a very small $M \ll N$. Then $\lim_{N\to\infty} \min_{M\neq M'} E\|x_M - x_{M'}\|_{l_1} = 1$, which means that the expected distance between any two points is equal to the side of the hypercube [28, 215].

An alternative approach is to introduce a greedy approach very similar to one provided in [81]. The key steps of this approach are as follows:

Problem: Determine a vector X s.t. $CX = Y$

Inputs: vector (samples) Y and sampling matrix C

Output: Approximation vector $\hat{X}$

Stopping criterion: till a level of accuracy (ϵ) is reached

Greedy Approach:

1) Start with an initial set of candidate solutions (vectors) $\{\hat{X}_0, \hat{X}_1, \ldots, \hat{X}_{N-1}\}$.

2) Every candidate is built and generated so as $\max \|d(X)\|_{l_2}$, $d(X) = 1/\|C_\ell (C_\ell^T C_\ell)^{-1} X\|_{l_2}$, $X \in \sum_K$, and C_ℓ is a submatrix of C with columns indexed by ℓ. The matrix C_ℓ [20] is extracted from C so as it has large isometry constant $\delta_K^{\max}(C_\ell)$ and full rank, with ℓ set be such that for $\forall i \in \ell \mid < C_i X_i, d(X) > \mid = 1$.

3) Select the largest value from the candidate solutions $\max\{\hat{X}_0, \hat{X}_1, \ldots, \hat{X}_{N-1}\} = \hat{X}_{\max}$.

4) Generate new values for each iteration i, $\hat{X}_{\max}^{(i)} = \hat{X}_{\max}^{(i-1)} + \xi D_r$, with $\xi \in \{+1, -1\}$ and D_r the dirac vector (delta function) at location r, with $r \in \{1, 2, \ldots, N\}$ (N is the

[19] A hypercube is an N-dimensional closed convex area of 2^N points in $\mathbb{R}^N$ with values 0 or 1 unit length (unit hypercube) [207].

[20] Given a set $\ell \subseteq \{1, 2, \ldots, N\}$ representing a collection of column indexes we can define a submatrix C_ℓ as the one formed by taking only the rows of C using the set ℓ so as $C_\ell \in \mathbb{R}^{M\times(\sharp(\ell))}$. Alternatively, using the same concept we can say that C_ℓ represents a submatrix from C by setting all but ℓ columns to zero.

length of X). The proper choice of ξ is made so as to $\max \|d(X)\|_{l_2}$. Note that an alternative selection rule could be $\min_{j \notin \ell} \|1 - |\langle d(X), C_j \rangle|\|_{l_2}$ (i.e. j-th column of C) instead of $\max \|d(X)\|_{l_2}$.

5) Stop if $\|X - \hat{X}\|_{l_2} \leq \epsilon$ $(0 \leq \epsilon \leq 1)$, otherwise go to Step (2)

However, this approach yet simple and straightforward is very computationally expensive especially for large M and N. Although the linear system is reduced from $CX = Y$ to $C_\ell X = Y$ by selecting only a subset of columns (components), the appropriate choice of the "best" M (or fewer) columns out of N which lead to the smallest feasible solution for X requires examining and comparing all $\binom{N}{M}$ choices [28, 142]. An alternative approach with similar computational cost is called Block triangular back-substitution [93]. Here we assume that C is a reducible matrix and thus C_ℓ can be constructed by performing C to upper triangular form (matrix decomposition) and then back substitution (i.e. $X_{\text{new}} = X_{\text{old}} - C_\ell X_{\text{old}}$) [93]. This is usually achieved iteratively by constructing a small block of column and row permutations with integer indexes, where the diagonal blocks of the newly permuted matrix are factored, thus saving time.

At this point, it is important to mention that the system in Equation (6.13) is what we call an ill-posed problem, where the number of equations is much smaller than the number of unknowns and as a result there are many possible solutions. In CS theory the reconstruction problem is always ill-posed and the matrix C not invertible. Therefore, alternative problem formulations have to be applied for solving this problem. In fact, to overcome this problem we need to additionally consider the sparsity and incoherence of the measurements. CS reconstruction is then typically solved using either a convex relaxation of the recovery in (6.16) such as (ϵ is a small constant close to zero) [28, 196, 215]:

$$\min_{\hat{X}} \|\hat{X}\|_{l_1} \text{ s.t. } \|Y - C\hat{X}\|_{l_2} \leq \epsilon \tag{6.17}$$

or with other Greedy algorithms (i.e. OMP, IHT) or Basis Pursuit methods (i.e. LASSO, IRLS). Further details about these methods will be given in the following Chapters 7 and 8.

6.6 Design of the heuristic algorithm

This subsection describes the proposed algorithm for solving the l_0 norm based problem. It is mainly based on the PSO algorithm to generate new solutions for solving the problem defined previously in Equation (6.8). Due to the ill-posedness of the research problem and the NP-hardness (reduction to the Set Covering problem), metaheuristic algorithms [21] are particularly suitable and useful. Another reason for using stochastic meta-heuristics is the high computational cost of log-barrier Newtonian methods (especially for the calculation of Hessian matrices due to the high dimensionality of the problem) and the high recovery error of conventional methods, such as the classical least squares method. Moreover, various studies show that PSO algorithms can outperform Genetic Algorithms and other conventional algorithms for solving many optimization problems. This is partially due to that fact that the broadcasting ability of the current best estimates gives a better and quicker convergence towards the optimality [5, 112, 166, 214]. However, PSO algorithms are almost memoryless since they do not record the movement paths of each particle, and it is expected that this can be further improved using short-term memory in the similar fashion as that in Tabu search. A slight variation of this development, which incorporates short term memory, is proposed for solving the l_0 norm-based sparse recovery problem.

For the implementation of this algorithm we have used the programming environment Matlab $R2014b$ since it has become a de-facto standard in a wide range of technical applications. Matlab is an interactive environment and programming language for scientific computation. Its distinguished feature is the use of matrices as the only data type, which is simply a rectangular array of real or complex numbers [93, 94]. Another benefit of using it is that many areas are catered for by a wide range of toolboxes along with extensive visualisation and analysis tools [94, 195]. Matlab has also an open and extensible architecture allowing individual users to develop further routines for their own applications [93]. The language, integrated tools, and built-in mathematical functions enable you to explore multiple research approaches and computational paradigms and thus solve the sparse recovery

[21] Throughout this thesis with the term heuristic/metaheuristic we mean an efficient version of an optimisation algorithm which is based on the specific problem structure, but it's convergence to an optimal solution has not been proven [135, 206]. Based on the nature of the problem and information gathered a heuristic can efficiently decide which candidate solution should be generated and used in every individual step of the method, following some predefined criteria [117, 135].

problem faster than with spreadsheets or other traditional programming languages, such as C/C++ or Java. All these qualities provide a uniform, familiar and ideal environment on which we can design, build and test our algorithm. It is not surprising that Matlab is used by more than 5000 universities and colleges worldwide. Further details and release information about Matlab can be found at http://www.mathworks.com/products/matlab/.

6.6.1 The Classical Least Squares case

Let's assume we have N unknown parameters $X_1, \ldots, X_N$ are to be estimated from M observations $Y_1, \ldots, Y_M$ to which they are linearly related as follows [28, 108, 188]:

$$Y_i = \sum_{j=1}^{N} C_{ij} X_j + U_i, \quad i = 1, 2, \ldots, M \tag{6.18}$$

where the C_{ij} are the known coefficients (i.e. sampling matrix C) and the U_i are the independent random variables with approximately identical distributions (i.e. noise). In vector notation we have:

$$Y = CX + U. \tag{6.19}$$

Here $C \in \mathbb{R}^{M \times N}$ (with $M \leq N$), $Y \in \mathbb{R}^M$ and $X \in \mathbb{R}^N$. A classical approach to solve this problem is by minimising the sum of squares [28, 108]:

$$\min_X \|CX - Y\|_{l_2} = \sum_{i=1}^{M} (Y_i - \sum_{j=1}^{N} C_{ij} X_j)^2, \tag{6.20}$$

where the objective is the difference of squares of the residuals and the vector X is the optimisation variable. This is an unconstrained Quadratic Programming problem, which has many applications and comes in many names, e.g., regression analysis or least-squares approximation. The solution of this problem can be reduced to solving a system of N linear

equations obtained by differentiating the Equation (6.20) [22].

$$\sum_{i=1}^{M}(Y_i - \sum_{j=1}^{N} C_{ij}X_j)X_{ij} = 0, \tag{6.21}$$

which in vector notation gives us,

$$C^TCX = C^TY, \tag{6.22}$$

which gives the analytical (unique) solution

$$\hat{X} = (C^TC)^{-1}C^TY, \tag{6.23}$$

assuming that C has full rank N and the columns of C are independent [28, 108]. When the solution is not unique (i.e. ill-posed problem with $M \ll N$) the calculation $(C^TC)^{-1}C^TY$ gives the solution with the minimum Euclidean norm, which is sometimes called pseudo-inverse as C is now singular, while in ill-posed problems the matrix $C\hat{X}$ gives the Euclidean projection on $\mathcal{R}(C)$ (i.e. real subspace of the transformation C) [28, 188].

The least squares estimates $\hat{Y}_i$ of the expected values $E(Y_i) = CX$ of the observations (i.e. fitted values) are given as [28, 108, 188]:

$$\hat{Y} = X(C^TC)^{-1}C^TY = HY, \tag{6.24}$$

with

$$H = (C^TC)^{-1}C^TY \tag{6.25}$$

The matrix H is sometimes called "hat matrix" or "model matrix". It provides the best possible estimation of the regression coefficients in terms of X. It is a symmetric $N \times N$ projection matrix (i.e. $HH = H$ and $\sum_{j=1}^{N} C_{ij} = 1, (1 \leq i \leq M)$) which has N eigenvalues equal to 1 and $M - N$ eigenvalues equal to 0 [33, 188]. The trace of H is $tr(H) = N$. In

[22] Note that (6.20) can be expressed as a convex quadratic function as: $R(X) = X^TC^TCX - 2Y^TCX + Y^TY$, which can be differentiated as: $\nabla R(X) = 2C^TCX - 2C^TY = 0$ and then finally give the so-called normal equations in (6.22).

general the least squares problem can be solved very accurately in a time approximately proportional to $O(M^2N)$, with a known constant [28, 33, 188]. Of course in cases where the matrix C is sparse, we can usually solve the least-squares problem much faster.

6.6.2 Regularised Least Squares

Again consider the problem of estimating $\hat{X} \in \mathbb{R}^N$ from measurements or experiments given by $C \in \mathbb{R}^{M\times N}$ and $Y \in \mathbb{R}^M$. Then we have two quadratic objectives to minimise [28, 163, 188]:

$$f(\hat{X}) = \|C\hat{X} - Y\|_{l_2} = \hat{X}^T C^T C\hat{X} - 2Y^T C\hat{X} + Y^T Y, \tag{6.26}$$

as a measure of misfit between $C\hat{X}$ and Y and

$$g(\hat{X}) = \|\hat{X}\|_{l_2} = \hat{X}^T\hat{X}, \tag{6.27}$$

as a measure of the size of $\hat{X}$, which is a regularisation (i.e. adding a term to the objective function that penalises large values of $\hat{X}$). In this case, the aim now is to find the best value for X so as it minimises both $f(\hat{X})$ and $g(\hat{X})$ objectives. The problem can be redefined as a quadratic optimisation problem [28, 163, 188]:

$$\min_{\hat{X}} \|C\hat{X} - Y\|_{l_2} + \delta\|\hat{X}\|_{l_2} = \hat{X}^T(C^TC + \delta I_N)\hat{X} - 2Y^TC\hat{X} + Y^TY, \tag{6.28}$$

which yields the analytical solution:

$$\hat{X} = (C^TC + \delta I_N)^{-1}C^TY, \tag{6.29}$$

where I_N is a positive-definite (diagonal) regularisation matrix often chosen to be the identity matrix and $\delta > 0$ is a weight (small positive constant) which yields the Pareto optimal solution for the two objectives [23]. In essence, the regularised version gets the minimum in terms of misfit and sparsity level (support), out of all possible vectors as an estimate of $\hat{X}$, which is unique even if C is low rank or singular (i.e. it is not rank deficient) [28, 33, 163].

[23] Note that in the special case when $\delta \to \infty$ the solution is $\hat{X} = 0$, while in the case when $\delta \to 0$ we obtain the solution $\hat{X} = (C^TC)^{-1}C^TY$, which is the pseudo-inverse or Moore-Penrose inverse, defined in (6.25) [28, 207].

This is due to the weight δ used in the inversion formula ($C^TC+\delta I > 0$), which can be found using the trace or eigenvalues of matrix C^TC (i.e. $\delta = \text{trace}(C^TC)/\text{trace}(I_N)$ or $\delta = 2/\lambda_{\max}$; the maximum eigenvalue)[24]. This general form of regularisation is sometimes also called Tikhonov regularisation or ridge regression with Euclidean norms [28, 33, 188].

In some applications a useful extension of Equation (6.28) is to replace the regularisation term $\|\hat{X}\|_{l_2}$ with $\|D\hat{X}\|_{l_2}$, where D is an approximate differentiation or second cone order differentiation, as a measure of variation or smoothness of $\hat{X}$ similar to TV [28]. In practice D can be a simple approximation of the i-th component of X represented as $N(\hat{X}_{i+1} - \hat{X}_i)$ [28]. An alternative approach could be to replace the identity matrix I_N in (6.29) with a (tridiagonal) Toeplitz or Vandermonde matrix $D \in \mathbb{R}^{N\times N}$, which have been efficiently used in Polynomial Least squares fitting and polynomial (piecewise cubic) interpolation methods (using polynomials as $a_0x^0+a_1x^1+a_2x^2+a_3x^3+\ldots$). In linear algebra a (diagonal constant) Toeplitz matrix can be defined as [28, 163, 207]:

$$T_N = \begin{bmatrix} X_0 & X_1 & X_2 & \cdots & X_{N-1} \\ X_{-1} & X_0 & X_1 & \cdots & X_{N-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ X_{-N+1} & \cdots & X_{-2} & X_{-1} & X_0 \end{bmatrix} \tag{6.30}$$

while a Vandermonde matrix can be defined as follows [28, 163, 207]:

$$V_N = \begin{bmatrix} 1 & X_1 & X_1^2 & \cdots & X_1^{n-1} \\ 1 & X_2 & X_2^2 & \cdots & X_2^{n-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & X_N & X_N^2 & \cdots & X_N^{N-1} \end{bmatrix} \tag{6.31}$$

where $X_i > 0$ is the corresponding non-negative i-th element of a matrix X. Further details on the different regularised variations of pseudo-inverse can be found in Section 9.4.3.

[24]Eigenvalue and eigenvector of a squared matrix $A = C^TC$ are a scalar λ and a non-zero vector $V \in \mathbb{R}^N \neq 0$ so as: $AV = \lambda V$, or $(A - \lambda I)V = 0$ using Crammer's rule ($V \neq 0$), which implies $A - \lambda I$ is singular and thus $\det(A - \lambda I) = 0$ [33, 163]. Note that in general it is required that $\delta \geq 2/(\lambda_{\max} + \lambda_{\min})$. However, calculating both the largest and smallest eigenvalue of the non-negative definite matrix C^TC can be very expensive for large matrices C (e.g. in medical images) and without much efficiency in the outcome ($\lambda_{\min}$ is very small and thus negligible especially in ill-conditioned cases). [28, 33, 163]

6.6.3 Generalised inverse as the best approximate solution

Least Squares method can be viewed as a special case of a Singular Value Decomposition (SVD), which is simply a regularisation (factorisation) strategy for inverting $C \in \mathbb{R}^{M\times N}$ (calculating its pseudo-inverse with $M \ll N$). SVD can find the best possible solution to a system of linear equations corresponding to a singular matrix with no exact solution. It is a complete orthogonal decomposition which can provide the l_2 norm of the minimum residuals in the null-space out of all possible $N - M$ solutions (apart from any degeneracies in the solution) [9, 95, 163]. In particular, a product of three matrices can be used to replace a singular matrix C^* as follows [9, 33, 95, 163]:

$$C^* = V^T \left[\mathrm{diag}(1/W)\right] U, \tag{6.32}$$

where $V \in \mathbb{R}^{M\times N}$ has orthonormal columns and $U \in \mathbb{R}^{N\times M}$ has orthonormal rows, while $\mathrm{diag}(1/W) \in \mathbb{R}^{M\times N}$ is a non-negative diagonal matrix with zero values apart from its diagonal, calculated based on the rank of matrix C ($W \in \mathbb{R}^{M\times N}$ are small values, as weights, or singular values). This matrix is close to the original one with respect to the Frobenius matrix norm ($\|C\|_F = \sqrt{\sum_i \sum_j |C_{ij}|^2}$), in terms of numerical stability and precision [9, 33, 95, 163].

Theorem 18: [33, 95] *Assume a linear system of equations of the form $CX = Y$, where $C \in \mathbb{R}^{M\times N}$ is a singular matrix which can be inverted given the Equation (6.32). If $V \in \mathbb{R}^{M\times N}$, $U \in \mathbb{R}^{N\times M}$, $Y \in \mathbb{R}^{M\times 1}$ and $diag(1/W) \in \mathbb{R}^{M\times N}$, then*

$$\hat{X} = V^T \left[diag(1/W)\right] UY \tag{6.33}$$

minimises $\|C\hat{X} - Y\|_{l_2}$ and has the smallest l_2 norm out of all minimisers with residual $R = \|C\hat{X} - Y\|_{l_2} = \|(I - CC^*)Y\|_{l_2}$. Note that if $\mathrm{rank}(C) = N$, then $C^* \approx (C^TC)^{-1}C^T$, while if $\mathrm{rank}(C) = N = M$ then $C^* = C^{-1}$. Indeed, the least squares solution is a stable solution of minimal norm which can be used as the best approximate solution.

Definition 12: [89, 190] *Let $C \in \mathbb{R}^{M\times N}$ be a bounded linear sampling matrix. Then*

1. *$X \in \mathbb{R}^N$ is called least squares solution of $CX = Y$ iff*

$$\|CX - Y\|_{l_2} = \inf \ \{\|CZ - Y\|_{l_2} \ |Z \in \mathbb{A}\} \tag{6.34}$$

2. *$X \in \mathbb{R}^N$ is called best approximate solution of $CX = Y$ iff X is a least squares solution of $CX = Y$ and*

$$\|X\|_{l_2} = \inf \ \{\|Z\|_{l_2} \ |Z \textit{is least squares solution of } CX = Y\} \tag{6.35}$$

holds.

In general terms, the generalised inverse C^* is a unique linear extension of C, which gives the best approximate solution as a solution operator mapping Y back onto the model $CX = Y$. This approximation of the minimum norm solution can be further improved by introducing a regularisation operator δ, especially in ill-posed cases. In such cases, the computational cost of the calculations of the generalised inverse may be high, as the number of search directions may be high, or the data model might not be known precisely (e.g. noisy cases). Regularisation operator comes as an improvement in the calculation of C^* in order to approximate better $\hat{X}$ on noisy data Y (solution stability under noise). These considerations lead to the following Definition.

Definition 13: [89, 190] *Let $C \in \mathbb{R}^{M \times N}$ be a bounded linear sampling matrix $\delta_0 \in (0, +\infty)$. For every $\delta \in (0, \delta_0)$ and $\upsilon \in [0, \|C\|_{l_2}]$ let $T_\delta = 1/(\delta + \upsilon) \in \mathbb{R}^{N \times M}$ be a continuous and not necessary linear operator. The regularisation operator T_δ exists for C^*, if for all $Y \in \mathbb{D}(C^*) \subseteq \mathbb{R}^{M \times N}$ (domain of generalised inverse) there exists a parameter choice rule $\delta = \delta(\epsilon, Y')$, depending on noise level ϵ (assuming Y' is the noisy data and Y the exact data), such that*

$$\lim_{\epsilon \to 0} \sup \ \{\|T_\delta Y' - C^* Y\|_{l_2} \ |Y' \in \mathbb{R}^M, \|Y' - Y\|_{l_2} \leq \epsilon\} = 0 \tag{6.36}$$

holds with

$$\delta : \mathbb{R}^+ \times \mathbb{R}^M \to (0, \delta_0), \tag{6.37}$$

such that

$$\lim_{\epsilon \to 0} \sup \ \{\delta(\epsilon, Y') |\ Y' \in \mathbb{R}^M, \|Y' - Y\|_{l_2} \leq \epsilon\} = 0 \tag{6.38}$$

In general, the aim is to choose T_δ so as $C^T C$ to be continuous invertible and thus achieve convergence in the solution $\hat{X}$. In the simplest case we can choose $\|C\|_{l_2} = 1$ or $\upsilon = 0$ to derive $T_\delta = 1/\delta$. In fact, the pair (T_δ, δ) is called convergent regularisation method (parameter and choice rule) for obtaining $\hat{X}$ under the assumption $Y \in \mathbb{D}(C^*)$. This can be considered as a regularisation form of the generalised inverse C^* where the solution, defined as $\hat{X} = (C^T C + \delta I_N)^{-1} C^T Y$, is a unique minimiser of the two quadratic objectives $f(\hat{X})$ and $g(\hat{X})$ (defined in Section 6.6.2) of the Regularised Least Squares [33, 89].

6.6.4 Initial Solution

Before defining in detail the l_0 heuristic and set the scheme for the procedure followed for updating the current solution, we firstly need to describe how the initial solution of the heuristic is defined. The heuristic works for any solution that belongs to the feasible set, as an initial estimation of the sparsest solution of the linear system $C\hat{X} = Y$. Based on similar approaches which are based on the l_0 norm based problem, particularly in [6, 141, 161, 199], we have chosen the regularised pseudo-inverse of C, as a rough estimate of the sparse solution (derived from the analytical solution of min $\|C\hat{X} - Y\|_{l_2} + \delta\|\hat{X}\|_{l_2}$) which will be refined through the iterations of the l_0 heuristic. This is also the unique minimiser of the objective function $f(\|\hat{X}\|, \sigma)$ on the feasible set, when $\sigma \to \infty$. In particular, for efficiently solving the problem, defined in Equation (6.8), we need to initially choose a relatively large $\sigma^{(0)}$ (in theory infinity [141, 142, 199]) based on the values of the vector X and an appropriate initial solution $X^{(0)}$ so as to achieve a good approximation of the global minimum. By setting $\nabla f(\|\hat{X}\|, \sigma) = 0$ we derive [108, 141, 142]:

$$C^T[\hat{X}_1(\exp(-\hat{X}_1^2/2\sigma^2)), \ldots, \hat{X}_N(\exp(-\hat{X}_N^2/2\sigma^2))] = 0, \tag{6.39}$$

where $\hat{X} \in \mathbb{R}^N$ is the estimate vector. If we now set $\sigma \to \infty$ (or alternatively $\sigma \gg \max\{\sigma_1, \sigma_2, \ldots, \sigma_m\}$), we have $C^T\hat{X} = 0$, which finally gives:

$$\hat{X} = (C^T C)^{-1} C^T Y \tag{6.40}$$

and hence the minimiser of $f(\|\hat{X}\|, \sigma)$ is the minimum l_2-norm solution of $C\hat{X} = Y$. Then, the initial solution in vector format is given as the minimum l_2-norm solution which also enforces sparsity for $\hat{X}$, namely:

$$X_i^{(0)} = (C^T C + \delta I_N)^{-1} C^T Y, \tag{6.41}$$

where $X_i^{(0)}$ is the initial solution vector ($t = 0$) for swarm i, $\delta = 2/(\lambda_{\max} + r)$ (follows conditions of Definition 12 of the previous section), $\lambda_{\max}$ is the maximum eigenvalue of matrix $C^T C$ and $0 \leq r \leq 1$ is a random number, different for every swarm, generated using Normal distribution, so as to increase the stability of the fraction in cases where eigenvalues are close to zero (the determinant is equal to the product of all eigenvalues, which is very close to zero for singular matrices). Note that $C^T Y$ is a back projection of the vector X (projection from the samples). Here we do not use the ordinary pseudo-inverse of C, namely $(C^T C)^{-1} C^T Y$, but its regularised version which increases the rank and enhances the sparsity [28, 163]. This is mainly based on SVD methods where a singular matrix is reformed and factorised with a help of a unitary (orthogonal) matrix and a diagonal matrix (δI_N in our case).

In essence, our aim is to start within the feasible set of values, equally and uniformly distributed among all swarms, based on the regularised pseudo-inverse. Then stochastically move towards better values in order to minimise the objective function, based on the nature and type of the problem (initial vector). The stability and performance of the initial solution is thus guaranteed as a good estimate of the quadratic cost function $\|Y - C\hat{X}\|_{l_2}$. Note that in the initial solution we also consider the case that $(C^T C)$ is the identity matrix (that is the rows of C are orthonormal), where the heuristic will still work. More importantly, for most cases C is created so as C and C^T can be quickly computed resulting on a fast calculation of the initial solution. Another important, but not neccessary, aspect is to keep the highest and lowest values of the coefficients or terms of the original vector so as to enhance the effectiveness of the l_0 heuristic. However, this is not a mandatory step as the new solutions generated by the l_0 heuristic will be corrected using the projection back to the feasibility set (see Section 6.6.8 for details). An alternative approach, if these ranges are not known, is to simply approximate them by calculating the minimum and maximum back-projection based on the samples $C^T Y$ (See Section 9.4.4 of Experiments for further details).

6.6.5 Appropriate decrease sequence for the parameter

After defining the initial solution we need to find a practical decreasing sequence for the σ parameter so as the objective function $f(\|X\|, \sigma)$ approximates the l_0 norm efficiently and the method is not trapped in a local minimum. This decrease has to be controlled by a decrease factor $0 < \beta < 1$ (say $\beta\sigma$), which will control the trade-off between the quality of the recovery and the computational cost. This choice depends on the nature (i.e. values and sparsity) of the initial vector, as a very fast decrease sequence in σ may result the method gets trapped in one of the numerous local minima or it does not converge at all. Here, the σ decrease value of $\beta = 0.5$ has been chosen throughout all the experiments as the proper choice to seek a final solution which will be very close to the global minimum, keeping a good trade-off between computational cost and quality of recovery. The same decrease value has also been chosen for the variation of Steepest Ascent method (discussed in Section 7.5) for consistency in comparing solutions in the Experiments, though slight differences in the σ decrease value do not seem to affect the performance of the l_0 heuristic due to its swarm properties (see Section 9.4.3 for details).

In Experiments Section 9 we will only use the number of the desired iterations T as the stopping criterion of the l_0 heuristic, in contrast to earlier published papers where the number of iterations and the target minimum σ value were the criteria for stopping the execution of the l_0 heuristic (whichever comes first). In theory we want $\sigma \to \infty$ but in practice for efficiency in the approximation we choose a very small final σ value based on the nature of the problem (less than 10^{-50}). This makes the proper choice of the number of iterations more broader and easier to choose as they do not depend on the nature (values) of the sparse vector. Also the number of iterations can affect and particularly increase the efficiency of the l_0 heuristic, without affecting the final σ value, which directly affects the approximation of the l_0 norm and thus the heuristic's performance (see for example Experiments 2 and 3 in Chapter 9).

For the initial value $\sigma^{(0)}$ we want the objective function to be convex near the initial solution $X^{(0)}$. It is also important that $f(\|X\|, \sigma)$ can be convex in the feasible region of X and particularly when the largest element of X is less than the value of σ parameter. This means that $\nabla^2 f(\|X^{(0)}\|, \sigma^{(0)}) > 0$, so $\sigma > |X_i|^{(0)}$, $\forall i = 1, \ldots, N$, where $X_i^{(0)}$ is the i-th

component of $X^{(0)}$. In particular, [108, 142, 161, 199]:

$$\exp(-X_i^2/2\sigma^2) = \begin{cases} 1, & \text{if } |X_i| \ll \sigma \\ 0, & \text{if } |X_i| \gg \sigma \end{cases} \tag{6.42}$$

In other words, the function $f(\|X\|, \sigma)$ remains convex in the region where the largest magnitude of the components of X is less than σ. Based on this a reasonable initial value for σ chosen in all the experiments in this thesis is $\alpha = 2$, $\sigma^{(0)} = \alpha\|X\|_\infty = 2\|X\|_\infty$, so as to ensure that the heuristic starts from a convex region. Of course, there are also other possible choices of the initial σ value which can be considered (though not used in the experiments for consistency purposes), as any initial value of σ roughly between two and five times the maximum absolute value of the initial vector, will virtually act as an infinity for the computed values of X [6, 141]. We can choose, for example, the initial value as $\sigma^{(0)} = \|X\|_\infty + c$, where $0 < c < 1$ is a small value, which can achieve the desired convexity of $f(\|X\|, \sigma)$ initially [161]. Alternatively, the initial value of σ can be calculated using the median absolute deviation, that is $\sigma = \text{median}\{|R_0|\}$, when expressed in terms of residual of the initial solution ($R_0 = \|CX^{(0)} - Y\|_{l_1}$), similar to the sample median proposed in [108]. Consequently, the main idea is to properly choose the current value of σ so as to be the minimiser of the objective function $f(\|X\|, \sigma)$ from the previous (larger) value of σ for every iteration. The randomises of the initial solution will also help the l_0 heuristic avoid being trapped into any local minima reaching the actual minimum for small values of σ and thus giving a good estimate of the minimum l_0-norm solution.

If the initial vector X is not known then we can define a rough estimate for the initial σ based on the collected samples Y; $\sigma^{(0)} = \|C^T Y\|_\infty + c$ for a small random value of $0 \leq c \leq 1$, which is what we have chosen for consistency and simplicity in the Experiments where the initial sampled vector is not known (See Section 9.4.4 of Experiments). Another similar approach (not used in the Experiments) has been introduced in [6, 141] where the parameter σ can be initialised as:

$$\sigma^{(0)} > \nu\|Y - C(C^T C)^{-1} C^T Y\|_\infty, \tag{6.43}$$

where $0 \leq \nu \leq 1$ is a small constant, depending on the samples observations Y. In general, it is expected that for exactly sparse vectors, the σ parameter can be decreased more arbitrary,

as the decrease factor highly depends on the desired accuracy. For noisy cases it is expected that the smallest σ will be roughly twice the generated noise, as small noisy samples must be considered as zeroes and thus ignored. However, for obviously very noisy cases, the search space may be affected more and thus more sophisticated decrease sequence will be needed. In practice the smallest value of σ cannot be zero but a small number very close to zero.

6.6.6 Solution Generation

The generation of a new solution is given in a vector format probabilistically by the following projected swarming gradient based approach (for minimisation problems):

(6.44)
$$X_i^{(t)} = \{X_i^{(t-1)} - \mu R_i^{(t)} \nabla_{X_i^{(t-1)}} f(\|X_i^{(t-1)}\|, \sigma^{(t-1)}) + \exp(-\mu L^{(t)})(C^T Y - (C^T C) X_*^{(t)})\}_{P_K},$$

where $L^{(t)} = \|X_i^{(t-1)} - X_*^{(t)}\|_{l_2}$ is the Euclidean difference vector (acting as a weight) for each element between the solution from the previous iteration ($X_i^{(t-1)}$) and the current best ($X_*^{(t)}$), $R_i^{(t)}$ is a vector of numbers, different for every swarm, randomly generated using Gaussian distribution between 0 and 1, (t) is the current iteration, while $X_i^{(t)}$ and $X_i^{(t-1)}$ is the new and the current solution carried by the i-th swarm which represents the corresponding solutions vector. The positive step $\mu = 2$ represents a small number based on the gradient of the function. This can be easily derived by taking the first derivative:

$$\nabla f(\|X\|, \sigma) = \nabla \exp(-\frac{|X|^2}{2\sigma^2}) = -\frac{2|X| \exp(-\frac{|X|^2}{2\sigma^2})}{2\sigma^2} \tag{6.45}$$

Note that in Equation (6.44) we could use the infinity norm ($\|.\|_\infty$) for calculating L (as presented in earlier research papers by the same author). However, using this approach we assume that the misfit between solution and samples contributes equally to all the elements of the new solution vector $X_i^{(t)}$, which is not the case if we compare it with the current best solution $X_*^{(t)}$ (better adaption to weight and decrease of distance between the swarms and their current best swarm). For a vector $X = [X_1, \ldots, X_N]$ in a finite dimensional coordinate space the infinity norm is defined as $\|X\|_\infty = \max\{\|X_1\|, \ldots, \|X_N\|\}$.

The operator P_K represents a hard-thresholding approach where we keep only the K-largest entries of our new vector $X_i^{(t)}$ and set all the others to zero ($\{.\}_{P_K} = \max(., 0)$). Note that the hard-thresholding approach is usually related to l_0 norm based problems, while the

soft-thresholding approach is usually related to l_1 norm based problems. The main difference is how we approach the smallest values of a vector which are below a certain threshold (based on some prior-knowledge of the structure of the vector values); hard-thresholding approach sets them to zero while soft-thresholding approach sets them very small values, which are very close to zero (typically less than 10^{-10}) [86, 186].

In essence, X values are recovered using some prior (a-priori) and some likelihood estimates (knowledge) for X. The prior estimate for X is not related to measurements Y and contains only information related to X (sparsity level) using the initial solution in Equation (6.41). New solutions satisfy the prior estimate for X by using the relaxed gradient of the function $\nabla_{X_i^{(t-1)}} f(\|X_i^{(t-1)}\|, \sigma^{(t-1)})$, while the likelihood estimate is based on the measurement model $Y = CX + e$ (where $0 \leq \|e\|_{l_2} \leq 1$) using forward-projection to the feasible set, namely $(C^T Y - (C^T C)X_*^{(t)})$. Here we have chosen to project to N elements in X instead of M elements in Y (i.e. $C^T(CX_*^{(t)} - Y)$ used in AIHT) due to the nature of the ill-posedness of the problem as $M \ll N$. The misfit of the new solution could be corrected more quickly and efficiently in the N-dimensional space ($\mathbb{R}^N$ or C^N), which also enhances the convergence of the l_0 heuristic to the optimal solution. Similar forward-projection approaches have been efficiently applied to relevant ill-posed inverse problems [33, 186].

This solution approach is enhanced by using a random vector $R_i^{(t)} \in (0, 1)$ for the prior knowledge as a relaxed gradient projection and a inertia weight vector $\exp(-\mu L^{(t)})$ so as to control the influence of the current best solution $X_*^{(t)}$ on the other solutions. This aspect improves the l_0 heuristic's search from an initially global search to a more local by assigning greater weight to values closer to the current best and lower to unexpected values. It also improves the l_0 heuristic's convergence rate as the swarms' solutions are influenced from the current best performing swarm. The random vector $R_i^{(t)}$ introduces randomness to the process which enables the heuristic to avoid being trapped to numerous local minima. In fact, Equation (6.44) is a variation of the solution generation for PSO algorithms explained in Equation (6.12) with combined relaxed gradient and forward-projection step (the smaller R means the smaller step sizes chosen, while the smaller the weight $\exp(-\mu L^{(t)})$ means the smaller the misfit between the solution generated and the observations). This is very common in swarm intelligence methods [112, 214], but here we have both a stochastic and deterministic step.

6.6.7 Assumptions and conditions

We will now turn our attention to the necessary assumptions and conditions for efficient sparse recovery using the l_0 heuristic. Before we review the necessary assumptions we will now provide some basic theory in a form of definitions which we will use.

Definition 14: [60, 183] *A function $f : \mathbb{A} \to \mathbb{B}$ is q-Lipschitz continuous, with respect to the l_2 norm on $\mathbb{A}$ if for every $X, Y \in \mathbb{A}$ we have*

$$\|f(X) - f(Y)\|_{l_2} \leq q\|X - Y\|_{l_2}, \tag{6.46}$$

for a small number $0 < q < 2\bar{\Gamma}$, with $\bar{\Gamma} = \|(C^T C)^{-1}\|_{l_2}$ (C is the Sampling vector) or $\bar{\Gamma} = \sup \|\nabla f(z)\|_{l_2}$, for $z \in\in \mathbb{A}$.

Definition 15: [33, 89] *A function $g : \mathbb{A} \to \mathbb{B}$ is called non-expansive with respect to the l_2 norm on $\mathbb{A}$ if for every $X, Y \in \mathbb{A}$ we have*

$$\|g(X) - g(Y)\|_{l_2} \leq \|X - Y\|_{l_2} \tag{6.47}$$

Note that if f is q-Lipschitz continuous then the function $g = 1/qf$ is non-expansive.

Definition 16: [33, 89] *A function $h : \mathbb{A} \to \mathbb{B}$ is called firmly non-expansive with respect to the l_2 norm on $\mathbb{A}$ if for every $X, Y \in \mathbb{A}$ we have*

$$\|h(X) - h(Y)\|_{l_2} \leq \langle (h(X) - h(Y)), (X - Y)\rangle, \tag{6.48}$$

where $\langle , \rangle$ represents the inner product.

Note that a firmly non-expansive function on $\mathbb{A}$ is also non-expansive.

Theorem 19: [10, 33] *Assume $f : \mathbb{A} \to \mathbb{B}$ is a convex and differentiable function and its derivative ∇f is non-expansive and q-Lipschitz with respect to the l_2 norm. Then ∇f is*

also firmly non-expansive. The following are equivalent

$$\|\nabla f(X) - \nabla f(Y)\| \leq q\|X - Y\|_{l_2}, \quad \forall X, Y \in \mathbb{A} \tag{6.49}$$

$$\|\nabla f(X) - \nabla f(Y)\|_{l_2} \leq \langle(\nabla f(X) - \nabla f(Y)), (X - Y)\rangle, \quad \forall X, Y \in \mathbb{A} \tag{6.50}$$

Definition 17 (Local extrema): [151, 183] *If a function $g := \nabla f : \mathbb{A} \to \mathbb{B}$ attains a local minimum or maximum at $X^* \in \mathbb{A}$, then $\nabla g(X^*) = 0$ and X^* is called critical point of $g := \nabla f$. If a local minimum X^* is strict then there is a neighbourhood $\mathbb{O} \subset \mathbb{A}$, containing X^*, such that $g(X^*) < g(Y)$, for any $Y \in \mathbb{O}\backslash\{X^*\}$.*

Note that here we adopt the notion of local minimiser in the gradient as the l_0 heuristic uses the $\nabla f(\|X\|, \sigma)$ to evaluate and generate new solutions. Indeed, a simple and efficient strategy to minimise the first derivative of the objective function in (6.8), used in the evaluation of solutions in the l_0 heuristic, is to construct a sequence $X^{(t)} \in \mathbb{R}^N$ with $t \in \mathbb{N}$ (assuming only one swarm for simplicity) so as:

$$\nabla f(\|X^{(t+1)}\|, \sigma^{(t+1)}) \leq \nabla f(\|X^{(t)}\|, \sigma^{(t)}), \quad \forall t \in \mathbb{N}, \tag{6.51}$$

which can be achieved by

$$X^{(t+1)} = X^{(t)} + R^{(t)} D^{(t)}, \tag{6.52}$$

where $D^{(t)} \in \mathbb{R}^N$ is a suitable search direction, usually expressed as the gradient $\nabla f(\|X^{(t)}\|, \sigma^{(t)})$ (it is differentiable), while $R^{(t)} \in (0, 1)$ is a positive step size with the aim to improve the search direction and increase the convergence of (6.52). Note that Equation (6.52) is equivalent to the first half of Equation (6.44), which generates the new solutions for the l_0 heuristic (μ parameter in Equation (6.44) comes from the gradient of the function $f(\|X\|, \sigma)$). We now proceed with the following assumptions for the l_0 heuristic.

Proposition 2:*Assume a generation of a solution based on Equation (6.52) with step size*

$R^{(t)} \in (0,1)$, $t \in \mathbb{N}$ and search direction selected based on the gradient $\nabla f(\|X^{(t)}\|, \sigma^{(t)})$ so as $X^{(t+1)} \leq X^{(t)}$ for every iteration $t \in \mathbb{N}$ (minimise the gradient at each iteration t).

Proposition 3: *Assume that the function $f(\|X\|, \sigma)$ is concave and twice differentiable. Assume also that the gradient $\nabla f(\|X\|, \sigma)$ is q-Lipschitz continuous. This means that for a small number $0 < q < 2\bar{\Gamma}$, with $\bar{\Gamma} = \|(C^T C)^{-1}\|_{l_2}$ [33] (C is the Sampling vector) or $\bar{\Gamma} = \sup \|\nabla f(\|X\|, \sigma)\|_{l_2}$ [89], for all $X_1, X_2 \in \mathbb{R}^N$ and some parameters $\sigma_1, \sigma_2 > 0$ we have with respect to vector norm $\|.\|_{l_2}$:*

$$\|\nabla f(\|X_1\|, \sigma_1) - \nabla f(\|X_2\|, \sigma_2)\|_{l_2} \leq q\|X_1 - X_2\|_{l_2} \tag{6.53}$$

Also assume that the gradient and the gradient of the gradient of the solution are also bounded:

$$\|\nabla f(\|X\|, \sigma)\|_{l_2} \leq q', \tag{6.54}$$

$$\|\nabla^2 f(\|X\|, \sigma)\|_{l_2} \leq q, \tag{6.55}$$

for all $X \in \mathbb{R}^N$ so as $\nabla f(\|X\|, \sigma)$ is q-Lipschitz with the respect to l_2 norm $q' \in [0, +\infty)$.

Propositions (3) and (2) are important assumptions which are also used in similar iterative sparse approximation techniques for solving ill-posed inverse problems, such as steepest decent, regularised gradient and truncated Newton. Proposition (3) is also reasonable in our broader research context, since the gradient $\nabla^2 f(\|X\|, \sigma)$ exists, which is also the case in the frequency domain (complex numbers of a Transform domain usually follow the power decay law). Indeed $X^{(t)}_{t\in\mathbb{N}}$ is a decreasing sequence and thus $\forall X \in \mathbb{R}^N |\ \nabla f(\|X^{(t+1)}\|, \sigma^{(t+1)}) \leq \nabla f(\|X^{(t)}\|, \sigma^{(t)})$ is a bounded set, where $X^{(t)}_{t\in\mathbb{N}}$ belongs to the compact subset/neighbourhood $\mathbb{O}$ of $\mathbb{R}^N$. Since $\nabla f(\|X^{(t)}\|, \sigma^{(t)})_{t\in\mathbb{N}}$ is decreasing and bounded, then $(\nabla f(\|X^{(t)}\|, \sigma^{(t)}) - \nabla f(\|X^{(t+1)}\|, \sigma^{(t+1)}))_{t\in\mathbb{N}}$ is also a non-negative sequence bounded and converging to zero. Consequently, a global minimiser exists and it is unique due to the decreasing properties of the sequence $\nabla f(\|X^{(t)}\|, \sigma^{(t)})_{t\in\mathbb{N}}$ (i.e. existence of local minimisers) and the properties of Theorems (16) and (17), given in Section 6.3. This outcome can also be verified by the numerical experiments discussed in Chapter 9, where the generated (current) solution $X^{(t)}$

is decreased through iterations (t), while the assumptotic convergence rate of the heuristic is relatively quick (due to its random step in the generation of new solutions). Indeed, the proper choice of the parameter $R^{(t)}$ is very instrumental so as to endow the system with the desired behaviour of swarms and simple enough to allow for mathematical trackability (i.e. to fit perfectly the deterministic model of the problem). In general, the aim is to start with an initial guess within the feasible region and then update the solutions vector $X^{(t)}$ at every iteration (t) (by decreasing the gradient at each iteration) according to the relaxed projected gradient formula in (6.52) so as the weights are unbounded and the residuals descending.

Similarly to the necessary conditions imposed by the gradient of the function used for the back-projection $(R^{(t)}\nabla_{X^{(t)}} f(\|X^{(t)}\|, \sigma^{(t)}))$, which is the first half of Equation (6.44), we also need to define the necessary conditions for the successive approximation through the regularisation step $(\exp(-\mu L^{(t)})(C^T Y - (C^T C)X_*^{(t)}))$, which is the second half of the Equation (6.44). The aim of this part of solution generation is to enforce better solutions with the use of weights both as a misfit between the current best solution $X_*^{(t)}$ and observations Y and as an attractiveness among the swarms $\exp(-\mu L^{(t)})$. For small values of $\hat{X}$ the re-weighting method needs to yield a large weight and thus enhance the search for the sparsest solution. The concept of this regularisation strategy is to add a relatively small value iteratively as a weight through iterations to allow the l_0 heuristic to approach a global optimum and thus finally converge. Here, we adopt a simple approach where all swarms are forced to follow the current best in terms of solution swarm, fact which increases the efficiency of the l_0 heuristic without increasing its execution time. In the Experiments (Experiment (9.4.3) of Chapter 9) we will explore and compare different re-weighting strategies in terms of quality of the derived solution and execution time.

We will now conclude this section with the following proposition as the necessarily condition for the regularisation strategy of the l_0 heuristic.

Proposition 4: *Assume a sequence $X_{t\in\mathbb{N}}^{(t)}$ in vector format, which is generated by the following regularisation step, similar to the l_0 heuristic*

$$X^{(t+1)} = \exp(-\mu L^{(t)})(C^T Y - (C^T C)X^{(t)}), \tag{6.56}$$

where $\mu = 2$ and $L^{(t)}$ is a simple inertia weight vector same as defined in Equation (6.44).

Then the following condition is necessary

$$\|I_N - \eta(C^T C + \delta C^T C)\|_{l_1} < 1, \tag{6.57}$$

where I_N is the $N \times N$ identity matrix and parameter $\eta \in (0, k)$ with $k = 2/\|C\|_{l_2}$ (for Iterative Soft Thresholding), $k = 1/\|C\|_{l_2}$ (for Iterative Hard Thresholding) or $k = 2/(\|C\|_{l_2} + 8\epsilon)$ with $\epsilon = 0.4$ (for Shepp-Logan Phantom image recovery), given that in all cases the Sampling matrix $C \in \mathbb{R}^{M \times N}$ is bounded so as $\|CX\|_{l_2} \geq C\|X\|_{l_2}$ [33, 194].

Note that the regularisation parameter $\delta \in (0, 2/g)$ controls the balance between the sparsity and data fidelity, while g is the largest eigenvalue or trace of matrix $C^T C$. An alternative approach proposed in [49, 196] is to set δ parameter equal to $\zeta\varepsilon\sqrt{2\log N}$, where $\varepsilon \in (0, 1)$ is the Gaussian noise variance, N is the length of original vector to be recovered ($X \in \mathbb{R}^N$) and $\zeta > 2\sqrt{2}$. Similarly for noisy cases it has been suggested in [132, 186] to change the δ values iteratively based on the residuals (de-noising heuristic):

$$\delta^{(t+1)} = \frac{\delta^{(t)}\varepsilon}{CX^{(t)} - Y} \tag{6.58}$$

where t is the current iteration, $\varepsilon = 1$ is the variance of additive Gaussian noise and initial $\delta^{(0)} = 1$ or equal to the initial residual $CX^{(0)} - Y$ in vector format.

6.6.8 Solution Correction

After the generation of the new solutions, we can check whether these solutions satisfy the initial constraints, namely the upper and lower bound of the elements of the vector we initially have. This is important, but not mandatory step, to prevent the tendency of the particle position to explode in magnitude and go beyond the pre-defined ranges of the given vector's values (the solutions are also corrected by projecting back to the feasibility set, so the first step of the minmax correction is not mandatory). In order to ensure these solutions are within the given ranges, we adopt a simple, yet efficient, repair process so as to guarantee the convergence of the l_0 heuristic to a feasible solution. In the following Pseudo-code we describe the whole procedure where $X_{\min}$ and $X_{\max}$ is the upper bound (highest value) and the lower bound (lowest value) of vector X (signal or image) respectively. The aim is to compare and correct the value of each element of X with the lower and upper bound. If it

is smaller than the lower bound, then the lower bound is the current value of the element, while if it is larger than the upper bound, then the current value is equal to the upper bound. The proper choice of these bounds highly depends on the nature and understanding of the problem. If these bounds are not known (initial sampled X vector not known) then back-projection to samples can be used as a rough estimate of them, namely $X_{\min} = \min\{C^T Y\}$ and $X_{\max} = \max\{C^T Y\}$ (See Experiments, Chapter 9 for details).

Note that randomness ($0 \leq \text{rand} \leq 1$ is a small random number) has also been introduced here as a property for each particle so as to enhance the performance of the heuristic and avoid being trapped at a local minimum. This is a common approach introduced to meta-heuristic methods, such as the Simulated Annealing, where some solutions are accepted randomly even if they do not improve the current best solution as an efficient way to help the method avoid being trapped to local extrema (minima or maxima). Here, we have extended this idea by using a local search based on the minimum and maximum values of our original vector together with a threshold, $X_i < 0.05X_{\min}$ for the minimum value and $X_i > 5X_{\max}$ for the maximum value (used only in signals and not in images). Finally, a back projection to feasible region (mandatory correction step) is used to make sure our new solution satisfies the given constraints ($CX = Y$), with I_N the N-element identity matrix and $\delta = 2/(\lambda_{\max} + r)$ the regularisation parameter of the initial solution given in Equation (6.41). All these rules can be summarised as follows:

Pseudocode for values correction
Aim: Feasibility - Ranges Test
Inputs: vector X, sparsity level K, regularisation δ, samples Y, matrix C
Outputs: Corrected vector X
Keep the K-largest entries of X.
for all $i = 1, \ldots, \text{size}(X)$ (Repeat for each dimension of X) **do**
  if $X_i < 0.05X_{\min}$ **then**
    $X_i = X_{\min} + \text{rand}(\text{size}(X_i)) * (X_{\max} - X_{\min})$
  else if $X_i < X_{\min}$ **then**
    $X_i = X_{\min}$
  else if $X_i > 5X_{\max}$ **then**
    $X_i = X_{\max} - \text{rand}(\text{size}(X_i)) * (X_{\max} - X_{\min})$

else if $X_i > X_{\max}$ **then**
$X_i = X_{\max}$
else
The value of X_i remains the same.
end if
end for
Feasible projection onto the set: $X = X - ((C^T C + \delta I_N)^{-1}(CX) - Y)$.
Display corrected vector X

6.6.9 Algorithm Description

In this section the pseudocode of the proposed l_0-heuristic together with the parameter settings used in the simulations are presented with a brief description. It is divided into three parts, namely the Input, Output and the Main Technique which constitutes the summary of the essential steps of the optimisation method. In this case we assume a highly under-sampled complex valued vector X represented in a frequency (transform) domain Ψ.

Heuristic for l_0 norm based recovery
Problem: Determine a vector $X \in \mathbb{C}^N$ s.t. $CX + e = Y$.
Inputs: $\alpha, \lambda_{\max}, \beta, \mu, \epsilon, X_{\max}, X_{\min}, C, Y, K, T, L, \nabla f(\|X\|, \sigma), X_{\min}, X_{\max}, e, \Psi$,
Outputs: best value $\nabla f_*(\|X_*\|, \sigma)$, best $X_* \in C^N$, $X_{**} \in \mathbb{R}^N$ (change domain).
Main steps of the swarm based l_0-heuristic:
1)Generate initial solution for S swarms (every swarm i):
2)Set: $\delta = 2/(\lambda_{\max} + \text{rand})$,
3)Calculate:

$$X_i^{(0)} = (C^T C + \delta I_N)^{-1} C^T Y, \tag{6.59}$$

4)Set all but K-largest entries of $X_i^{(0)}$ to zero, correct its values based on $X_{\max}, X_{\min}$ values (if known).
5)Set initial σ parameter value same for all swarms:

$$\sigma^{(0)} = \|C^T Y\|_{l_\infty}, \tag{6.60}$$

5)Alternatively if $X_{\max}$ is known then ($\alpha = 2$):

$$\sigma^{(0)} = \alpha X_{\max}, \tag{6.61}$$

while $t < T$ (for all iterations T) **do**

for all $i = 1, \ldots, S$ (S Swarms) **do**

6)Evaluate $\nabla f(\|X_i^{(t)}\|, \sigma^{(t)})$ for every swarm i.

7)Find the current best swarm with $X_{\acute{i}}^{(t)}$ so as $\min \nabla f(\|X_{\acute{i}}^{(t)}\|, \sigma^{(t)})$.

8)Keep the current best solution $X_*^{(t)} = X_{\acute{i}}^{(t)}$.

9)Calculate misfit between optimal solution and samples using forward-projection:

$$D^{(t)} = C^T Y - (C^T C) X_*^{(t)}, \tag{6.62}$$

10) Calculate attractiveness-weight parameter between swarms ($\mu = 2$):

$$L^{(t)} = \mu \|X_i^{(t-1)} - X_*^{(t)}\|_{l_2}, \tag{6.63}$$

11)Generate new solutions for all the other swarms following sparsity pattern K by applying hard-thresholding $\{.\}_{P_K}$ ($i \in S - \{\acute{i}\}$):

$$X_i^{(t)} = \{X_i^{(t-1)} - \mu R_i^{(t)} \nabla_{X_i^{(t-1)}} f(\|X_i^{(t-1)}\|, \sigma^{(t-1)}) + \exp(-L^{(t)}) D^{(t)}\}_{P_K}, \tag{6.64}$$

12)Check for feasibility Ranges in $X_i^{(t)}$ (if $X_{\max}, X_{\min}$ are known).

13)Feasible projection onto the set: $X_i^{(t)} = (C^T C + \delta I_N)^{-1}(C X_i^{(t)} - Y)$.

14)Set $\sigma^{(t)} = \beta \sigma^{(t-1)}$, with $\beta = 0.5$.

end for

end while

15)Display $\nabla f_*(\|X_*^{(t)}\|, \sigma)$ and $X_*^{(t)} \in C^N$.

16)Reconstruct vector $X_{**} = \Psi^{-1} X_*^{(t)}$ (apply Inverse transform to derive $X_{**} \in \mathbb{R}^N$). Otherwise, $X_{**} = X_*$.

if $\|e\|_{l_2} \neq 0$ (noisy samples) **then**

17)Apply a filter to smoother vector, $X_{**} = X_{**} \star h(\text{size}(X_{**}))$ (optional step).

else

Keep X_{**} without changes.

end if

18)Display the recovered vector X_{**}.

19)Post-process results (recovery error and execution time) and visualisation:

$$\|X_{**} - X\|_{l_2} \tag{6.65}$$

We can see that initially each swarm i starts with a random allocation as a solution vector using the pseudo-inverse in Equation (6.41) which is then updated through the iterations of the heuristic based on the current best swarm $X_i^{(t)}$ (carrying the current optimal solution). Note that we do not need to project back to feasible set for correction in the initial solution $X_i^{(0)}$ since the regularised pseudo-inverse guarantees the feasibility of the solution given the constraints. Note also that the randomness (rand) in the initial solution and in the solution generation $0 < R_i^{(t)} < 1$ (vector of small random numbers) which makes every potential solution slightly different for every swarm. This fact represents a positive step size which increases the convergence of the l_0-heuristic. This randomness to the search process where swarms move stochastically within the problem search space in order to find the optimal minimum solution helps the l_0-heuristic to avoid being trapped in the numerous local minima. This is also enhanced by collectively finding and keeping the current best solution at each iteration in a form of weight which is then incorporated in the generation of new solutions at each iteration. By this way every particle can potentially find the optimal solution to the problem, starting from an initial feasible solution which can be seen as a modification of the l_2-norm based sparse solution (analytical solution of the regularised least squares).

At each iteration (t) the current best solution $X_*^{(t)}$ which minimises both the gradient of the objective function $\nabla f_*(\|X_*^{(t)}\|, \sigma^{(t)})$ and parameter $\sigma^{(t)}$ is chosen and each particle's solution is updated based on this current global best solution. The minimum of $\nabla f(\|X\|, \sigma)$ is used as a starting point to locate the minimum for the next (smaller) $\sigma^{(t+1)}$ and to generate the new solution (by ranking the swarms). Note that the generation of a new solution consisted of a gradient step and a misfit step between the current optimal solution and the samples, followed by a correction to the feasible set. If for some values of X we have $|X_i| \gg \sigma$ then the heuristic does not change the value of X_i in that step, though this might change from the projection to the feasible step. Also note that the heuristic forces all values of X_i satisfying $\|X_i\|_{l_2} \ll \sigma$ towards one, while all values of X_i satisfying $\|X_i\|_{l_2} \gg \sigma$ towards

zero. For this purpose we use the function $f(\|X\|, \sigma) = \exp(-\frac{X_i^2}{2\sigma^2}) \leq 1$ for $|X_i| \leq \sigma$. The σ value is initially assigned to twice larger than the maximum value of vector X and then it is gradually decreased at each iteration. This assignment was chosen experimentally based on the nature (elements) of the original vector (assuming values between zero and one).

Moreover, note that σ is a variable depending on the problem and changes through iterations. Initially, if X is not known we can use the back-projection of the samples as a rough estimate, $\sigma^{(0)} = \|C^T Y\|_{l_\infty}$. Otherwise, if X is provided σ value is initially assigned to twice ($\alpha = 2$) the maximum value of the vector X and then it is gradually decreased at each iteration, $\beta = 0.5$ times. This assignment is chosen experimentally based on elements values of test vector and desired accuracy so as $\nabla f(\|X\|, \sigma)$ and σ be positive throughout the iterations. An optional application of a filter on the vector found (e.g. thresholding for smoothness on X_*) [25] can provide some robustness to small noise levels, overcoming difficulties encountered by other sparse recovery methods. Experiments discussed in Section 9 reveal more details about the efficiency of the l_0 heuristic compared to other sparse recovery methods which are based on different approaches and assumptions.

It is also important to pinpoint that every swarm carries a potential current best solution. At every iteration $(t) < T$, S potential solution vectors (swarms carrying out a solution) are generated following similar principle of traditional swarm based algorithms. This concept also bears some similarities with the Firefly algorithm [5, 214]. The whole procedure for S swarms at iteration (t_1) can be described as follows:

swarm 1: $X_{(1)}^{(t_1)} = [X_{1(1)}^{(t_1)}, X_{2(1)}^{(t_1)}, \ldots, X_{N(1)}^{(t_1)}]$
swarm 2: $X_{(2)}^{(t_1)} = [X_{1(2)}^{(t_1)}, X_{2(2)}^{(t_1)}, \ldots, X_{N(2)}^{(t_1)}]$
...
...

[25] A straightforward and simple way to deal with noise is to introduce a thresholding technique which will remove all values of the recovered vector which are above a certain limit. This degradation in values is important as some high numbers might indicate additive noise in the sampling process (non-uniformly sampled areas). For example, in the frequency domain by creating a series of averages between neighboring values of our data we are able to identify and thus eliminate certain patterns in the original data, assuming that our data follow a slow-moving trend in terms of values. In fact, we know that due to the nature of the frequency transform domains, the pattern that elements of a vector follow is; the greater their distance from the origin, the higher their frequency value. This is the major idea behind the moving average filter assuming periodicity of the data. For more details please see Appendix B.

swarm S: $X_{(S)}^{(t_1)} = [X_{1(S)}^{(t_1)}, X_{2(S)}^{(t_1)}, \ldots, X_{N(S)}^{(t_1)}]$,

The efficient number of iterations and swarms chosen can be found experimentally based on the nature of the problem (i.e. size of feasible space, sparsity level, noise level, number of samples). Note that in the experiments the number of swarms and iterations has been set to $S = 12$, $T \in [50, 300]$ for signals ($1D$-vectors) and $S = 10$, $T \in [700, 1000]$ for test images ($2D$-vectors).

Due to the small size of our problems and due to computational limitations in terms of hardware we have not introduced an acceptance function for the number of swarms through iterations. In real life applications an acceptance function could be used to determine the number of swarms. In particular, the number of swarms can decrease as the number of iterations increase as follows [112, 206]:

$$S_{\text{accepted}} = \theta\% \; S(1 + t^{-1}), \tag{6.66}$$

where θ is a parameter set by the user specifying the top performing swarms, S is the number of swarms and t is the number of iterations.

Finally note that a filter is applied after the execution of the l_0-norm based heuristic in noisy cases as a final optional post-processing step for enhancing the performance of the final solution. A filter is simply a function used to remove an undesired component or feature usually caused by noise or aliasing. As a brief reminder the aliasing effect is caused by inadequate sampling frequency or rate of a signal (measuring the value of a signal in specific time so as to collect samples). In essence the samples collected from the values of the signal measured at certain time intervals is not enough to fully recover the original signal. This is a phenomenon called aliasing, as a presence of unwanted components in the recovered from its samples signal (i.e. these components were not present when the original signal was sampled). Also some components of a signal may be lost in the reconstructed signal simply due to noise (no useful information collected) during the sampling process. To remedy aliasing and low levels of noise we can filter out some components of the recovered signal. If this filtering involves the removal of some high frequency components while keeping the lower ones, then we have what we call a "low-pass filter". A filter function simply keeps some components of the signal below a threshold (cut-off frequency) and completely removes any

components above this threshold. In this thesis we will only use some simple low-pass filters for the experiments in noisy sampled signals. For further details see Experiment 10 for filters in Section 9.4.10 and sparse noisy image recovery in Section 9.5 of Chapter 9.

Chapter 7

Algorithms for sparse recovery

Several numerical methods have been proposed in the literature for solving sparse approximation problems, including many methods for obtaining signal representations in overcomplete dictionaries. These range from general approaches, like the Basis Pursuit (BP), Orthogonal Matching Pursuit (OMP) and the method of Matching Pursuit (MP) [35, 177, 196] to more sophisticated ones such as Iterative Hard Thresholding (IHT) and Iterative Reweighted Least Squares (IRLS) methods [19, 22, 56]. The most commonly used methods for compressing and decompressing signals/images are Lasso, Orthogonal Matching Pursuit (OMP) and Basis Pursuit methods. In general, all these methods have both advantages and shortcomings. Orthogonal Matching Pursuit (OMP) is a straightforward method which goes through an iteration process so as to construct the most closely approximation of the input vector (signal) generated as a solution. It is a greedy approach which iteratively tries to find a good estimate of a solution of the sparse approximation problem by selecting atoms of a dictionary and their corresponding weights so that the recovered signal is a linear combination of these vectors with small error. It is very fast but does not always provide good estimation of the solution especially in noisy cases [196, 197]. Another major problem in greedy approaches is that the local optimisation step they use can be erroneous, particularly in cases where a transformed vector (signal or image as a linear combination of atoms) is highly correlated with another vector from the same dictionary [130, 132]. Decomposition in highly correlated and redundant dictionaries (e.g. Gabor wavelet functions in $L^2(\mathbb{R})$ vector space) can cause inconsistencies in solution choices and thus non-optimal representations

[132].

On the other hand, Basis Pursuit (BP) methods are among the most successful ones. They are based on a global criterion which needs to be minimised, avoiding some mistakes made by greedy approaches [132, 215]. They perform a more global optimisation where, for large system of equations, the minimum l_1 norm can replace the minimum l_0 norm as a solution to the ill-posed inverse problem $CX = Y$. Such a solution can be found using several Linear Programming (LP) methods, usually interior-point LP solvers. LP algorithms are able to solve large scale problems with thousands of sources and mixtures but they are still very slow in terms of generating solutions and convergence (see Chapter 9 of experiments for more details). They are also not optimal as in general they can achieve nearly optimal approximations based on the vector support (sparsity) and the dictionary [132]. The Lasso method replaces the sparse approximation problem by a non-convex optimisation (l_2 norm based) problem, so as to effectively solve it. It is usually faster than OMP and IRLS methods, but it's estimation quality is worse, especially if the sparsity level of the solution is large. A comprehensive comparison of many different sparse recovery methods can be found in [110, 183, 196, 218]. In general, typical computations performed by these methods include matrix pseudo-inverses, sparse basis transformations and vector multiplications which in general make some of these methods very computationally expensive, such as the l_1 Magic package (for details see Chapter (9)). In this Section we will briefly describe the aforementioned sparse recovery methods together with some other efficient, yet less known, methods established in the scientific community for solving sparse approximation problems.

7.1 Orthogonal Matching Pursuit method

The Orthogonal Matching Pursuit (OMP) algorithm is a method used for the recovery of a high-dimensional sparse signals, usually based on a small number of noisy linear measurements [35, 69, 177]. It is an iterative greedy algorithm which selects at each step the column which is most correlated with the current residuals. In fact, it is a type of numerical technique which involves finding the "best matching" projections of multidimensional data, such as signals, given that the signal is expressed as a weighted sum of functions (called atoms) using an over-complete dictionary or basis Ψ. Under certain conditions on the mutual incoherence property and the minimum magnitude of the nonzero components of the signal, in a noiseless environment, the original signal can be recovered exactly by the OMP algo-

rithm with high probability even if the nonzero components are possibly small [35]. In this case, using some modified stopping rules, the OMP algorithm will select all the significant coefficients of the signal without any zero components being selected.

The OMP method constructs an approximation by going through an iteration process. At each iteration the locally optimum solution is calculated, by finding the column vector in C which most closely resembles the residual vector r. Initially, the residual vector is equal to the vector X (signal) used as input for approximation and then is adjusted at each iteration based on the vector previously created. The algorithm stops when the required level of accuracy (stopping criterion) is achieved, which is expected to give the global optimum, while the method has been proven to converge with a finite number of iterations in finite dimensional spaces. The algorithm can be summarised on the following pseudocode [35, 69, 177]:

Problem: Determine a vector X s.t. $CX = Y$
Inputs: vector (samples) Y and sampling matrix C
Output: Approximation vector $\hat{X}$
Stopping criterion: till a level of accuracy (ϵ) is reached
OMP Algorithm:
1) Set residual $r_0 = X$, time $t = 0$ and index set $V_0 = 0$
2) Set $v_t = i$, $c_i = max < r_t, c_k >$, c_k: row vectors of C
3) Update the set V_t with v_t: $V_t = V_{t-1} \bigcup \{v_t\}$
4) Solve the least-squares problem:

$$\min_{\hat{X} \in \mathbb{R}^{V_t}} \|X - \sum_{j=1}^{t} \hat{X}(v_j) c_{v_j}\|_{l_2} \tag{7.1}$$

5) Calculate the new residual using $\hat{X}$:

$$r_t = r_{t-1} - \sum_{j=1}^{t} \hat{X}(v_j) c_{v_j} \tag{7.2}$$

6) Set $t = t + 1$
7) Check stopping criterion: $\|X - \hat{X}\|_{l_2} \leq \epsilon$, otherwise go to Step 2

The Orthogonal matching pursuit algorithm is based on an earlier algorithm, called Matching Pursuit (MP) introduced by Mallat and Zhang, which is simply a slight variation [72, 132, 177]. The MP algorithm simply removes the selected column vector from the residual vector at each iteration. Many other improved variations have been introduced; namely, the Stagewise Orthogonal Matching Pursuit (StOMP) [80], which builds the solution by adding several (instead of one) vectors at a time, the Regularised Orthogonal Matching Pursuit (ROMP) [148], which also uses several vectors at each iteration to build the solution, using a threshold, and the Compressive Sampling Matching Pursuit (CoSaMP) [149] which also takes a number of vectors to build the approximation at each iteration. However, the CoSaMP method selects a preset number of vectors from C and generates the approximation based on the given level of sparsity, by removing only the required number of entries. In this thesis, only CoSaMP and OMP algorithms will be used in the experiments (Chapter 9).

7.2 Lasso optimisation method

The aim of this recovery approach is to build an approximation using groups of vectors at a time, like the previously discussed methods while maintaining a relatively fast run time like OMP algorithm. Instead of trying to minimise the Problems in (4.34), (4.29) and (4.1), the Lasso method places a restriction on the solution of the signal reconstruction problem. It is based on a variation of a shrinkage and selection method for linear regression which minimises the usual sum of squared errors, with a bound on the sum of the absolute values of the coefficients so as to enhance sparsity. It can be defined as follows [29, 196]:

$$\min_{\hat{X}} \|C\hat{X} - Y\|_{l_2} \quad s.t. \quad \|\hat{X}\|_{l_1} \leq \lambda \tag{7.3}$$

where, λ is a small constant depending on the nature of the problem (usually representing the sparsity level of the vector), C is the Sensing matrix, Y are the samples and $\hat{X}$ is the unknown estimate of signal. As we can see, the Lasso allows us to find an approximation $\hat{X}$ of the original compressed signal as an optimization principle and not as a recovery algorithm itself. However, there are numerous algorithms to solve these types of problems, such as the SQP method and many other Newton based methods. The Matlab software package CVX and l_1 Magic are some out of many packages commonly used for solving such non-convex optimisation problems.

7.3 Iterative Hard Thresholding method

Iterative Hard Thresholding Algorithm (IHT) is a very simple, yet efficient iterations based algorithm which is very different from the previously discussed algorithms [19, 22, 23]. It is not based on OMP, or Lasso algorithm, as it uses a non-linear operator (P_K) to reduce the value of the l_0 norm at every iteration. In particular, the Iterative Hard Thresholding (IHT) algorithm starts with zero as initial solution and then uses the following iteration:

$$X^{(t+1)} = \{X^{(t)} + C^T(Y - CX^{(n)})\}_{P_K} \tag{7.4}$$

where, Y is the samples vector, C the Sensing matrix, and $X^{(t)}, X^{(t+1)}$ are the (t-th) current and the newly generated ($t+1$-th) solution. $\{.\}_{P_K}$ is a hard thresholding operator that keeps the largest (in magnitude) K elements of a vector and sets all the others to zero, or more generally, it is a projector onto the closest element in the model (i.e. from $\mathbb{R}^N$ to $\mathbb{R}^K$). According to Blumensath and Davies, the algorithm is guaranteed to work and converge to a local minimum of $\|Y - C\hat{X}\|_{l_2}$ if $\|\hat{X}\|_{l_0} \leq K$ whenever $\|C\|_{l_2} < 1$ [19, 20, 22, 23]. They showed that the algorithm has also the following properties:

1. It gives near-optimal error guarantees.
2. It is robust to observation noise.
3. It succeeds with a minimum number of observations.
4. It can be used with any sampling operator (Sensing Matrix C) for which the operator and its Adjoint can be computed (C and C^T).
5. The memory requirement is linear in the problem size.
6. Its computational complexity per iteration is of the same order as the application of the measurement operator or its adjoint.
7. It requires a fixed number of iterations depending only on the logarithm of a form of signal to noise ratio of the signal.

The algorithm can be summarised in the following Pseudo-code [19, 20, 22, 23]:

Problem: Determine a vector X s.t. $CX = Y$
Inputs: Matrix C, vector Y, sparsity level K, number of iterations T
Output: Approximation vector $\hat{X}$
Stopping criterion: till a number of iterations T reached
The IHT Algorithm:
Set $X^{(0)} = 0$
while $t < T$ **do**
  Set $X^{(t+1)} = \{X^{(t)} + C^T(Y - CX^{(t)})\}_{P_K}$
  Set $t++$
end while
Display vector $\hat{X} = X^{(t)}$

Several variations of IHT have been introduced in the literature recently. Accelerated Iterative Hard Thresholding (AIHT) [21] and Normalised Iterative Hard Thresholding (NIHT) [20] are probably the most well-known. In AIHT algorithm the new solution is generated according to the following iteration [21]:

$$X^{(t+1)} = \{X^{(t)} + \mu C^T(Y - CX^{(t)})\}_{P_K}, \tag{7.5}$$

where $\{.\}_{P_K}$ is the hard thresholding operator that sets all but the K largest (in magnitude) elements of a vector to zero and μ is a step size of the method which has to be chosen appropriately to avoid instability and provide a linear rate of convergence for the method. Usually, $\mu = (CC^T)^{-1}$ which guarantees stability and thus enhancing the performance of the algorithm. The NIHT algorithm follows a similar pattern for the generation of a new solution at each iteration [19, 20]:

$$X^{(t+1)} = \{X^{(t)} + \mu^{(t)} C^T(Y - CX^{(t)})\}_{P_K}, \tag{7.6}$$

where, $\{.\}_{P_K}$ is the hard thresholding operator and now μ is an optimal step size which changes at each iteration (in terms of reduction of the squared approximation error between the two vectors $X^{(t+1)}, X^{(t)}$) as follows [19, 20]:

$$\mu^{(t)} = \frac{g_K^T g_K}{g_K^T C_K^T C_K g_K}, \tag{7.7}$$

where, $g = C^T(Y - CX^{(t)})$, while K represents the sparsity level of the unknown vector X (number of non-zero entries). Of course we can assume that K remains the same for all iterations since $\|X^{(t+1)}\|_{l_0} = \|X^{(t)}\|_{l_0} = K$ for all the iterations T.

7.4 Iteratively Reweighted methods

Several methods have been proposed based on non-linear programming, including the famous iterative reweighed methods. In signal analysis, these methods are proven to converge quickly and thus be very efficient for sparse signal recovery. The recovery is based on the l_p re-weighted minimisation principle, for $0 < p < 3$, which is very similar to l_1 and l_2 norm-based minimisation principles. In general, these methods try to recover sparse signals from much fewer measurements, than traditionally believed necessary according to l_1 norm based recovery. This is achieved by a sophisticated approach which reweights the l_2 norm in order to avoid the numerous local minima and thus to efficiently solve the non-convex sparse recovery problem [52, 56]. In essence, a regularisation strategy is used to improve the ability of a re-weighted least-squares algorithm. This approach aims to incorporate the weighted l_2 norm, which is defined as follows [56, 196]:

$$\min_{\hat{X}} \sum_{i=1}^{N} W_i \hat{X}_i^2, \quad s.t. \;\; C\hat{X} = Y, \tag{7.8}$$

where, the weights W_i are calculated based on the previous generated solution so as the objective function is a first order approximation of the l_p objective function ($0 \leq p \leq 1$). In fact, the new solution at k-th iteration is generated as follows [52, 56]:

$$X^{(k)} = Q_n C^T (C Q_n C^T)^{-1} Y, \tag{7.9}$$

where, Q_n is a diagonal matrix with entries $1/W_i = 1/\|X_i^{(k-1)}\|^{(2-p)}$, which is derived from solving the Euler-Lagrange Equation of (7.8); using the constraint to solve the Lagrange multipliers and then substituting this value back to the solution. The whole procedure is repeated up to a number of desired iterations.

A very similar approach is to replace the previous objective function in (7.8) with the

following l_1 weighted problem [45, 196]:

$$\min_{\hat{X}} \sum_{i=1}^{N} W_i \|\hat{X}_i\|_{l_1}, \quad s.t. \;\; C\hat{X} = Y, \tag{7.10}$$

where, the weights are given as follows:

$$W_i = ((X_i^{(k-1)})^2 + \epsilon)^{p/2-1}, \tag{7.11}$$

where, $\epsilon > 0$ is a small constant used to regularise the optimisation problem and avoid the instability of weight in case $X_i^{(k-1)}$ is close to zero. Then an interior linear programming method can be used to solve this problem. Another similar approach for solving the sparse recovery problem transforms the objective function as follows [45, 196]:

$$\min_{\hat{X}} \sum_{i=1}^{N} W_i \|\log \hat{X}_i\|_{l_2}, \quad s.t. \;\; C\hat{X} = Y, \tag{7.12}$$

which actually represents the Euclidean distance between the log of the values of the signal. The steps of the IRLS algorithm can be summarised as follows [52, 56]:

Problem: Determine a vector X s.t. $CX = Y$
Inputs: Matrix C, vector Y, number of iterations T
Output: Approximation vector $\hat{X}$
Update the weights for each iteration $t < T$:
$W^{(t)} = ((X_i^{(t-1)})^2 + \epsilon)^{p/2-1}$.
$k^{(t)} = 1/W_i$.
Solving the weighted l_2 minimisation problem:
$Q_n = \text{diag}(k^{(t)})$.
$R = (CQ_nC^T)^{-1}$.
$X^{(t)} = Q_nC^TRY$.
Termination on convergence otherwise continue iterations.
Output the reconstructed $\hat{X} = X^{(T)}$.

7.5 A variation of Steepest Ascent method

Steepest Ascent algorithms represent a family of maximisation methods where given the direction that solves a certain linearised problem at a point $X^{(t)}$, a new stepwise is computed according to the principle that new point $X^{(t+1)}$ gives better value so as $f(X^{(t+1)}) > f(X^{(t)})$ (maximises the objective function). Recently, G. H. Mohimani et al. [141, 142] have presented a new algorithm for complex-valued sparse representation that provided considerable reduction in complexity. The SL0 (smooth l_0 norm) method is simply a fast variation of a steepest ascent method for finding the sparsest solution of an under-determined system of linear equations, which is based on the minimisation of an approximation of the l_0 norm [141, 142]. Its performance, which was measured against the FOCUSS [116, 144] and L_1 Magic [38] packages, has been found to be two to three orders of magnitude faster while providing approximately the same level of accuracy [141, 142].

This method tries to minimise the l_0 norm by maximising a smoother approximation for sufficiently small σ parameter. This maximisation is done on the affine set of feasible solutions in $CX = Y$. The proper smoothed and easy to differentiate cost function used for the problem to be solved is as follows [141, 142]:

$$\|X\|_{l_0} = N - \lim_{\sigma \to 0} \sum_{i=1}^{N} \exp(-X_i^2/2\sigma^2) \tag{7.13}$$

for any vector $X \in \mathbb{R}^N$, then the objective maximisation becomes:

$$\max_{\hat{X}} \;\; F_\sigma(\hat{X}) \;\; s.t. \;\; C\hat{X} = Y, \tag{7.14}$$

It can be seen that $F_\sigma(\hat{X})$ works as a smooth measure of sparsity of the estimate vector $\hat{X}$ for small values of σ. By choosing a slowly decreasing sequence of σ, the method can escape from getting trapped into local maxima and thus obtain the sparsest solution. The method SL0 can be summarised in the following Pseudo-code [141, 142]:

Problem: Maximise $F_\sigma(X)$ s.t. $Y = CX$
Inputs: function $F_\sigma(X)$, matrix C, vector Y, iterations T
Output: Approximation vector $\hat{X}$
Stopping criterion: till the number of iterations T is reached

The SL0 Algorithm:

Choose an arbitrary solution v_0 from the feasible set, i.e. the minimum the l_2 norm solution of $Y = CX$

Choose a suitable decreasing sequence for the σ parameter, $[\sigma_1, \ldots, \sigma_K]$

for all $k = 1 \ldots K$ (Repeat for each decreasing sequence) **do**

  Set $\sigma = \sigma_k$

  $\max F_\sigma(\hat{X})$ approximately on the feasible set using T iterations (Steepest Ascent method)

  Initialise $s = v_{k-1}$

  for all $t = 1 \ldots T$ (Repeat T times/iterations) **do**

    Set $\Delta X = [X_1 \exp(-|X_1|^2/2\sigma_k^2), \ldots, X_n \exp(-|X_n|^2/2\sigma_k^2)]^T$

    $X = X + \mu \Delta X$ (μ is a small constant)

    Project X back onto the feasibility set: $X = X - C^T(CC^T)^{-1}(CX - Y)$

  end for

end for

Set $v_k = X$

Final solution: $\hat{X} = v_T$

The steepest ascent method consists of iterations of the form $X = X + \mu \nabla F_\sigma(X)$ where the parameter μ is decreasing together with σ. Note also that for smaller values of σ, smaller values of μ also have to be applied, while the stopping criterion of the method is the maximum number of iterations (gradient steps) which have been achieved, usually a small number $T \in [3, 5]$ (found through experimentation) [141, 142]. In fact it has been shown in [142], where the convergence analysis of the algorithm is discussed, that the gradient step loop needs to be repeated for only a small number of iterations since the algorithm is expected to converge to a value very close to the global maximum of the problem, based on the appropriate choice of σ and μ parameters.

Chapter 8

Packages for sparse recovery

These packages solve the sparse recovery problem as an optimisation problem, which can be either convex or non-convex. They are mainly based on interior-point methods (e.g. l_1 Magic, NESTA and CVX) for efficient sparse recovery. Other approaches include a combination of iterative thresholding and gradient projections (TwIST package) and a Bayesian framework that finds the maximum a-posteriori estimator using a concave function and assuming a prior distribution of the unknown coefficients of the signal (FOCUSS package).

8.1 The l_1 Magic package

The l_1 Magic is actually a software package; a collection of MATLAB routines, developed by Emmanuel Candès and Justin Romberg at Georgia Institute of Technology, for solving the convex optimization programming problems relevant to Compressive Sampling. The algorithms are based on standard interior-point methods for solving optimization problems relevant to Compressive Sampling and are suitable for large-scale problems [38, 43]. The signal reconstruction problem with and without noise is formed as a l_1-norm based optimisation problem with linear (noiseless case) and non-linear (noisy case) constraints. The definition of the problem is as follows [38, 42, 43, 48]:

$$\min_{\hat{X}} \|\hat{X}\|_{l_1} \quad s.t. \quad C\hat{X} = Y, \tag{8.1}$$

$$\min_{\hat{X}} \|\hat{X}\|_{l_1} \quad s.t. \quad \|C\hat{X} - Y\|_{l_2} \leq \epsilon, \tag{8.2}$$

where $\hat{X} \in \mathbb{R}^N$ is an estimate sparse vector representing the signal which can be recovered from a small number of noise-free linear measurements $Y = C\hat{X} \in \mathbb{R}^M$ with $M \ll N$ in (8.1) or $Y = C\hat{X} + \epsilon$ for small measurements with bounded noise in (8.2). In both cases this optimisation principle is also known as Basis Pursuit with the smallest l_1 norm, $\|X\|_{l_1} = \sum_i \|X_i\|$. Note that l_1 Magic also solves a similar optimisation problem [48, 87]:

$$\min_{\hat{X}} \|Y - C\hat{X}\|_{l_1} \tag{8.3}$$

which finds the error $Y - C\hat{X}$ with the minimum l_1 norm (i.e. sparsest difference). This problem has high applicability in channel coding and it will not be used for comparison in this thesis.

In case the recovered signal is a 2D image, an alternate recovery model assumes that its gradient is sparse. The software package solves the TV minimisation problem with equality constraints which is actually a Second Order Cone Programming (SOCP see Appendix B) for the noise-free case (8.4) and noisy case (8.5) respectively [38, 39, 42]:

$$\min_{\hat{X}} \mathrm{TV}(\hat{X}) \quad s.t. \quad C\hat{X} = Y, \tag{8.4}$$

$$\min_{\hat{X}} \mathrm{TV}(\hat{X}) \quad s.t. \quad \|C\hat{X} - Y\|_{l_2} \leq \epsilon \tag{8.5}$$

where C is the Sampling/Sensing matrix (the under-sampling Fourier operator F_u), Y is the measurements vector, while TV stands for the Total Variation, which is the sum of magnitudes of the discrete gradient at every point/pixel X_{ij} of an image X with i representing the rows and j representing the columns (assume sparsity in gradients) [38, 39, 42, 43]:

$$TV(X) = \sum_{i,j=1}^{N} \sqrt{(D_{h;ij}X)^2 + (D_{u;ij}X)^2} = \sum_{ij} \|D_{ij}X\| l_2, \tag{8.6}$$

$$D_{h;ij}X = \begin{cases} X_{i+1,j} - X_{ij} & i < N \\ 0 & i = N \end{cases} \quad \text{and} \quad D_{u;ij}X = \begin{cases} X_{i,j+1} - X_{ij} & j < N \\ 0 & j = N \end{cases} \tag{8.7}$$

The l_1 Magic uses a log-barrier method to solve the SOCPs defined in Equations (8.4) and (8.5). It initially transforms the problem into a series of linearly constrained problems and then solves them by forming a a series of quadratic approximations (i.e. a Newtonian

iteration step which proceeds by minimizing each of these systems of equations) [38]. A Matlab implementation, which was used for testing purposes and experimentation in this thesis is available at http://users.ece.gatech.edu/ justin/l1magic. Note that there is also a similar optimization problem solved for sparse image recovery; the Dantzig Total Variation problem [38, 49]:

$$\min_{\hat{X}} \text{TV}(\hat{X}) \quad s.t. \quad \|C^T(C\hat{X} - Y)\|_\infty \leq \epsilon, \tag{8.8}$$

This problem has high applicability in biomedical imaging, analog to digital systems and sensor networks, but it will not be used in this thesis.

8.2 The FOCUSS package

FOCUSS package, which stands for FOCal Under-determined System Solver, is an algorithm designed to obtain sub-optimally sparse solutions to linear inverse problems in relatively noise-free environments [96, 116, 144]. It is an efficient method capable of simultaneously learning over-complete dictionaries and solving sparse inverse problems, based on the ideas drawn from Bayesian estimation theory, nonlinear regularised optimization and convex analysis. In particular, this nonparametric [26] algorithm can reconstruct signals and images from limited data (under-determined number of equations) without any prior knowledge of the structure or shape of the region (signal or image) on which the solution is assumed to be nonzero.

The algorithm has two integral parts: an initial estimate of the signal or image and an iteration process that refines the initial estimate to the final solution. The iterations are based on the re-weighted norm based minimisation of the dependent variable with the weights being a function of the preceding iterative solutions, similar to Iteratively Reweighted Algorithms, discussed in Section 7.4 [96, 116, 144]. Then, the problem is solved using an affine-scaling transformation (allowing negative values for $\hat{X}$) interior point optimisation algorithm which is based on conjugate gradient factorisation (reforming the constraints as $1/2\hat{X}^T C\hat{X} - \hat{X}^T Y$)

[26] With the word/term "non-parametric" we mean an algorithm which is not based on predefined parameters fitting a theoretically motivated function as a best-fit between the projected data (e.g. $\hat{X}$) and the model ($C\hat{X} - Y$). In real life applications data come from different sources without any established pattern or approach, which stress the need to form a more diverse or general methods for obtaining optimal solutions. The nearest neighbourhood algorithm is an example of a non-parametric method [62, 147].

for finding sparse solutions of the following concave function [116, 144]:

$$\min_{\hat{X}} \frac{1}{2}\|Y - C\hat{X}\|_{l_2} + \lambda d_p(\hat{X}), \tag{8.9}$$

where C is the Sampling/Sensing matrix, Y is the measurements vector and $0 < \lambda < 1$ is a regularisation parameter. This reflects the trade-off between the sparse residual $\|Y - C\hat{X}\|$ and the sparse source vector estimate $\hat{X}$ and depends on the compression rate of the sampled vector X.The quantity $d_p(\hat{X})$ corresponds to the following norm [116, 144]:

$$d_p(\hat{X}) = \|\hat{X}\|_{l_p} = \sum_{i,j=1}^{N} \|\hat{X}_{ij}\|_{l_p}, \tag{8.10}$$

for $0 < p \leq 1$ which enforces sparse solutions to the problem. Note that there are also some other dictionary learning algorithms as an extension of the FOCUSS algorithm, such as unit Frobenius-norm prior-based algorithm denoted by FOCUSS-FDL and the column-normalised prior-base algorithm denoted by FOCUSS-CNDL [116]. A more detailed analysis of the theoretical foundation of these optimisation methods together with proofs of global and local convergence is thoroughly discussed in [96, 116]. In the experiments Section we will consider only the main FOCUSS algorithm, discussed here, for sparse image recovery, which bears some similarities with the problems defined in Equations (4.1) and (3.26). A Matlab implementation of this package, which was used for comparisons in this thesis is available at http://dsp.ucsd.edu/ jfmurray/software.htm.

8.3 The NESTA package

NESTA is a fast and robust first-order (based on first derivatives) method, developed by Emmanuel Candès, Jerome Bobin and Stephen Becker at California Institute of Technology, for solving minimum l_1-norm based problems and a large number of extensions including total-variation minimisation used for sparse image recovery [13, 37]. This suite of optimization algorithms is based on ideas from the famous Russian mathematician Yurii Nesterov, namely for accelerated descent methods and smoothing (Basis Pursuit) techniques. The first idea is an accelerated convergence scheme for first-order methods, giving the optimal convergence rate for this class of problems. The second idea is a smoothing technique that

replaces the non-smooth l_1 norm with a smooth version. In particular, the Basis Pursuit optimisation problem that the package solves is the following [13, 37]:

$$\min_{\hat{X}} \|\hat{X}\|_{l_1} \quad s.t. \quad \|Y - C\hat{X}\|_{l_2} \leq \epsilon \tag{8.11}$$

$$\min_{\hat{X}} TV(\hat{X}) \quad s.t. \quad \|Y - C\hat{X}\|_{l_2} \leq \epsilon \tag{8.12}$$

where Y vector represents the measurements, C is the orthogonal measurement matrix, though some other general matrices are also allowed (special cases). Note that Equation (8.11) represents the signal recovery problem while Equation (8.12) represents the total-variation (TV) minimisation problem, which is often used to recover images from noisy and/or under-sampled measurements (see Section 3.7.3). The parameter ϵ is a small constant, proportional to an estimate of the error in the sparse approximation or the standard deviation of any noise in the collection of measurements [13, 37]. This problem is also known as Basis Pursuit (noise-free case) or Basis Pursuit de-noising (noisy case).

Also note that, for real-world signals or images, which may be approximately sparse (i.e. most energy is concentrated in only a few coefficients) we have to use a dictionary Ψ as a basis (Discrete Cosine Basis, a Gabor frame, a curvelet frame, a Wavelet basis, etc.). In this case, NESTA can directly solve this synthesis problem by substituting $\hat{X}$ with $\Psi\hat{X}$ both in the objective and constraints. In the Experiments Chapter 9 we will use NESTA package for sparse signal recovery, solving a slight variation of the problems discussed above (Equations (8.11) and (8.12)) [13]:

$$\|\hat{X}\|_{l_1} + \Omega_p(\hat{X}) \tag{8.13}$$

where the feasible set is defined as $\Omega_p(\hat{X}) = \{\hat{X} : \|Y - C\hat{X}\|_{l_2} \leq \epsilon\}$ with $\Omega_p(\hat{X}) = 0$ if $\hat{X} \in \Omega_p$ and $+\infty$ otherwise. The main idea is to approximate the objective function with a smoother one $f_\mu(\hat{X})$ in a primal feasible set $\Omega_p(\hat{X})$ with $\epsilon = \sqrt{M + 2\sqrt{2M}\sigma}$, σ the standard deviation of noise and M the number of measurements [13, 37]. When μ is zero, $f_\mu(\hat{X})$ is identically to the l_1 norm [13]. For higher accuracy, μ should be set a small value, while if μ is large, the algorithm converges faster. For this purpose the algorithm initially solves the problem with large μ as an initial starting point. The number of continuation steps (iterations) is another parameter of the algorithm that has to be chosen based on the

nature of the problem (usually $T \geq 5$), while the stopping criterion is controlled by another parameter δ. For high accuracy, δ should be small. If μ is small, then δ should be small, but if μ is large, then δ should be larger as well. For the total-variation minimisation problem, the setup is set to an analogous way and μ should be small for high accuracy, or large for faster performance [13, 37]. In the experiments we will use the default options, namely $\mu = \epsilon = 10^{-8}$, $\delta = 0$ and initial solution $X^{(0)} = C^T Y$. NESTA is also capable of solving many other relevant de-noising and sparse recovery problems as an extension of those mentioned above, which are however beyond the scope of this thesis. For further details see [37]. A Matlab implementation of this package, used for comparisons in this thesis is available at http://www-stat.stanford.edu/ candes/nesta/nesta.html.

8.4 The TwIST package

Two-step Iterative Shrinkage/Thresholding Algorithm (TwIST) is a Matlab package for signal and image restoration and linear inverse problems in general. It was recently developed by Jose Bioucas-Dias and Mario Figueiredo at Instituto Superior Técnico, Technical University of Lisbon as an efficient way to handle high-dimensional convex unconstrained optimization problems arising in image restoration and other similar inverse problems (i.e. wavelet based deconvolution) under non-quadratic convex regularisation (i.e. Total Variation) [16–18]. It extends and improves the convergence speed of Iterative Shrinkage or Thresholding (IST) algorithms which have been proposed for solving severely ill-conditioned problems. Two-step Iterative Shrinkage/ Thresholding TwIST algorithm overcomes this shortcoming by implementing a nonlinear two-step (also known as "second order") iterative version of IST [17, 18]. The resulting algorithms exhibit a much faster convergence rate than IST for ill-conditioned and ill-posed problems.

IST methods approach the signal/image processing problem of Compressive Sampling method as a minimiser of a non-convex and sometimes non-smooth objective function $f : \hat{X} \to \bar{\mathbb{R}} = [-\infty, +\infty]$ [16–18].

$$f(\hat{X}) = \frac{1}{2}\|Y - C\hat{X}\|_{l_2} + \lambda\Phi(\hat{X}), \tag{8.14}$$

where Y is the observed data, C is the linear direct operator (Sensing matrix), $\lambda \in [0, +\infty)$ is a parameter, usually calculated as $\|C^T Y\|_{l_\infty}$ and $\Phi(\hat{X})$ is a regulariser (TV norm for

images, see Section 3.7.3 and l_1 or l_0 norm for signals). Minimising the function $f(\hat{X})$ is an efficient way to overcome the singular or ill-conditioned nature of C with λ the so-called regularisation parameter which controls the relative weight of the two terms; namely between the lack of fitness of a candidate estimate $\hat{X}$ to the observed data Y (measured by $\|Y - C\hat{X}\|_{l_2}$) and its degree of undesirability, given by $\Phi(\hat{X})$. However, all the IST algorithms have a very slow rate of convergence, as they heavily depend on this linear observation operator C, becoming very slow when this operator is ill-conditioned or ill-posed. The Two-step Iterative Shrinkage/Thresholding (TwIST) algorithm overcomes this drawback by implementing a nonlinear two-step iterative minimiser, which is highly applicable to a vast class of non-quadratic convex regularisers (vector norms and total variation). Results and proofs in [17, 18] show that TwIST converges to a minimiser of the objective function, for a given range of values of its parameters, while for non-invertible observation operators a monotonic version of TwIST (MTwIST) is introduced with experimental evidence that MTwIST exhibits similar speed as IST [17, 18].

8.5 The CVX package

CVX is a general software or package in Matlab, developed by Michael Grant and Stephen Boyd at Stanford University, for solving convex optimisation and other related problems, such as geometric programming (GP), through the use of a special mode [28, 97, 130]. Constraints and objectives are expressed using some certain rules similar to those of Matlab and then they are automatically transformed to a canonical form (i.e. standard LP form) and solved using interior-point methods as solvers. This makes CVX a reliable and accurate system for small and medium-sized problems and not so capable for extremely large-scale problems [97, 130]. The recovery of vectors is not based on any specific sparse recovery method but it is a front-end implementation that uses two general optimisation solvers, namely SeDuMi and SDPT3, which are also written in Matlab.

SeDuMi solver is the one used by default and the one we used for the test runs (see Chapter 9). It is a software package to solve optimization problems over symmetric cones, which includes linear, quadratic, second-order conic and semi-definite optimization problems, together with any combination of these [162, 187]. SeDuMi stands for Self-Dual Minimisation and implements a self-dual embedded technique which usually involves solving large-scale linear problems by an efficient extension of interior point methods; namely the famous Se-

quential Quadratic Programming method (SQP) [28, 162, 187]. This primal-dual (meaning that both the primal and the dual programs are solved simultaneously) interior-point method involves a number of preprocessing steps before the main steps of the algorithm begin to iterate. It can be considered as a Newton-like method, which supports a particular type of convex optimization that we call disciplined convex programming [97, 130]. According to this approach, convex functions and sets are built up from a small set of rules from convex analysis, starting from a base library of convex functions and sets. Constraints and objectives that are expressed using these rules are automatically transformed to a canonical form and solved (for more information see [28, 97, 130]).

CVX also supports geometric programming (GP) through the use of a special GP mode. Geometric programs are not convex, but can be made so by applying a certain transformation (see Appendix B for definitions). In this mode, CVX allows GPs to be constructed in their native, nonconvex form, transforms them automatically to a solvable convex form, and translates the numerical results back to the original problem [97, 130]. The latest version of CVX (version 2.0 as of June 2013) also brings support for mixed integer disciplined convex programming (MIDCP). In this thesis, we will apply CVX only to Linear Programming (noiseless case) and Quadratic Programming (noisy case) problems as a sparse recovery problem, defined as follows:

$$\min_{\hat{X}} \ \|\hat{X}\|_{l_1} \quad s.t. \ \ C\hat{X} = Y, \tag{8.15}$$

$$\min_{\hat{X}} \ \|\hat{X}\|_{l_1} \quad s.t. \ \ \|C\hat{X} - Y\|_{l_2} < \epsilon, \tag{8.16}$$

where $\hat{X}$ is the unknown (estimate) signal in vector format, C is the Sensing matrix and Y is the samples vector and ϵ is a small constant which models the observation error in the noisy case (8.16). A Matlab implementation of the CVX package, used for comparisons in this thesis is available at http://cvxr.com/cvx/, while some general description together with a Matlab implementation of SeDuMi can be found at http://sedumi.ie.lehigh.edu/.

Chapter 9

Experiments - Simulations

This Chapter describes and discusses the simulations and experiments conducted so as to test the efficiency and the effectiveness of the proposed l_0 heuristic for sparse signal and image recovery. Its performance is experimentally verified and compared with alternative algorithms and packages discussed in Chapters 7 and 8. A complete description of the effects of the parameters of the heuristic, it's convergence properties and how the sparsity level, the noise level, the measurements size and the types of Sensing matrices affect it's performance are also presented and experimentally tested. The effectiveness of the heuristic is experimentally confirmed on problems of sparse image and signal representation and restoration with missing samples (both test signals and images). Finally, the complexity of the heuristic in terms of execution time is also discussed.

In all the experiments conducted sparse signals were generated randomly in a form of vector using the Gaussian model for the noiseless cases, while the Gaussian model or the Bernoulli-Gaussian model was used for the noisy cases. In the case of images the noise level was also introduced using the Gaussian model. This is common practice for sparse signals and images referred in the bibliography; see for example [42, 76, 132, 141, 186, 215]. Moreover, in the tests conducted in signals (apart from the experiment discussed in Section 9.4.8) the Sensing matrix was chosen as a 2D vector whose values were created randomly in the interval between 0 and 1 using the Gaussian distribution with normalised columns. For the tests in images we assume the Sensing matrix represents the partial Fourier (DFT) coefficients, obtained by randomly adding less than half the rows of the Fourier (DFT) Basis.

All the numerical experiments discussed in this Chapter were performed on an Dell Intel(R) Core(TM) $i5$ CPU (3.20 GHz) with 4 GB RAM, using Matlab R2014b under Microsoft Windows XP Professional. All the practical experiments in signals (apart from the case of images discussed in Section 9.5, transform Domains discussed in 9.4.9 and the Filters Section discussed in 9.4.10) the quantitative performance of the competing methods (l_0 heuristic and other alternative sparse approximation methods) was performed based on a series of experiments (trials) which follow the following pattern:

1. Randomly generate a signal X as a vector with finite length which is fixed throughout the experiment.

2. Select the support set (sparsity level) and sample the vector X randomly using different sample sizes based on the type of the experiment. The entries of the Sensing matrix C follow the standard Normal distribution, i.e. $C \sim \mathcal{N}(0,1)$, whose columns can be scaled so as to have unit l_2 norm (the normalisation step is optional).

3. Formulate and store the sample vector $Y = CX + e$ ($e \sim \mathcal{N}(0,1)$), setting $e = 0$ for noiseless cases.

4. Repeat the experiment for each sparsity level, sample size or noise level. Calculate the average time and recovery error for each sparse recovery method.

Note that due to the stochastic (random) nature of the experiments (i.e. random generation of initial signal X, randomly generated Sampling matrix C and random step in solution calculation of the l_0 heuristic) every test run will produce slightly different results, resulting to somewhat different calculations, plots and evaluations of error metric and execution time. Also due to the nature of how $rand()$, $randi()$ and $randn()$ are defined in Matlab, for generating random vectors, sometimes the l_0 heuristic stalls and fails to efficiently recover the initial vector. This is a common problem with random vectors discussed also in other packages such as CVX, which uses gradient based algorithms [130]. This problem can be attributed to the mistakes in the calculation of the Euclidean distance of the residuals ($\|C\hat{X} - Y\|_{l_2}$), which is propagated through the iterations of the sparse recovery method. This wrong distance information between some points in the Euclidean space exists due to the nature of the ill-determined system $C\hat{X} = Y$, where the generated solution $\hat{X}$ is not uniquely

determinable and due to the fact that the columns of the Sampling matrix are sometimes correlated to each other and to the original randomly generated vector X (mutual coherence is high). Euclidean space ($\mathbb{R}^N$) is simply a finite dimensional real vector space with inner-product and distance metric defined on it. However, in cases where the sampling matrix is normalised (of unit norm), the initial vector is not completely random or has complex values (e.g. $f(t) = e^{it}$, application of a unitary transform, or we sample images) this problem is less frequent and tends to disappear through test runs. Further details about this problem can be found in [65, 132] where this drawback is analysed for greedy and convex iterative approaches, respectively, used for solving ill-posed inverse problems.

9.1 Test signals

An initial and straightforward way to create a random signal is to generate a sequence of randomly generated values in vector format using the Gaussian distribution. The test vector of N elements, as a randomly generated signal can be defined as follows:

$$X = [X_1, X_2, \ldots, X_N], \tag{9.1}$$

where X_i ($i = 1, \ldots, N$) represents an element of vector X at position i, whose value is randomly generated using the Gaussian distribution with zero mean and one as variance.

For further testing the efficiency of the heuristic a random signal can also be generated of the following form, which represents a general yet simple one-dimensional discrete time sinusoidal model:

$$X_t = \sum_{j=1}^{M} W_j \sin(\omega_j t) \tag{9.2}$$

where W_j is the amplitude and ω_j the frequency of the model. We also assume zero phase for simplicity in calculations. Note that both of these parameters are not fixed constants but uncorrelated independent randomly generated real numbers using the Gaussian distribution. This is very important so as to apply the CS theory and measure the process and efficiency of the algorithms.

An alternative way to generate a sparse signal under the presence of noise is to use the

Bernoulli-Gaussian model, where each element x_i of the vector (signal) X is "active" with probability p and inactive with probability $1-p$. If it is active, it means that it is a zero-mean Gaussian random variable with variance σ_a^2, while if it is inactive then the element of the vector is a zero-mean Gaussian random variable with variance σ_i^2 and $\sigma_a^2 \ll \sigma_i^2$. This can be modelled as follows [141, 142]:

$$X_i \sim p\mathcal{N}(0, \sigma_a^2) + (1-p)\mathcal{N}(0, \sigma_i^2), \tag{9.3}$$

where p denotes the probability of activity of the vector elements X_i, with sparsity implying that $p \ll 1$, while σ_i^2 represents the noise in the elements or small value of the sparse elements in their inactive case. This realistic model represents the signals in telephone (copper) network lines, where elements in their inactive case are not exactly zero but proportional to the noise level in the lines. Obviously in the noiseless case we can set σ_i^2 to zero. We assume only additive Gaussian noise in this model which can model the sensor noise or the decomposition inaccuracy.

A different type of randomly generated signals with real and complex values, with and without noise, which has also been used for testing the recovery error and the rate of convergence of methods. The signal can be modelled as [57, 108]:

$$s(t) = s_0 \exp(-kt), \tag{9.4}$$

where $s(t)$ the signal value at time t, s_0 is the initial value of the signal at time 0 and k is a small random number to accelerate the decay of the signal's value through time. Of course nothing is measured perfectly, so k also represents some random variation which should be allowed. Note that this model simulates the theoretical property of perfectly sparse signals whose values decay through time (i.e. time dependent relationship). We assume that a random signal is sampled and compressed at a specific time interval for more realistic results. For this purpose a signal is modelled as a function of time where every signal value derived from assigning a time value to the function will represent an entry vector format for that time.

A statistical model for generating signals under the presence of noise is [57, 108]:

$$x(t) = \alpha s(t) + e, \tag{9.5}$$

where $s(t)$ is an observable random quantity that changes through time (i.e. using Equation (9.4)), α is an unobservable unknown parameter randomly generated, e is the error or noise term, having a normal distribution $\mathcal{N}(0, \sigma^2)$ and the variance σ^2 being a further unknown parameter randomly generated. Thus, if we set $e = 0$ then the signal is modelled in a noiseless environment. In general, this probabilistic nature of this model reflects its stochastic uncertainty in terms of generating values of $x(t)$. In general, we assume that the sparsity level affects the number of non-zero entries of the signal (vector); high level of sparsity means small number of non-zero entries.

9.2 Test images

A test image is a standard digital image file used across different researchers and institutions to test image processing, classification, segmentation and compression algorithms [7, 64, 125, 201]. By using the same standard test images, different researchers are able to compare more efficiently their results both visually and quantitatively. Usually, these images are chosen to represent a natural image and exhibit some particular properties, such as uniform regions or sharp transitions/edges, required for testing the efficiency of a image reconstruction or a general image processing technique.

There is a number of test images used for scientific testing purposes widely available through on-line repositories [7, 64, 125, 201]. In this thesis, we will use only three of them which are considered to be the most representative ones for scientific testing purposes and have been widely used in image processing literature; Phantom, Circles and Checkerboard. All of them belong to the standard synthetic images for testing in Matlab which are also publicly available from the image repositories in [7, 125, 201]. The aim of choosing them is to provide a common, yet diverse, testing framework for image processing methods so as the results can be easily compared. These images can be easily imported into Matlab environment as grayscale images represented using vectors. Any colour image can also be converted to grayscale image if needed. As Shapiro noted in his paper [178] about the zerotree wavelet (EZW) algorithm, there are many options for transforming a colour image to gray-level including taking only the green component of the RGB representation and using a luminance-only version or the most obviously by taking the average of the RGB components.

There are many variations of test images available in the web with differences between them due to cropping, scanning, resizing, compression or even conversion from color to gray-level [125, 201]. We shall use only 128×128 grayscale version of all these images, widely available as TIFF, JPG or PNG file format. This resolution (i.e. simply the measurement of a pixel plane) has been chosen mainly for simplicity reasons in calculations and the time required to process these images. In general, the resolution is measured in pixels or megapixels. To a certain degree,it can be a major factor in determining image quality, as with more pixels you can maintain a good, sharp image even if you enlarge it or crop it; a higher resolution picture keeps the visible degradation low. However, greater in resolution images would cause an "out of memory" error, mainly due to the amount of virtual memory Matlab can use, depending on the software and hardware specifications of the machine running Matlab [73]. Note also, that among the given test images, the Circles image is actually a binary image, with only two light intensity values (0 and 1 representing black and white respectively) [7, 64, 125, 201].

9.3 Recovery error metrics

In order to test the efficiency and robustness of the proposed heuristic we have also introduced the calculation of the error of recovery of signals or images. The mean square error (MSE) as a signal/image recovery error is an absolute (exact) metric commonly used and easy to calculate [132, 157, 186]:

$$MSE = \|X - \hat{X}\|_{l_2}, \tag{9.6}$$

where vectors X and $\hat{X}$ which represent the original and approximated (or recovered) signal/image respectively. The l_2 norm represents the Euclidean distance between the two vectors with N elements defined as:

$$\|X - \hat{X}\|_{l_2} = \sum_{i=1}^{N} \sqrt{(X_i - \hat{X}_i)^2} \tag{9.7}$$

for images the l_2 norm is defined slightly differently:

$$\|X - \hat{X}\|_{l_2} = \sum_{i=1}^{N} \sum_{j=1}^{M} \sqrt{(X_{i,j} - \hat{X}_{i,j})^2}, \tag{9.8}$$

where i, j is the pixel location of an $N \times M$ image, X and $\hat{X}$ is the original and recovered image. In the experiments we assume that $M = N$ (images are represented as square matrices).

An alternative definition for the MSE defined above can be as follows:

$$r = \|CX - C\hat{X}\|_{l_2}, \tag{9.9}$$

where C is the Sensing or Sampling matrix used for the compression of the signal/image. This metric is used if only the samples $CX = Y$ of the original vector X are known. Note that in this case r could be zero even if $X \neq \hat{X}$.

Another slight variation of this metric which uses the Euclidean distance as a relative recovery error metric in the numerical solution of the model is [11, 39, 42, 45]:

$$r = \frac{\|X - \hat{X}\|_{l_2}}{\|X\|_{l_2}}, \tag{9.10}$$

where X and $\hat{X}$ is the original and the recovered vector. This metric is used in cases we want to keep the estimation error within a certain range so as to calculate the tradeoff between approximation error and the estimated terms more efficiently.

For the noisy cases, in signals, we can use the Signal-to-Noise ratio (SNR) defined as [136, 157, 186]:

$$SNR = 20 \log_{10} \frac{\|X\|_{l_1}}{\|X - \hat{X}\|_{l_1}}, \tag{9.11}$$

where l_1 norm is simply the sum of absolute values for a vector and the absolute difference between the elements of the original vector X and its estimate $\hat{X}$. This is mainly used in noisy cases as an efficient comparison between the vector and the noise introduced in the measurement. It measures how much a vector is distorted by noise; the lower the noise, the

higher the SNR and the better the recovery.

For noisy images we can use the peak Signal-to-Noise ratio (PSNR) as a criterion of the quality of the recovery. It is defined as follows [99, 111, 136, 186]:

$$PSNR = 20\log_{10}(Z/MSE), \tag{9.12}$$

where MSE is the MSE metric defined in (9.6) and Z is a single value that represents the maximum entry or element of the original 2D vector X. This is a very common unequal weights relative metric in images used to calculate the ratio between the maximum entry of a vector and the quantity of corrupting noise affecting the image. It is very efficient in images as their values range widely (difference between the smallest and largest entry is great) meaning that the higher the PSNR, the better the recovery as $\hat{X}$ is very similar to X and thus the reconstructive method is efficient.

9.4 Experimental results in signals

9.4.1 Sparse recovery using the l_0-heuristic

We present two simple numerical experiments which will illustrate how the l_0 heuristic works and will provide some insights about the l_0-heuristic's performance. In the first example, we sample an initial 300 element vector with real values randomly generated using the Normal distribution, while in the second example we sample the same 300 element vector which now has complex values randomly generated by the Normal distribution. In both cases we work with sparsity level $K = |\{i : X_i \neq 0\}| = 30$ (X has 30 non-zero entries with values between 0 and 1) and a Sensing matrix C with only $M = 180$ rows (samples collected) and $N = 300$ columns, which are independent un-normalised Gaussian entries (the columns of C do not have unit-norm). The samples Y are formed based on the model $Y = CX + \epsilon$, where $\epsilon = 0$ (noiseless environment). The results of the Numerical Experiments are presented in the following four Figures; Figure (9.1) represents the sparse recovery of a real-valued sparse vector (initial and recovered elements of a vector), Figure (9.2) represents the error between the elements of the l_0-norm based estimate and the original real-valued sparse vector, Figure (9.3) represents the sparse recovery of a complex-valued sparse vector (initial and recovered elements of a vector) and Figure (9.4) represents the error between the elements of the

l_0-norm based estimate and the original complex-valued sparse vector.

It can be observed that the l_0 heuristic correctly identifies all the non-zero components of X and correctly sets all the others to zero. Clearly the method exhibits a hard-thresholding type of behaviour which quantitatively speaking achieves small recovery error $\|X - \hat{X}\|_{l_1} \approx 10^{-15}$, for both real and complex-valued vector, after 300 iterations using 12 swarms. Smaller number of iterations $(150 - 200)$ with the same number of swarms achieve slightly worse results with typical error between 10^{-8} and 10^{-10}. Both of the experiments took roughly 3 seconds on a high-end PC (Intel(R) Core(TM) $i5$ CPU (3.20 GHz) with 4 GB RAM) using Matlab R2014b, though the experiment with complex numbers took slightly longer (2 seconds instead of 1 second for real numbers). This is expected as the complex vector is decomposed into real (Real) and imaginary (Imag) components (for efficient recovery) which share the same sparsity pattern (the non-zero entries appear at the same position).

If a vector has complex values, which can represent phase encoding data (e.g. MRI or PET measurements), then the real and imaginary parts have to be recovered separately by re-arranging the real and imaginary parts into a real vector. If $X \in \mathbb{C}^N$ then by applying a simple operator R we can re-arrange this complex-valued vector into a real vector as [75, 168, 179]:

$$R(X) \triangleq \begin{bmatrix} \text{Real}(X) \\ \text{Imag}(X) \end{bmatrix}, \tag{9.13}$$

where

$$X \triangleq \text{Real}(X) + \text{Imag}(X)i. \tag{9.14}$$

In this case we assume that vector X is strictly complex which means that its non-zero elements have both real and imaginary parts, which implies [75, 168, 179]:

$$\|R(X)\|_{l_0} = 2\|X\|_{l_0}, \tag{9.15}$$

where $\|.\|_{l_0}$ norm counts the sparsity level (non-zero entries) of a vector. In this case the

sparse recovery as a l_0-norm based problem becomes:

$$\min_{\hat{X}} \|R(\hat{X})\|_{l_0} \text{ s.t. } R(C\hat{X}) = Y(= R(CX)). \tag{9.16}$$

Note since $X \in \mathbb{C}^N$ is re-arranged into $R(X) \in \mathbb{R}^{2N}$, we also need to re-arrange $C \in \mathbb{R}^{M\times N}$ into $R(C) \in \mathbb{R}^{2M\times 2N}$. Also notice that obviously $R(CX) = R(C)R(X)$. This is a common approach adopted by other sparse recovery methods, such as TwIST, l_1-magic and CVX, where the TV norm of the sparse recovery problem is minimised as a sum of norms [16, 38, 42, 168]:

$$\begin{aligned} \min_X \|X\|_{TV} &= \min_X \sum_{i,j=1}^{N} \sqrt{\text{Real}((X_{i+1,j} - X_{i,j})^2 + (X_{i,j+1} - X_{i,j})^2)} \\ &+ \sqrt{\text{Imag}((X_{i+1,j} - X_{i,j})^2 + (X_{i,j+1} - X_{i,j})^2)} \end{aligned} \tag{9.17}$$

This is a very common approach in sparse image recovery, which we will also adopt in this experiment (for more details about how this approach is adopted in sparse image recovery, please refer to Section 9.5). In all the other experiments, excluding those in images, we will adopt a slightly different approach where the real and imaginary parts of a complex-valued vector (e.g. test signal) are sampled and recovered separately as two vectors instead of one vector. Although the outcome is the same as with the approach discussed here, we use this alternative approach so as to measure the computation time of the l_0 heuristic as a worst case scenario (i.e the computational cost of recovering two vectors separately is greater than recovering one vector twice its original size). This approach is also convenient in terms of memory efficiency using Matlab, since the Sensing matrix used for sampling both real and imaginary parts is the same, $C \in \mathbb{R}^{M\times N}$ and half the size of the matrix used here, $R(C) \in \mathbb{R}^{2M\times 2N}$. In this similar case the model for sparse image recovery is defined as:

$$\min_{\hat{X}} \|\text{Real}(\hat{X})\|_{l_0} \text{ s.t. } C\text{Real}(\hat{X}) = \text{Real}(Y) \ (= C\text{Real}(X)) \tag{9.18}$$

$$\min_{\hat{X}} \|\text{Imag}(\hat{X})\|_{l_0} \text{ s.t. } C\text{Imag}(\hat{X}) = \text{Imag}(Y) \ (= C\text{Imag}(X)), \tag{9.19}$$

where $C \in \mathbb{R}^{M\times N}$, $\text{Real}(\hat{X}) \in \mathbb{R}^{N\times 1}$, $\text{Imag}(\hat{X}) \in \mathbb{R}^{N\times 1}$, $\text{Real}(Y) \in \mathbb{R}^{M\times 1}$, $\text{Imag}(Y) \in \mathbb{R}^{M\times 1}$.

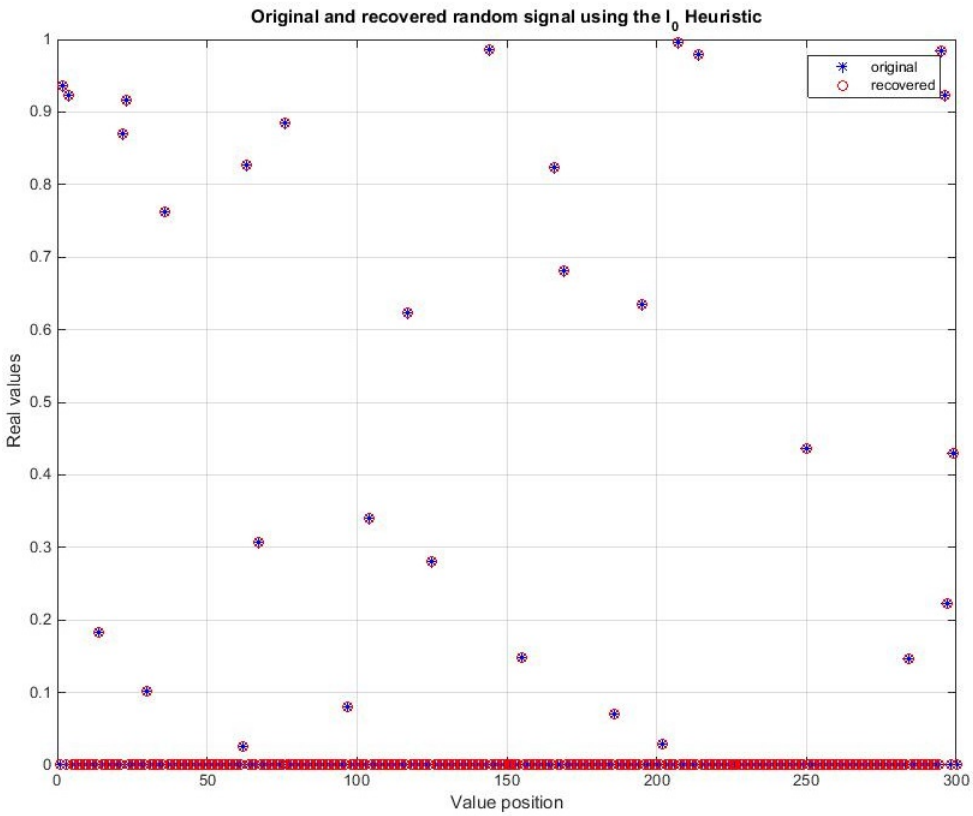

Figure 9.1: Estimation of a simple one-dimensional 300-element real valued sparse vector using l_0 approximate minimisation with equality constraints, with C a 180 by 300 sampling matrix with independent Normal distribution entries. The blue star indicates the original (true) values of the vector and the red circle the estimate using the l_0-norm based approximation optimisation principle solved by the l_0 heuristic. In this example, $\sigma^{(0)} = 2X_{\max}$, $T = 300$ iterations, $S = 12$ swarms, sparsity $K = 30$ and σ decrease factor $\beta = 0.5$ (example0.m).

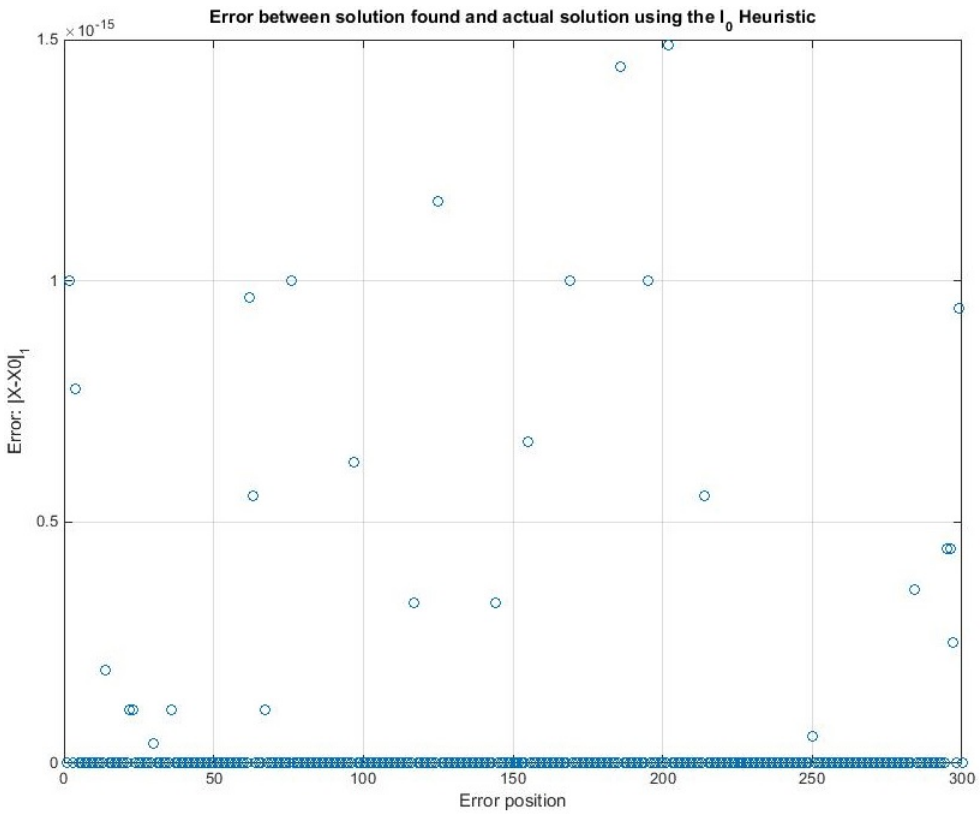

Figure 9.2: Absolute value of the recovery error $\|X - \hat{X}\|_{l_1}$ between 300-element real valued sparse vector X and its estimate $\hat{X}$ using the l_0-norm based approximation optimisation principle solved by the l_0 heuristic. In this example, $C = \mathbb{R}^{180\times 300}$ is the sampling matrix with independent Normal distribution entries, $\sigma^{(0)} = 2X_{\max}$, $T = 300$ iterations, $S = 12$ swarms, sparsity $K = 30$ and σ decrease factor $\beta = 0.5$. Observe that although the error is small, due to efficiency of the l_0 heuristic, it is not absolutely zero due to the nature of CS technique (example0.m).

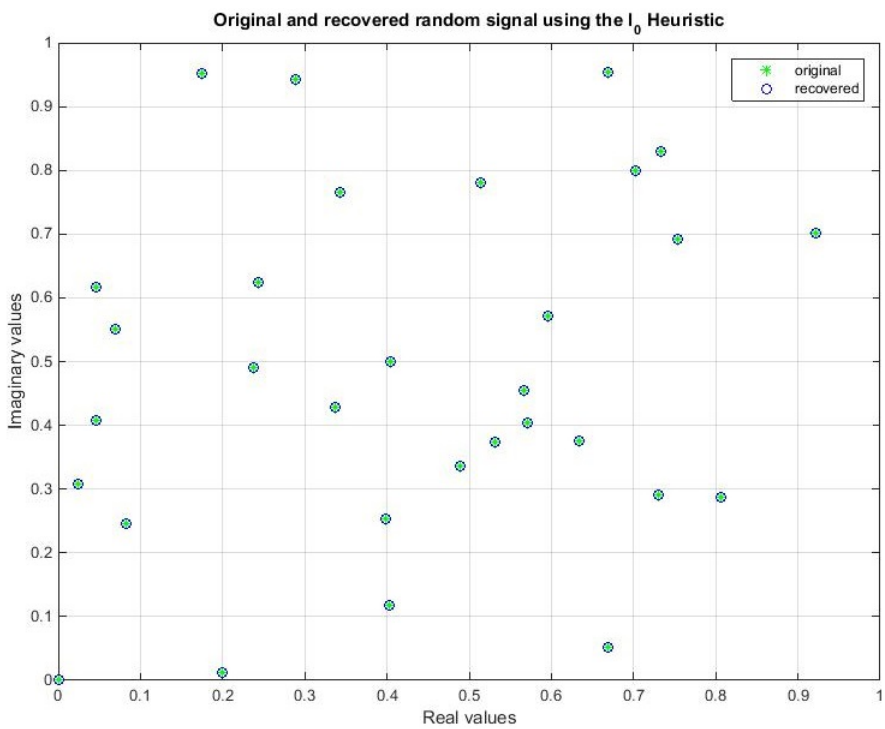

Figure 9.3: Estimation of a simple one-dimensional 300-element complex valued sparse vector using l_0 minimisation with equality constraints, with C a 360 by 600 sampling matrix with independent Normal distribution entries. The blue star indicates the original (true) values of the vector and the red circle the estimate using the l_0-norm based approximation optimisation principle using the l_0 heuristic. In this example, $\sigma^{(0)} = 2X_{\max}$, $T = 2*300$ iterations, $S = 12$ swarms, sparsity $K = 30$ and σ decrease factor $\beta = 0.5$. Note that since the sparse vector is complex, the heuristic decomposes it into real and imaginary parts which recovers separately (example00.m).

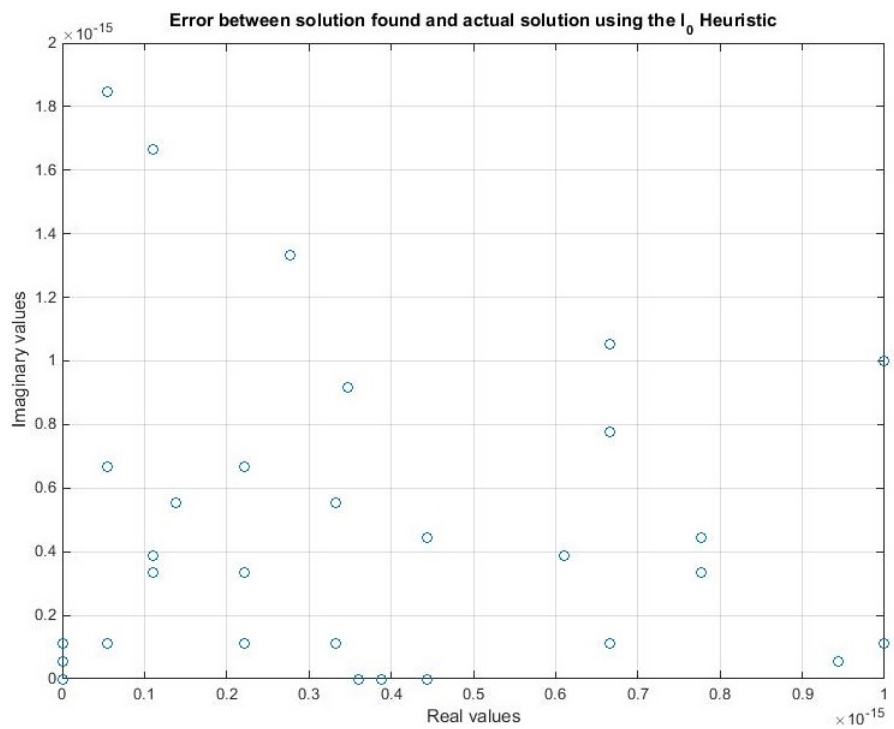

Figure 9.4: Absolute value of the recovery error $\|X - \hat{X}\|_{l_1}$ between 300-element complex valued sparse vector X and its estimate $\hat{X}$ using the l_0-norm based approximation optimisation principle solved by the l_0 heuristic. In this example, $C = \mathbb{R}^{360\times600}$ is the sampling matrix with independent Normal distribution entries, $\sigma^{(0)} = 2X_{\max}$, $T = 2 * 300$ iterations, $S = 12$ swarms, sparsity $K = 30$ and σ decrease factor $\beta = 0.5$. Observe that although the error is small, due to efficiency of the l_0 heuristic, it is not absolutely zero due to the nature of CS technique (example00.m).

9.4.2 Performance analysis

In this experiment we study the execution time and recovery error of the l_0 heuristic as iterations increase through observation. We artificially generate a simple real-valued signal ($f(t)$) in vector format of 70 elements and 8 non-zero entries, namely $f(t) = 2\sin(t) + 8\sin(2t) + 14\sin(4t) + 16\sin(5t) + 20\sin(12t) + 22\sin(30t) + 25\sin(50t) + 31\sin(70t)$ at positions (assigning $t = 1$ to derive the samples),

(9.20)

Position	Element
1	$f_1(t) = 2\sin(t)$
2	$f_2(t) = 8\sin(2t)$
4	$f_4(t) = 14\sin(4t)$
5	$f_5(t) = 16\sin(5t)$
12	$f_{12}(t) = 20\sin(12t)$
30	$f_{30}(t) = 22\sin(30t)$
50	$f_{50}(t) = 25\sin(50t)$
70	$f_{70}(t) = 31\sin(70t)$

In this experiment two different types of measurements of $M_1 = 40$ and $M_2 = 12$ samples are used, both of which are created using a Sampling matrix C with elements drawn from the Normal Distribution and values between zero and one (un-normalised entries). The l_0 heuristic's parameters are $T = 300$ iterations, $S = 12$ swarms, σ decrease factor $\beta = 0.5$ and initial $\sigma^{(0)} = 2\max(f(t))$ (twice the maximum entry). CPU cycles have been used as a rough estimate of execution time in seconds and the relative MSE $\|\hat{f}(t) - f(t)\|_{l_2}/\|f(t)\|_{l_2}$ as the recovery error metric. Figure (9.5) shows the execution time of the heuristic in CPU cycles for 300 iterations of the heuristic. Notice the (approximately) linear complexity of the l_0 heuristic in terms of time (linear execution time in terms of the number of iterations). The small perturbations in the line plotted can be attributed to the stochastic nature of the proposed method and the approximation of time by the CPU cycles of the $i5$ Intel processor.

Figure (9.6) presents the quality of the estimated vector in terms of recovery error through iterations for two different sample sizes; one under-sampled ($M_1 = 40$) and one highly under-sampled ($M_2 = 12$). Note that for highly under-sampled cases a smaller number of iterations is required for efficiently recovery, roughly two-three times the sparsity level of the recovered

vector. In particular, it can be seen from Figure (9.6) that after roughly 25 iterations it is not possible to achieve relative recovery error smaller than 0.5 for highly under-sampled cases ($M_2 = 12$). However, in under-sampled cases where samples are roughly half the length of vector ($M_1 = 40$) possible improvement of the solution is possible until the heuristic reaches 50 iterations where further improvement is not possible. The solution of the heuristic still improves exponentially between 10 and 50 iterations for 40 samples; the recovery error after 10 iterations is 10^{-1}, while after 50 iterations we obtain 10^{-15} as a recovery error. This is expected as the null space of the sampling matrix C, $Null(C) = \{X : CX = 0\}$, does not allow efficient invertible mapping from M back to N elements, as important information about the structure of the sampled vector is missing, resulting to infinite many possible solutions to the same under-determined linear system. However, even in these cases, the heuristic's performance, in terms of solution quality, is superior compared to other sparse recovery approaches, as we will see in a later experiment, due to the fact that it does not require the RIP/UUP properties to hold in terms of measurements (see Sections 4.3, 9.4.5 and 9.4.6 for further details).

9.4.3 Dependence of parameters

In this experiment we study how the performance of the l_0 heuristic is affected by its parameters. In particular we test different initial solution approaches, different σ decrease factors β and target values $\sigma_{\min}$, different weights in the generation of new solutions and finally we plot the distance between the original (sampled) vector and its estimate through iterations (different error metrics are plotted through iterations) as an efficient way to visualise the performance of the l_0 heuristic, used for solving the approximating l_0-norm based method. Table (9.1) summarises the outcome of experimental results conducted for different initial solutions under different sample sizes, $M = 30, 40, 70, 100, 150, 200, 250$. It represents the average recovery error for 100 test runs for each sample size, using a real-valued random vector of 200 elements and 20 non-zero entries (sparsity level) generated using the Normal distribution. The initial solutions tested are:

1. Back-Projection (Option 1):

$$X_{\text{init}} = C^T Y. \tag{9.21}$$

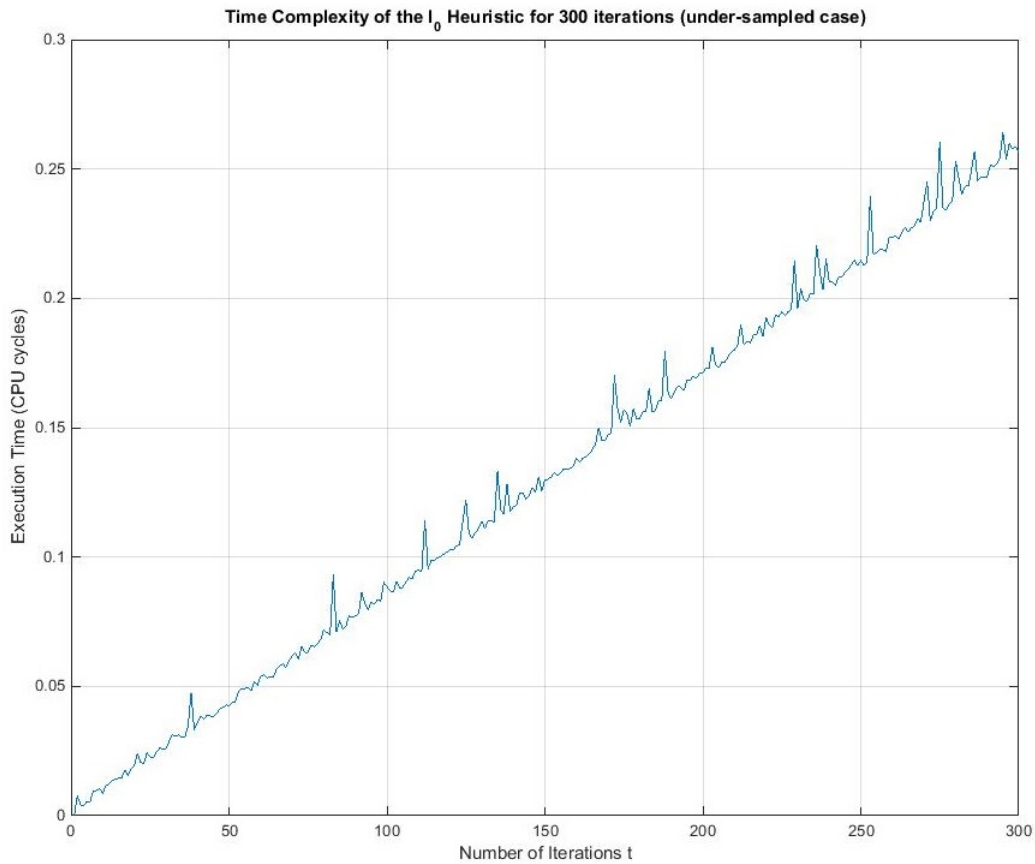

Figure 9.5: Plot of the execution time of the l_0 heuristic in CPU cycles for 300 iterations and $M_1 = 40$ measurements. The l_0 heuristic's parameters are $T = 300$ iterations, $S = 12$ swarms, σ decrease factor $\beta = 0.5$ and initial $\sigma^{(0)} = 2\max(f(t))$ (twice the maximum entry). The small perturbations in the approximately linear time complexity can be attributed to the stochastic step of the heuristic for the generation of the sparse solution (experiment2.m).

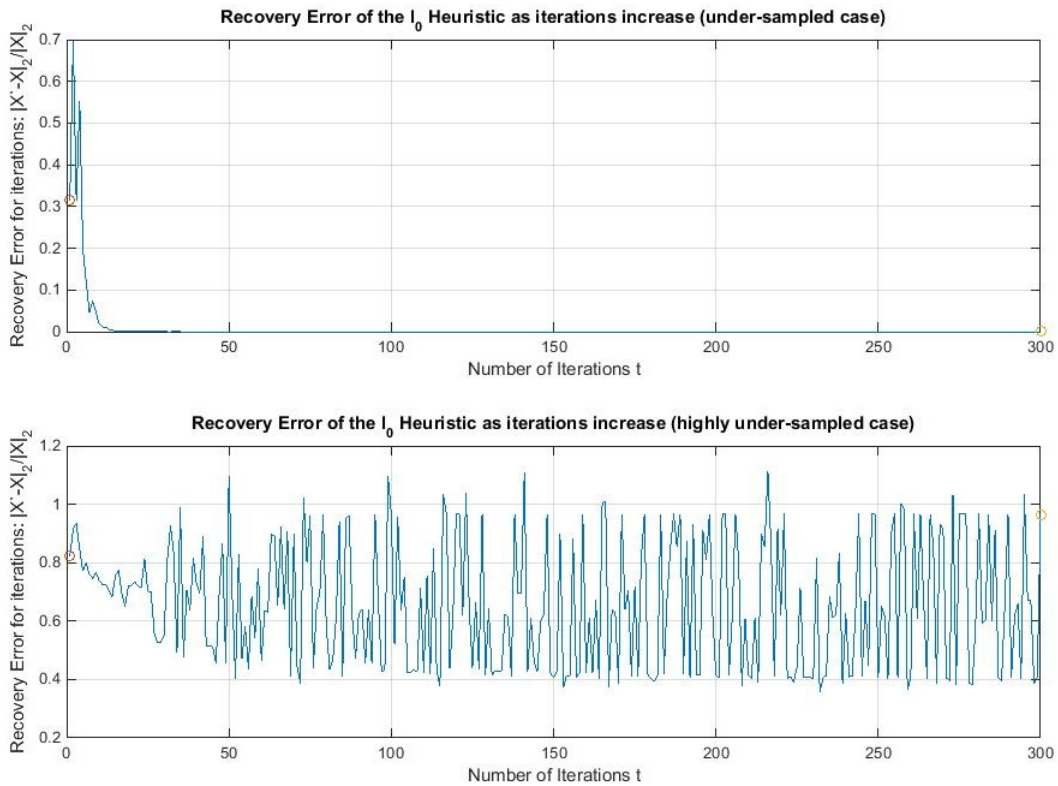

Figure 9.6: Plot of the quality of the solution generated by the l_0 heuristic through iterations. The l_0 heuristic's parameters are $T = 300$ iterations, $S = 12$ swarms, σ decrease factor $\beta = 0.5$ and initial $\sigma^{(0)} = 2\max(f(t))$ (twice the maximum entry). From top to bottom: the first plot represents the recovery error as iterations increase for $M_1 = 40$ measurements, the second plot represents the recovery error as iterations increase for $M_2 = 12$ measurements. In both plots, the recovery error is calculated as $\|\hat{f(t)} - f(t)\|_{l_2}/\|f(t)\|_{l_2}$, where $\hat{f(t)}$ is the estimate of the original (sampled) $f(t)$ (experiment2.m).

2. Pseudo-inverse based on least squares method (Option 2):

$$X_{\text{init}} = (C^T C)^{-1} C^T Y. \tag{9.22}$$

3. Regularised pseudo-inverse with trace (Option 3):

$$X_{\text{init}} = (C^T C + \delta I)^{-1} C^T Y, \quad \delta = \text{trace}(C^T C)/\text{trace}(I). \tag{9.23}$$

4. Regularised pseudo-inverse with l_∞ norm (Option 4):

$$X_{\text{init}} = (C^T C + \delta I)^{-1} C^T Y, \quad \delta = \|C(C^T Y) - Y\|_{l_\infty}. \tag{9.24}$$

5. Regularised pseudo-inverse with Toeplitz matrix (Option 5):

$$X_{\text{init}} = (C^T C + \text{toeplitz}(CC^T))^{-1} C^T Y. \tag{9.25}$$

6. Regularised pseudo-inverse with Vandermonde matrix (Option 6):

$$X_{\text{init}} = (C^T C + \text{vander}(C^T (CC^T)^{-1}))^{-1} C^T Y. \tag{9.26}$$

7. Regularised pseudo-inverse of l_0 heuristic defined in (6.41) (Option 7):

$$X_{\text{init}} = \min\{(C^T C + \delta I)^{-1} C^T Y\}_i \quad \delta = 2/(\lambda_{\max} + r), 0 \leq r \leq 1, i \in \{1, 10\}. \tag{9.27}$$

In the above solutions C is a randomly generated vector with unit norm columns, using Normal distribution, while toeplitz(CC^T) and vander$(C^T(CC^T)^{-1})$ represent the Toeplitz matrix with entries CC^T, defined in Equation (6.30) of Section 6.6 and the Vandermonde matrix with entries $C^T(CC^T)^{-1})$ defined in Equation (6.31) of Section 6.6 respectively. Since the l_0 heuristic is a swarm based method we generate $i = 10$ slightly different solutions in Equation (9.27), based on the random number r and using the maximum eigenvalue $\lambda_{\max}$ of matrix $C^T C$. Then we select the minimum of these solutions based on the minimum value of the gradient of the function defined in Equation (6.4) of Chapter 6. The absolute MSE has been used as a metric to calculate the difference between the original vector X

and its estimate $\hat{X} = X_{\text{init}}$ as an index of solution quality of the different initial solution approaches:

$$\|\hat{X} - X\|_{l_2}, \tag{9.28}$$

where $\|.\|_{l_2}$ represents the Euclidean distance between two vectors.

We can see from Table (9.1) that the regularised least squares approach is very efficient for generating an initial feasible solution, which will be improved through iterations from the l_0 heuristic. In particular, as it has been discussed in Sections 6.6.1 and 6.6.2 the regularised least squares approach provides the best possible feasible solution for both under-determined and over-determined linear systems, as it is simply the analytical solution of the least squares method based on matrix multiplication. This is important so as to enforce sparsity of the solution together with the feasibility. The regularisation using the l_∞ norm seems a bit more efficient for all samples size in contrast to the traditional case of the regularisation using trace, usually suggested as a straightforward method for regularisation in bibliography (see for example [33, 163, 194]). The Toeplitz and Vandermonde based regularisation techniques are efficient only for small sample sizes ($M < 70$) where the regularisation technique is more necessary as they do not involve regularisation only in the main diagonal.

The regularisation with eigenvalues, adopted by the l_0 heuristic, and the least squares pseudo-inverse have very similar performance in terms of efficiency, which is anticipated. In small sample sizes the eigenvalues together with the small random number r provide the necessary regularisation of matrix C^TC for efficient inversion, while for larger samples sizes, regularisation is no longer needed as matrix C^TC can be inverted. Regularisation using eigenvalues follows the dimensionality of the sampling matrix C which is the reason that has been chosen as an initial solution of the l_0 heuristic. It combines the characteristics of the classical least squares approximation so as to generate initial solutions within the feasible region together with the necessary (swarm-based random) regularisation of almost singular sampling matrix for small sample sizes. The value NaN in the Regularised pseudo-inverse with Vandermonde matrix indicates that a solution cannot be found since the matrix $C(C^TC)^{-1}$ is close to singular for over-determined linear systems where $M \geq N$ (i.e. badly scaled as Matlab indicates). Note also that Back-Projection derives the worst results since we try to project back from M to N elements using the highly incomplete sampling vector Y,

which does not provide any knowledge about the structure of the original (sampled) vector X. Indeed this approach is used as an initial feasible solution from a number of sparse recovery methods, including L_1 Magic, NESTA and OMP methods.

Table 9.1: Comparison Table of different initial solutions under different sample sizes for the l_0 heuristic. The recovery error is measured as an average of 100 test runs for each sample size using a real-valued random vector X with length $N = 200$ and sparsity $K = 20$. The absolute MSE $\|\hat{X} - X\|_{l_2}$ between the known vector X and its estimate $\hat{X}$ is used as a measure for the recovery error. The value NaN means that the recovery error cannot be calculated because matrix $C(C^TC)^{-1}$ is close to singular (experiment31.m).

Sample size	1	2	3	4	5	6	7
30	107.4544	2.30826	2.43228	2.53679	2.73318	2.30990	2.30828
40	106.3634	2.21693	2.39073	2.51173	7.86955	2.22110	2.21698
70	105.6323	1.98465	2.30837	2.48246	14.08283	2.20841	1.98522
100	105.1959	1.75891	2.25791	2.48030	11.84971	4.37925	1.76243
150	107.3924	1.26382	2.18342	2.50480	15.24465	11.7058	1.31469
200	105.9464	0.00000	2.07865	2.48386	34.51302	NaN	0.52229
250	104.7186	0.00000	1.98165	2.46167	18.97198	NaN	0.34951

The second experiment involves the research of two alternative weights used for the generation of new solutions for all swarms. As a reminder new solutions are generated based on Equation (6.44) as follows:

$$X_i^{(t)} = \{X_i^{(t-1)} - \mu R_i^{(t)} \nabla_{X_i^{(t-1)}} f(\|X_i^{(t-1)}\|, \sigma^{(t-1)}) + \exp(-L^{(t)}) D^{(t)}\}_{P_K}, \tag{9.29}$$

where $\mu = 2$, $D^{(t)} = C^TY - (C^TC)X_*^{(t)}$, P_K is an operator which sets all but K largest entries equal to zero and $L^{(t)}$ represents the attractiveness of the current best solution as a weight for the relaxed projection $D^{(t)}$. The l_0 heuristic uses the following weight:

$$L^{(t)} = \mu \|X_i^{(t-1)} - X_*^{(t)}\|_{l_2}, \tag{9.30}$$

where $X_*^{(t)}$ is the current (optimal) solution at iteration (t), while $X_i^{(t-1)}$ is the solution carried by the i-th swarm the (previous) iteration $(t-1)$. A slightly different weight for all swarms can be calculated as:

$$L^{(t)} = \mu \|X_i^{(t-1)} - X_j^{(t-1)}\|_{l_2}, \tag{9.31}$$

where $\mu = 2$, $X_i^{(t-1)}$ and $X_j^{(t-1)}$ represent the solutions from the previous iteration carried by swarms i and j. Note that in this case we do not keep the current best solution $X_*^{(t)}$. Instead, we calculate the distance between any two swarms i and j as an attractiveness so that a swarm is attracted by any other swarm which carries a better solution. This is a common approach introduced in firefly algorithm, an efficient variation of swarm intelligence [5, 214]. Finally, a very similar to the l_0 heuristic approach is to use a similar weight to the one introduced in the re-weighted versions of the l_1 and l_2 optimisation principles [45, 56]:

$$L^{(t)} = (\mu(\delta + \|X_i^{(t-1)}\|_{l_1}))^{-1}, \tag{9.32}$$

$\mu = 2$, $0 < \delta \leq 1$ is a small parameter to avoid instabilities of the fraction when the previous solution $X_i^{(t-1)}$ is close to zero. Table (9.2) summarises the experimental results which clearly favour the approach introduced in Equation (9.31) as more efficient in terms of quality of the estimate, requiring though twice execution time. Note that approach suggested in Equation (9.30), which the l_0 heuristic follows, provides a good balance between quality of estimate and execution of time. This observation can be made more clear in cases where the vector to be recovered is consisted of thousands of elements, such as in images, where a typical 256×256 image can be represented as a 65536-element vector. In terms of time complexity of the solution generation, both the approach in (9.30) and (9.32) have linear complexity $O(S)$ while the approach in (9.31) has quadratic complexity $O(S^2)$, where S is the number of swarms used in the calculation of new solutions in the l_0 heuristic.

The third Experiment involves different σ decrease factors β, which directly affect the σ sequence of values but not the final outcome (experiment33.m). The final optimal solution is not affected by the decrease factor since we manage to derive the same optimal solution (estimate) of a 300 element real-valued vector with 10% sparsity (30 non-zero entries) under different sample sizes ($M \in [40 - 350]$) and number of iterations ($T \in [25 - 250]$). This is expected as we use a swarm based method where in every iteration we select the current optimal solution by ranking the swarms based on the value of the gradient of the function. Different decrease sequences can affect the heuristic only if we use one solution generated at each iteration, where $\beta \approx 0.5$ gives better performance with average computation, which has also been discussed in [142, 161, 199] (For further discussion see Sections 6.2 and 6.6.5). Also it seems that σ value is not affected by sparsity levels but only by noise, which has also

Table 9.2: Comparison Table of different approaches for generating new solutions under different sample sizes and iteration numbers for the l_0 heuristic. The l_0 heuristic's parameters are $S = 12$ swarms, σ decrease factor $\beta = 0.5$ and initial $\sigma^{(0)} = 2\max(X)$ (twice the maximum entry), using a real-valued 300 element vector with 23 non-zero randomly generated elements. The relative MSE $\|\hat{X} - X\|_{l_2}/\|X\|_{l_2}$ between the known vector X and its estimate $\hat{X}$ is used as a recovery error metric, while CPU cycles have been used as a rough estimate of execution time (experiment32.m).

	samples size	iterations	Eq.(9.30)	Eq.(9.31)	Eq.(9.32)
recovery error	100	200	$1.3058 * 10^{-6}$	$4.3141 * 10^{-8}$	$2.4681 * 10^{-5}$
	100	300	$1.9035 * 10^{-11}$	$2.7787 * 10^{-13}$	$8.7249 * 10^{-10}$
	180	200	$2.2336 * 10^{-12}$	$3.9406 * 10^{-14}$	$3.5214 * 10^{-11}$
	180	250	$2.1357 * 10^{-12}$	$1.9994 * 10^{-14}$	$1.4674 * 10^{-11}$
execution time	100	200	0.4382	1.1005	0.4059
	100	300	0.6313	1.5329	0.6342
	180	200	0.4609	1.1359	0.4547
	180	250	0.6758	1.5106	0.6675

been discussed in [141]. On the other hand, if we set small σ target value this will affect the performance of the heuristic as we need to have very small recovery error. The target σ value highly depends on the elements of the vector sampled X and also by the initial $\sigma^{(0)}$ value which is calculated either using the maximum element of the vector ($\sigma^{(0)} = 2\|X\|_{l_\infty}$) or from the samples ($\sigma^{(0)} = \|C^T Y\|_\infty$). Larger target value for σ might affect convergence and thus the performance of the heuristic in terms of recovery error as the values of the elements of the vectors used normally range between zero and one and thus the so small value for $\sigma_{\min}$.

The final (fourth) experiment examines how the solution quality changes through the iterations for the l_0 heuristic. Figure (9.7) presents the change of the estimate $\hat{X}^{(t)}$ through iterations (t) for a real-valued 300 element vector using 100 samples and assuming 10% sparsity. In this case we use two different metrics for this purpose, namely $\|X - C^T(C\hat{X}^{(t)} - Y)\|_{l_2}$ for the first top graph and $(X - \hat{X}^{(t)})^T(X - \hat{X}^{(t)})$ (element-by-element subtraction) for the second bottom graph. The sequence of σ is chosen as a decreasing geometrical sequence $\sigma^{(t)} = \beta\sigma^{(t-1)}$, with scaling factor $\beta = 0.5$, $T = 300$ iterations and $S = 12$ swarms. Notice the significant improvement of the estimate in the first 50 iterations and then the gradually smoothness in the graph as the l_0 heuristic keeps improving the estimate till it reaches 250 iterations where further improvement of the solution is not possible. In general, the

proper choice of the number of iterations for the l_0 heuristic highly depends on the available measurements and the sparsity level of the vector to be recovered. Small sparsity levels with small number of measurements require more iterations than small sparsity levels with great number of measurements. If however, the number of measurements is roughly the sparsity level then a high number of iterations will not significantly improve the derived solution for the l_0 heuristic.

9.4.4 Effects on sparsity in recovery

In this experiment we conduct simulations using different sparsity levels, in both real and complex-valued vectors, so as to measure the performance of the l_0 heuristic. In particular, we discuss how the sparsity level affects efficient recovery for three different types of signals; a completely random unknown initial signal (without noise), a random signal with real values (without noise) and a approximately sparse complex-valued signal (with noise). This task is achieved experimentally by choosing different levels of sparsity and then measuring the recovery error using relative MSE defined in Equation (9.10) for the noiseless case and SNR defined in Equation (9.11) for the noisy case.

It is expected that sparsity level $K \leq M/2$ (half the number of measurements) is the limit to ensure uniqueness of the sparsest solution based on the theoretical results discussed in [42, 76], though most algorithms cannot achieve this limit [116, 141]. The theoretical lower limit for sparse decomposition based on the l_0 norm based problem is $M = K + 1$ measurements, while the theoretical upper bound on sparsity level is $O(\sqrt{M})$ or $K \leq M/2$ (see Chapter 4 for details). However, the lower measurement bound in practice cannot be achieved by any algorithm due to the NP-Hardness of the problem [39, 41]. This condition on the number of measurements is very important as otherwise we might experience aliasing problems, even when $CX \approx C\hat{X}$ though X and $\hat{X}$ are completely different. In such cases of aliasing, the use of an ordinary sparse recovery technique is not enough in terms of efferent recovery as the use of some extra preprocessing steps or filters is required so as to enforce sparsity or to remove the possibility of strong aliasing. However, aliasing techniques for recovery problems are beyond the scope of this thesis. We will only briefly discuss some filters and how these could be used together with a sparse recovery method, such as the l_0 heuristic, in Section 9.4.10 as an efficient way to remove some certain levels of noise, acquired in the sampling process. For a more detailed treatment on filters and noise the interested

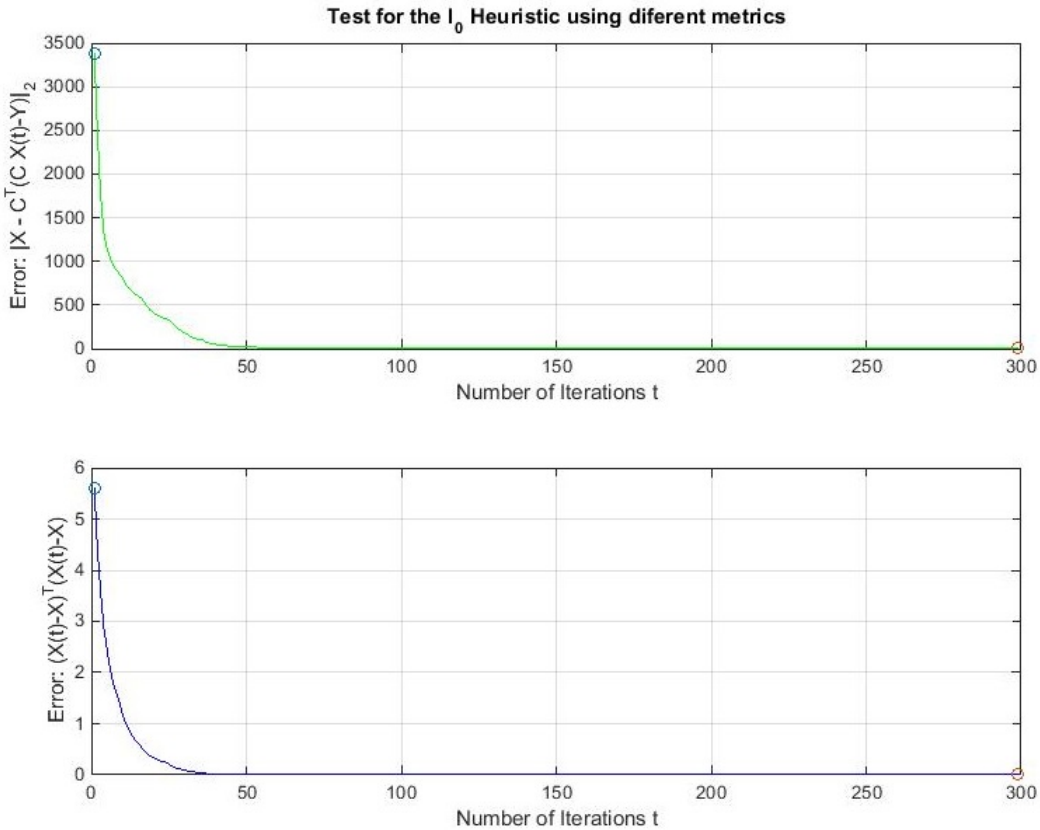

Figure 9.7: Plot of the quality of the solution generated by the l_0 heuristic through iterations in order to recover a real-valued 300 element vector from 100 samples, assuming 10% sparsity. The l_0 heuristic's parameters are $T = 300$ iterations, $S = 12$ swarms, σ decrease factor $\beta = 0.5$ and initial $\sigma^{(0)} = 2\max(X)$ (twice the maximum entry). From top to bottom: the first plot represents the recovery error as iterations increase using $\|X - C^T(C\hat{X}^{(t)} - Y)\|_{l_2}$ metric, the second plot represents the recovery error as iterations increase using $(X - \hat{X}^{(t)})^T(X - \hat{X}^{(t)})$ metric (element-by-element subtraction), where $\hat{X}^{(t)}$ is the estimate of the original (sampled) X vector at iteration (t) (experiment34.m).

reader is advised to see [32, 136, 157, 159].

Figures (9.8) and (9.9) represent the recovery error of a completely random unknown initial vector from its samples assuming different sparsity levels ($K = 1, 2, \ldots, 100$). Both the samples vector Y and the Sensing matrix C have been generated using the Normal distribution with values between zero and one. The l_0 heuristic's parameters used in the experiment are $T = 300$ iterations, $S = 12$ swarms, σ decrease factor $\beta = 0.5$ and initial $\sigma^{(0)} = 2\max(C^T Y)$ (twice the maximum back-projection of samples). The relative recovery error was calculated using the samples vector as $\|C\hat{X} - Y\|_{l_2}/\|Y\|_{l_2}$ as a measure of the quality of the solution found by the l_0 heuristic. It can be seen that in both cases of under-sampled measurements (50 and 200 samples) efficient recovery is possible with error 10^{-1} assuming small sparsity level, namely $K \leq 10$. As the sparsity level increases with the same samples size, the recovery error deteriorates. In fact, the recovery error in both sample cases ($M = 50$ and $M = 200$) increases exponentially as sparsity level decreases, which is expected as more samples are required for efficient recovery. Another important aspect of this experiment in terms of poor results of the l_0 heuristic is the fact that the samples vector Y and the Sensing matrix C are purely random, in essence that Y is not derived though sampling ($CX = Y$). In general, efficient recovery using the l_0 heuristic is possible and efficient for small sparsity levels and sample sizes, given that the sparsity of the original sampled vector is known and the samples Y have been derived from the original sampled vector (i.e. the samples are not random).

Figure (9.10) represents the recovery error of a 100 element real-valued random vector from 30 samples, without noise, using different sparsity levels, namely $K = 1, 2, \ldots, 80$. For every sparsity level a new random vector is generated using the Normal distribution and then sampled, keeping only 30 samples (un-normalised entries in C). The l_0 heuristic's parameters using in the experiment are $T = 300$ iterations, $S = 12$ swarms, σ decrease factor $\beta = 0.5$ and initial $\sigma^{(0)} = 2\max(X)$ (twice the maximum entry of original vector). The relative recovery error was calculated using the relative MSE, $\|\hat{X} - X\|_{l_2}/\|X\|_{l_2}$, in order to measure the quality of the estimate $\hat{X}$ found using the l_0 heuristic. It can be seen that for small sparsity levels, $K \leq 10$, efficient recovery is possible (10^{-1} error) given the 30 measurements. This observation follows the theoretical principle we already discussed about sparsity levels in terms of measurements. Note that due to approximation of the l_0 norm based problem, efficient recovery is achieved in high under-sampled cases if sparsity level is a bit less than

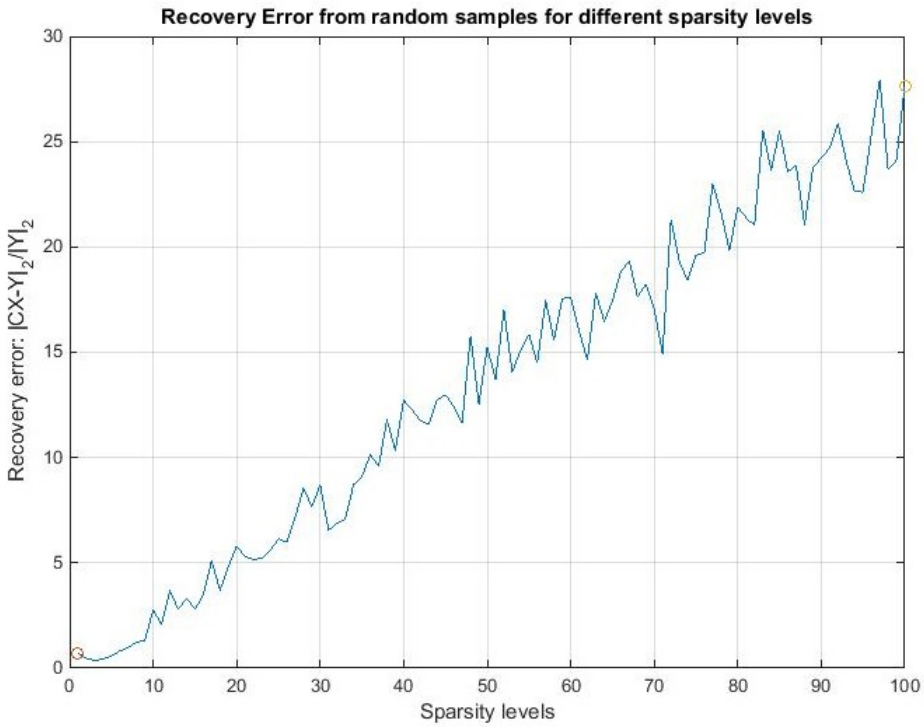

Figure 9.8: Sparse recovery of an initially unknown 100 element real-valued vector, from 50 samples generated using the Normal distribution, assuming different levels of sparsity $K = 1, 2, \ldots, 100$. The l_0 heuristic's parameters are $T = 300$ iterations, $S = 12$ swarms, σ decrease factor $\beta = 0.5$ and initial $\sigma^{(0)} = 2\max(C^T Y)$ (twice the maximum back-projection of samples). The relative recovery error was calculated using the samples vector as $\|C\hat{X} - Y\|_{l_2}/\|Y\|_{l_2}$ (experiment4.m).

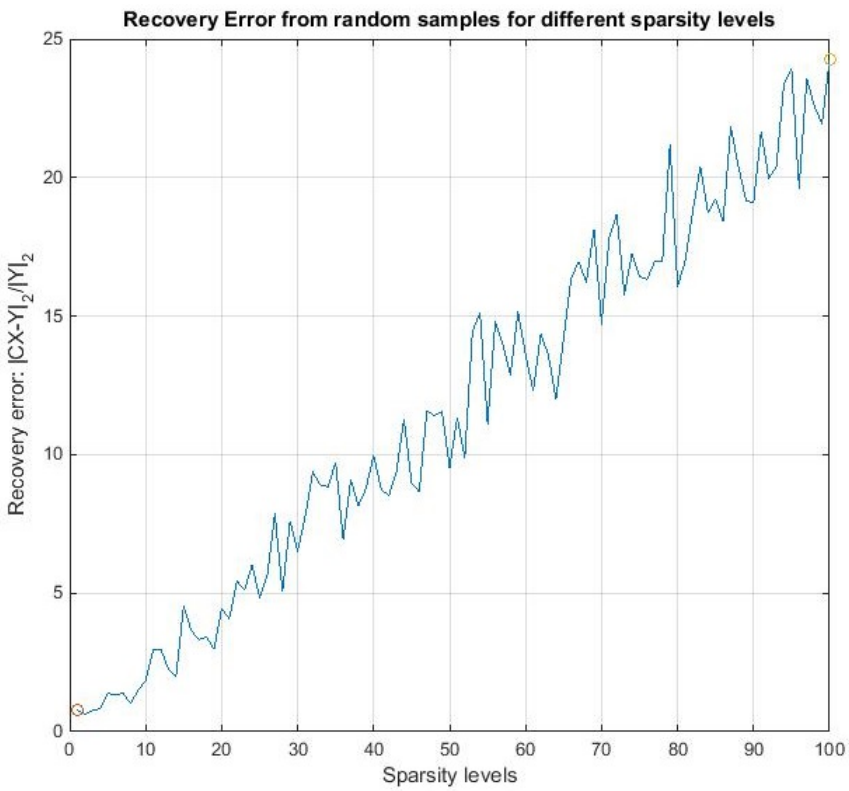

Figure 9.9: Sparse recovery of an initially unknown 100 element real-valued vector, from 200 samples generated using the Normal distribution, assuming different levels of sparsity $K = 1, 2, \ldots, 100$. The l_0 heuristic's parameters are $T = 300$ iterations, $S = 12$ swarms, σ decrease factor $\beta = 0.5$ and initial $\sigma^{(0)} = 2\max(C^T Y)$ (twice the maximum back-projection of samples). The relative recovery error was calculated using the samples vector as $\|C\hat{X} - Y\|_{l_2}/\|Y\|_{l_2}$ (experiment4.m).

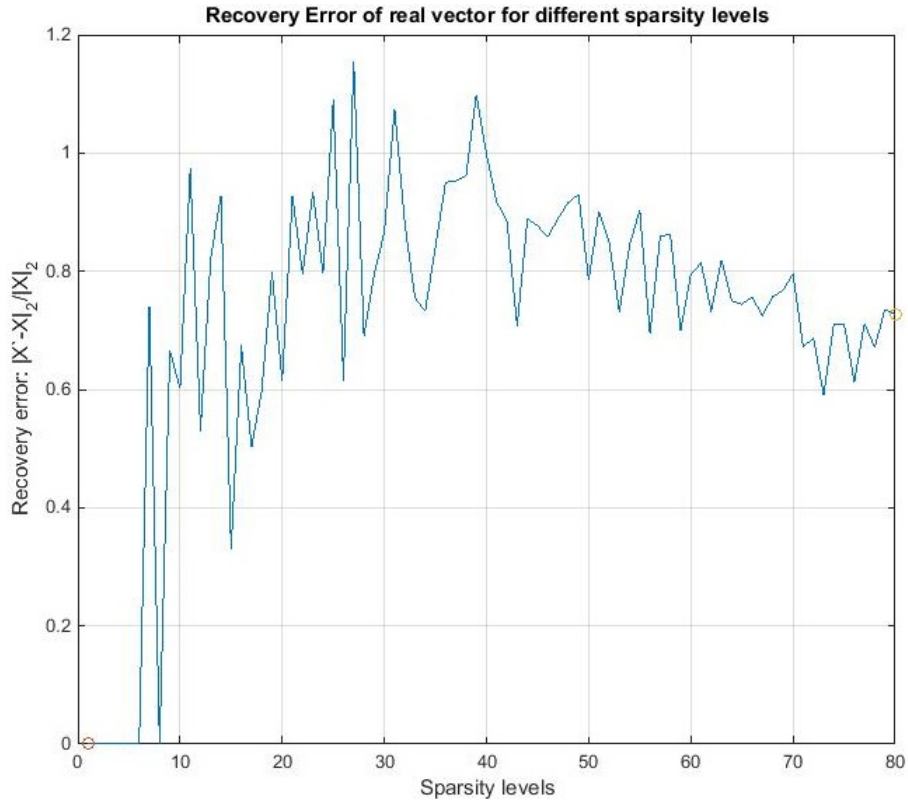

Figure 9.10: Sparse recovery of a 100 element real-valued random vector, from 30 samples, assuming different sparsity levels $K = 1, 2, \ldots, 80$. The initial random vector and the (un-normalised) sampling matrix C were generated using the Normal distribution with values between zero and one. The l_0 heuristic's parameters are $T = 300$ iterations, $S = 12$ swarms, σ decrease factor $\beta = 0.5$ and initial $\sigma^{(0)} = 2\max(X)$ (twice the maximum entry). The relative recovery error was calculated using MSE, $\|\hat{X} - X\|_{l_2}/\|X\|_{l_2}$, as a measure of the quality of the estimate $\hat{X}$ (experiment4.m).

half of the number of measurements, according to the theoretical framework. Moreover, approximately sparse vectors with $K \geq 50$ (greater than half of their length) require more samples, for efficient recovery, as a number of non-zero (significant) entries are lost in the sampling process of a vector (due to the null space of C).

The final Experiment (Figure (9.11)) represents sparse recovery using the l_0 heuristic for a approximately sparse complex-valued vector in a noisy environment. The noisy linear measurements Y are drawn from the model $Cf(t)+\epsilon = Y$, with C a 40 by 50 sampling matrix with randomly generated elements between zero and one using the Normal distribution,

without normalised columns. The random parameter $0.01 \leq \epsilon \leq 0.1$ represents the small additive noise in the measurement model, generated using Normal distribution. Vector $f(t)$ represents a approximately sparse vector of complex numbers whose entries follow the exponential decay condition:

$$f(t) = \exp(-3ut), \tag{9.33}$$

where $u = 0.1 + 0.5i$ is a small complex number and t a parameter affecting sparsity. In general, the entries of $f(t)$ satisfy the decreasing sorted order of magnitude, i.e. $|f_1(t)| \geq |f_2(t)| \geq \ldots \geq |f_N(t)|$ where the K-th largest entry, out of N, obeys the rule:

$$|f_k(t)| \leq Rk^{-1/s}, \tag{9.34}$$

for a small positive constant R and $s \leq 1$. This is very common in unitary transforms where a suitable dictionary (e.g. wavelets) changes the order of magnitude of the vector's entries [41, 42]. Previous work in this area focused on cases of $0 < s < 1$ for every vector that is compressible with fixed s. Let $\hat{f}(t)$ be the best sparse approximation we could obtain if the locations and amplitudes of the K-largest entries of $f(t)$ were known. Then the error due to the worst case over all s-compressible signals in the class we have [43, 75, 145]:

$$\|\hat{f}(t) - f(t)\|_{l_2} = O(K^{1-2/s}), \tag{9.35}$$

for the K largest entries with respect to appropriate scaling parameter s as earlier. This experiment follows the same pattern. To efficiently measure the efficiency of the l_0 heuristic against the sparsity level we generate a vector by activating $1 \leq K \leq N$ largest elements out of N (length of vector).

Figure (9.11) plots the recovery error of this complex-valued vector against sparsity levels, using SNR metric as a measure of performance versus different sparsity levels. In all the simulations the l_0 heuristic's parameters are $T = 300$ iterations, $S = 12$ swarms, σ decrease factor $\beta = 0.5$ and initial $\sigma^{(0)} = 2\|f(t)\|_{l_\infty}$ (twice the maximum entry of vector $f(t)$). We also assume that vector $f(t)$ is strictly complex, which means that its non-zero elements have both real and imaginary parts, so as to decompose it into real and imaginary parts in both the sampling and recovery process. Observe that efficient recovery with de-noising (removing

small error) is possible for small sparsity levels between 5 and 30, where the metric values are very small close to zero. For sparsity levels below 5 we can see, as expected, that most of the magnitudes ($f(t)$ entries) are below or equal to the noise level and thus the high recovery error. In such cases we need more samples to efficiently estimate the vector's values. Small fluctuations in the plot can be attributed partially to the stochastic nature of the l_0 heuristic and partially to the fact that small noise levels $\epsilon \approx 0.01 - 0.03$ have not been removed from the recovery process and thus been kept in the estimate $\hat{f}(t)$ after recovery. Efficient recovery is again achieved if sparsity level is less than half of the number of measurements. Notice also that as sparsity level increases above 35 the SNR decreases below zero indicating poor results in terms of efficient recovery. This is expected as the vector is now weakly sparse and thus more samples are needed for efficient recovery under the presence of noise.

9.4.5 Test run for a real-valued noiseless vector

In this experiment we measure the performance of different sparse recovery methods in terms of execution time (measured in CPU cycles) and recovery error (measured as a relative MSE defined in (9.10)). For this purpose a 70-element real-valued vector, of 15 non-zero entries with random values between zero and one drawn from the Normal distribution, was created and sampled using different sample sizes, namely $M = 1, 2, \ldots, 80$. The samples collected are used as the same input data for all methods, in order to recover the original real-valued sparse vector. This broader sample size was chosen so as to measure the efficiency of sparse recovery methods from highly under-sampled cases $M \leq 20$, where loss in recovery is expected, to highly over-sampled cases $M \geq 60$, where measurement bounds and necessary conditions for the stability of the recovered solution are only conjectured to work, without any formal proofs in the literature [189].

Figure (9.12) presents the execution time, expressed in CPU cycles, and recovery error, expressed as relative MSE, for different sparse recovery methods under different sample sizes. In general, no preprocessing steps have been applied to any of the CS methods presented in Figure (9.12) to enhance the performance of the calculations or the sparsity of the initial vector. Figure (9.13) presents the average execution time of each sparse recovery method for all the samples for better representation of the time complexity of each method. Based on the nature of the random test vector to be recovered, 20 iterations have been chosen for the OMP method, 20 iterations as a threshold for the Lasso method, 60 iterations for the AIHT,

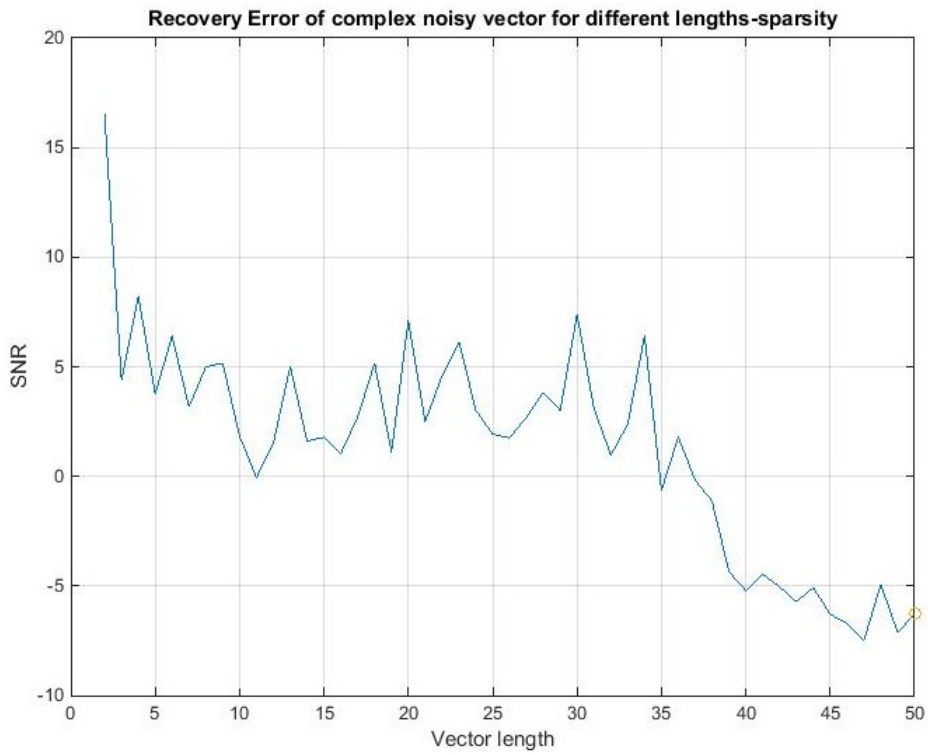

Figure 9.11: Sparse recovery of a 50 element complex-valued vector $f(t)$, from 40 noisy measurements, assuming different levels of sparsity $K = 1, 2, \ldots, 50$. In this setting samples Y are obtained from noisy linear measurements as $Cf(t) + \epsilon = Y$, with C a 40 by 50 sampling matrix with randomly generated elements (between zero and one using the Normal distribution), $0.01 \leq \epsilon \leq 0.1$ is a small randomly generated number simulating noise, using Normal distribution and $f(t) = \exp(-3ut)$ is the signal, with parameters $u = 0.1 + 0.5i$ and t, which affects sparsity. The l_0 heuristic's parameters are $T = 150$ iterations, $S = 12$ swarms, σ decrease factor $\beta = 0.5$ and initial $\sigma^{(0)} = 2\|f(t)\|_{l_\infty}$ (experiment4.m).

$\mu = \epsilon = 10^{-4}$, $\delta = 0$ and initial solution $C^T Y$ for the NESTA package. The l_0 heuristic parameters are $T = 25$ iterations, $S = 10$ swarms, σ decrease factor $\beta = 0.5$ and initial $\sigma^{(0)} = 2\|X\|_{l_\infty}$ (twice the maximum element of the initial random vector). Note that CVX package was used for the implementation of the LASSO and l_1 norm based optimisation principles, using the SEDUMI solver. CVX is a Matlab based package which incorporates different types of solvers, including SEDUMI, which is based on the Interior Point (Quasi Newtonian) methods [97, 130]. In general, the experimental results shown in Figures (9.12) and (9.13) suggest the efficiency of the l_0 heuristic, in terms of recovery error and execution time, in highly under-sampled cases ($M = 15 - 25$ measurements) and over-sampled cases ($M \geq 50$ measurements) for a random real-valued vector in a noiseless environment. The l_0 heuristic performs better than other competing approaches in these cases, while for the rest of the measurement size $M = 30 - 60$ it performs the same or slightly worse than other competing approaches.

In particular it can be seen from Figure (9.12) that for sample sizes less than 15 the performance of all methods is very poor in terms of recovery error, as due to sparsity level ($K = 15$) of the original vector more samples are required for efficient recovery. Note that the performance of the l_1 minimisation principle is by far worse in this case than any other sparse recovery method. As the sample size M increases from 15 to 25, SL0, l_0 heuristic, LASSO and l_1 norm based minimisation principle perform better in terms of sparse recovery, with the l_0 heuristic and SL0 to perform slightly better than the other methods with error 10^{-1}. Note that the minimum of the l_0-norm depends on the number of similar components of the original vector X, which means that in such highly under-sampled cases the optimal solutions of the l_0 and l_1 norm based problems do not coincide since RIP/UUP properties do not hold. As the samples size increases to 30 and 40 samples the l_1 norm minimisation achieves better results, with error 10^{-2}, than any other recovery method. This can be attributed to the fact that now due to the sample size RIP/UUP properties hold, and thus the l_1 norm based solution is the same as the l_0 norm based solution. As samples increase even more to 50 and 60 the error recovery for the l_1 minimisation is 10^{-20} in contrast to 10^{-8} for the l_0 heuristic and 10^{-5} for the SL0. Note that the recovery error for LASSO, IRLS and NESTA is 10^{-1}, while the recovery error for OMP and AIHT is greater than 10^{+1}, which is expected as these recovery methods are designed for efficient sparse recovery in highly under-sampled cases. Note that as we approach 70 samples and over-sampled cases of 80 samples the recovery error

of the l_0 heuristic is 10^{-25} in contrast to the recovery error of SL0 and l_1 minimisation where the error is between 10^{-15} and 10^{-18}. The recovery error for AIHT, LASSO and NESTA is 10^{-1}, while OMP and IRLS cannot efficiently solve the over-sampled case yielding an error of 10^{+2}.

It is important to note that the l_0 heuristic converges in only a few steps (small execution time) but the quality of its solutions sometimes deteriorates due to its stochastic nature (cases of 25 and 40 samples). This stochastic property also makes the running time of the l_0 heuristic one of the smallest (Figure (9.13)), while the gradient based methods (LASSO, NESTA, l_1 minimisation) seem to have greater running times as the sample size increases. In particular, NESTA seems to have the largest running time, roughly three times more than any other recovery method, while IRLS, AIHT and SL0 methods seem to have slightly better running times than the l_0 heuristic. Another interesting aspect is the special case of OMP method, which is not based on a gradient based step, but it has the third largest execution time. This fact can be attributed to the greedy approach of the OMP to find the sparsest solution by minimising the misfit between the observations and the estimate through iterations using the under-sampled measurements. On the other hand, the assignment of weights in the l_2 norm in IRLS seems to perform poorly in terms of recovery error but very fast, in fact the fastest recovery method, in terms of execution time. Another important characteristic of sparse recovery is that due to the approximation of the l_0 norm 30 samples appear to be the threshold for efficient recovery based on sparsity levels though the theoretical threshold is 16 (level of sparsity K plus one [39]). Till now none of the sparse recovery methods suggested in bibliography are able to reach this theoretical threshold [39, 85, 196, 215].

9.4.6 Test run for a complex-valued noisy vector

In this experiment we measure the performance of different sparse recovery methods in terms of execution time (measured in CPU cycles) and recovery error (measured as a relative SNR defined in (9.11)). For this purpose a 70-element complex-valued random vector, of 15 non-zero entries with values between zero and one drawn from the Normal distribution was created and sampled using different sample sizes, namely $M = 1, 2, \ldots, 80$. The sampling matrix C has i.i.d. entries following Normal distribution ($C \sim \mathcal{N}(0, 1)$) under small random additive noise levels $[0.1 - 0.01]$. Note that the samples collected are used as the same input data for all methods, in order to recover the original complex-valued sparse vector.

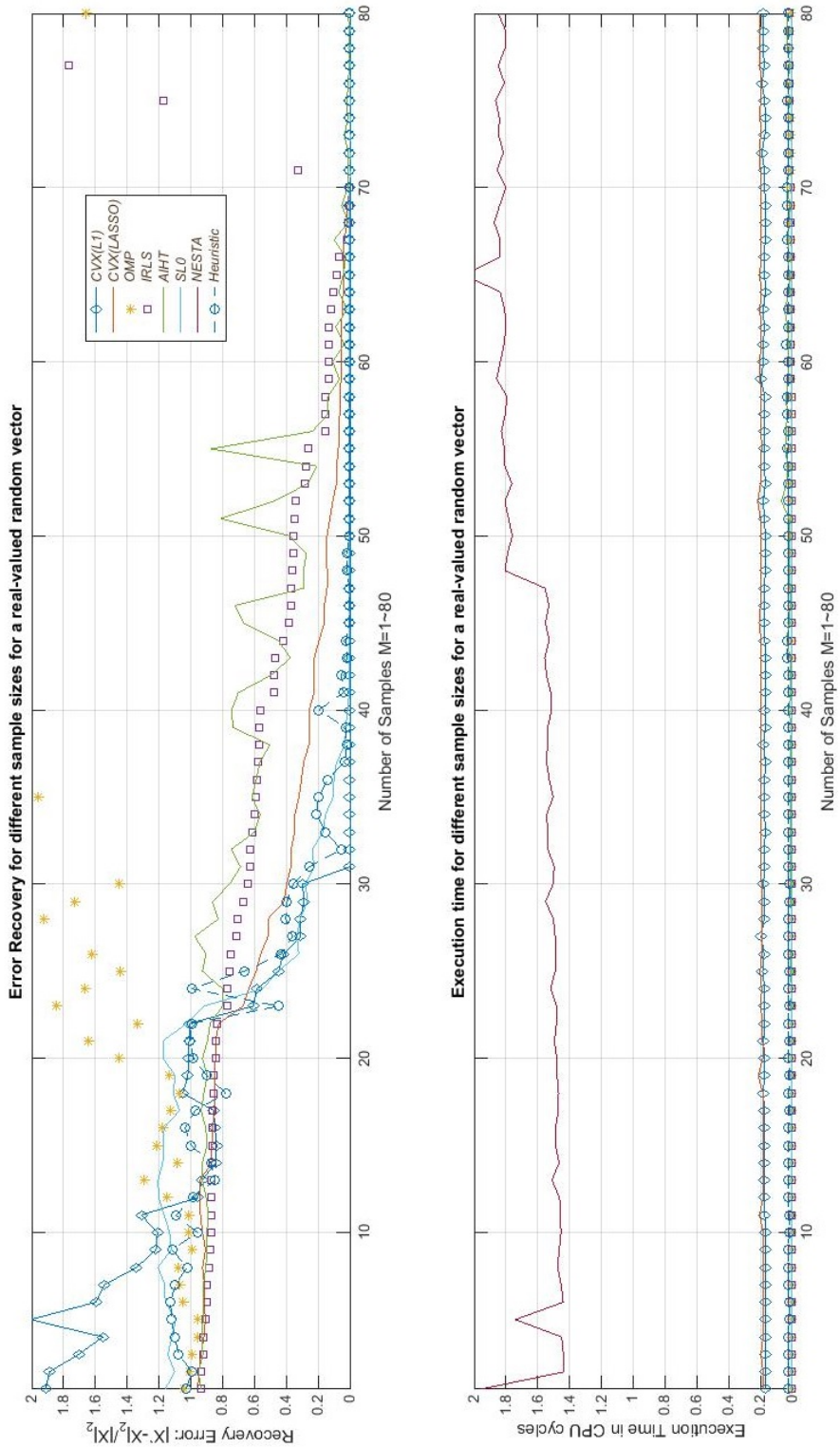

Figure 9.12: Plot of benchmark results showing average running time, in CPU cycles, and recovery error, in relative MSE, of different sparse recovery methods versus sample size. A 70-element real-valued random vector, of 15 non-zero entries with values between zero and one drawn from the Normal distribution, was created and sampled using different sample sizes, namely $M = 1, 2, \ldots, 80$. The parameters of sparse recovery methods are: 20 iterations for OMP, 20 iterations for Lasso, 60 iterations for AIHT, $\mu = \epsilon = 10^{-4}$, $\delta = 0$ and initial solution $C^T Y$ for NESTA, $T = 25$ iterations, $S = 10$ swarms, σ decrease factor $\beta = 0.5$ and initial $\sigma^{(0)} = 2\|X\|_{l_\infty}$ for l_0 heuristic (experiment5.m).

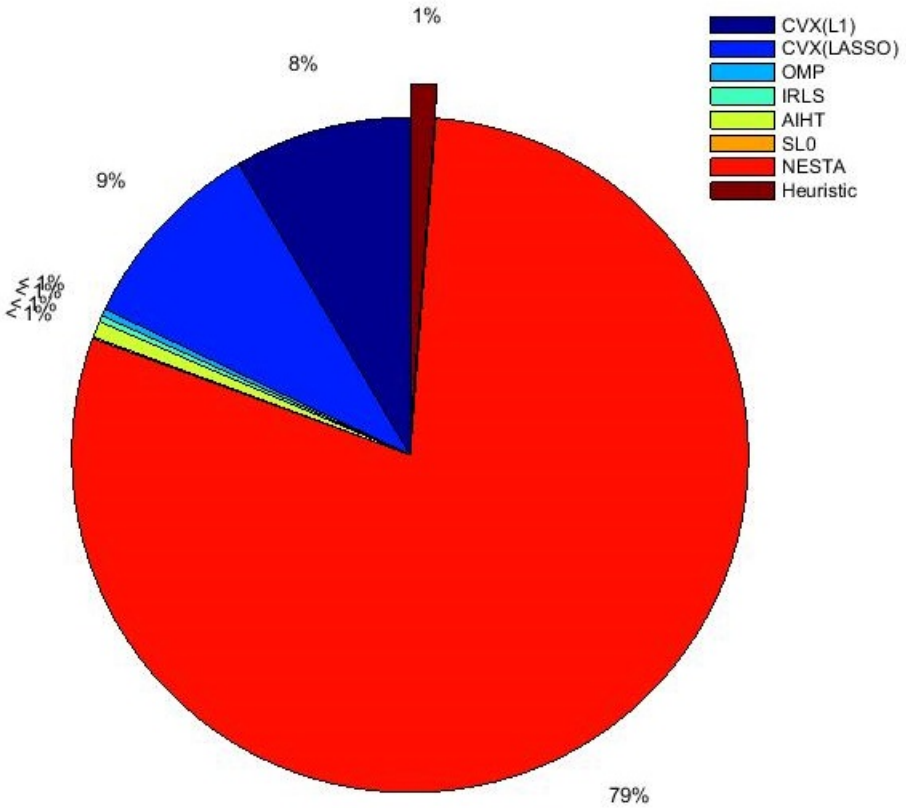

Figure 9.13: Plot of benchmark results showing average execution time of all sparse recovery methods for all the sample sizes $M = 1, 2, \ldots, 80$ (measured in CPU cycles), for a 70-element real-valued random vector, of 15 non-zero entries with values between zero and one. The parameters of sparse recovery methods are: 20 iterations for OMP, 20 iterations for Lasso, 60 iterations for AIHT, $\mu = \epsilon = 10^{-4}$, $\delta = 0$ and initial solution $C^T Y$ for NESTA, $T = 25$ iterations, $S = 10$ swarms, σ decrease factor $\beta = 0.5$ and initial $\sigma^{(0)} = 2\|X\|_{l_\infty}$ for l_0 heuristic (experiment5.m).

This broader noisy sample size was chosen so as to measure the efficiency of sparse recovery methods from highly under-sampled cases $M \leq 20$, where loss in recovery is expected due to small number of samples and noise, to highly over-sampled cases $M \geq 60$, where measurement bounds and necessary conditions for the stability of the recovered solution are only conjectured to work, without any formal proofs in the literature [189]. In particular, we consider noise as a random process; an undesired small random quantity which often occurs in the presence of a desired signal (vector). This approach in terms of noise includes both deterministic and non-deterministic signals. A deterministic is the one that is perfectly predictable from its past values, such as a sinusoid signal, which may be represented by a fixed amplitude and a time-varying phase angle of $2\pi ft$, with frequency f known. In general, in deterministic signals the amplitude and phase are known for past time and for any future time. Nevertheless, random signals such as noise do not have this property.

The aim of this experiment is to investigate the effect of noise on the performance of the l_0 heuristic against other sparse recovery methods discussed in Chapter 7. The parameters of sparse recovery methods are: 18 iterations for Lasso, 80 iterations for COSAMP, $T = 20$ iterations, $S = 10$ swarms, σ decrease factor $\beta = 0.5$ and initial $\sigma^{(0)} = 2\|X\|_{l_\infty}$ (twice the maximum element of the initial random vector) for l_0 heuristic. Note that IRLS, AIHT and NESTA do not support noisy complex random measurements and thus they have been omitted, while COSAMP as an efficient tailored variation of OMP for complex noisy samples is used. For the implementation of the l_1 minimisation principle and LASSO we have used CVX package with SEDUMI solver. The results of the simulation are presented in the Figures (9.14) and (9.15). No preprocessing steps have been applied to any of the CS methods to enhance the performance of the calculations or the sparsity of the initial vector. Figure (9.14) presents the performance of different sparse recovery methods in terms of estimate efficiency using the SNR metric.

Notice that from the logarithm expansion rule, SNR ratio can be re-written as $10\log_{10}(\|X\|_{l_2}) - 10\log_{10}(\|X - \hat{X}\|_{l_2})$, which implies that efficient recovery can be if $SNR \gg 10$, since $10\log_{10}(\|X\|_{l_2}) \gtrsim 10$ for $\|X\|_{l_2} \geq 10$ (based on the values of the initial vector) and thus the smaller the distance $\|X - \hat{X}\|_{l_2}$ the greater the SNR metric. For sample size smaller than 30 the performance of all methods is very poor in terms of recovery error ($SNR \approx 0$) since there are not enough measurements for efficient recovery under the presence of noise. This is in accordance with the theoretical limit for efficient recovery which sets that the necessary

number of measurements for efficient recovery must be at least $M \geq 2K$, with K the sparsity level of the vector and assuming low levels of sparsity ($K < 30\%N$ of the vector length N) [42, 46, 48]. As the number of measurements increase for $M \geq 30$ the SNR also increases which is expected as more samples contribute towards the vector recovery and eliminate the small noise levels. For sample size between 25 and 35 the l_1 minimisation principle performs better recovery and de-noising than any other method with $SNR \approx 30$. The SL0 method comes second in terms of efficiency with $SNR \in [10, 25]$, while the l_0 heuristic and LASSO methods come third with similar results, $SNR = 10$. This efficiency of the l_1 minimisation principle can be attributed to the fact that RIP/UUP properties hold, due to sample size, and thus the l_1 norm based solution is equivalent to the l_0 norm based solution.

As samples increase even more to 35 and 50 the SNR for SL0 is almost 50, while for l_1 minimisation is 40 and for the l_0 heuristic 50. For sample sizes greater than 50 the l_0 heuristic achieves by far the best estimate in terms of $SNR \geq 50$, with SL0 and l_1 minimisation following in terms of efficiency. At this point note the small fluctuation of SNR exactly on the 70-th sample where in this case we have a fully determined linear system of measurements. In this case traditional methods of matrix inversion (e.g. Crammer method, Gauss-Jordan elimination) can be used to invert C matrix and derive a good quality of estimates. In general, the experimental results shown in Figures (9.14) and (9.15) suggest the efficiency of the l_0 heuristic, in terms of recovery error and execution time in over-sampled cases ($30 \leq M \geq 50$ measurements) where better SNR is derived for a random noisy complex-valued vector. For under-sampled cases ($M \leq 50$ measurements) other sparse recovery methods perform better or equally better with the l_0 heuristic in terms of SNR.

As noted earlier the heuristic has both a stochastic and a deterministic step which make it very efficient and fast in terms of execution times. As an example of an efficient yet random recovery observe the SNR of Figure (9.14) at the 42-nd sample and them compare it with the SNR of the l_0 heuristic at the 50-th and 52-nd sample. We can also see from Figure (9.15) that the l_0 heuristic's execution time is smaller than l_1 minimisation principle and LASSO but a bit greater than those of SL0 and COSAMP. In fact, SL0 method seems to have the smallest running time of all the other methods. This can be attributed partially to the gradient step used in the calculations for the l_1 minimisation principle and LASSO and the separately sample and recover of the real and imaginary parts of the initial complex valued vector, resulting in twice the execution time the l_0 heuristic usually needs.

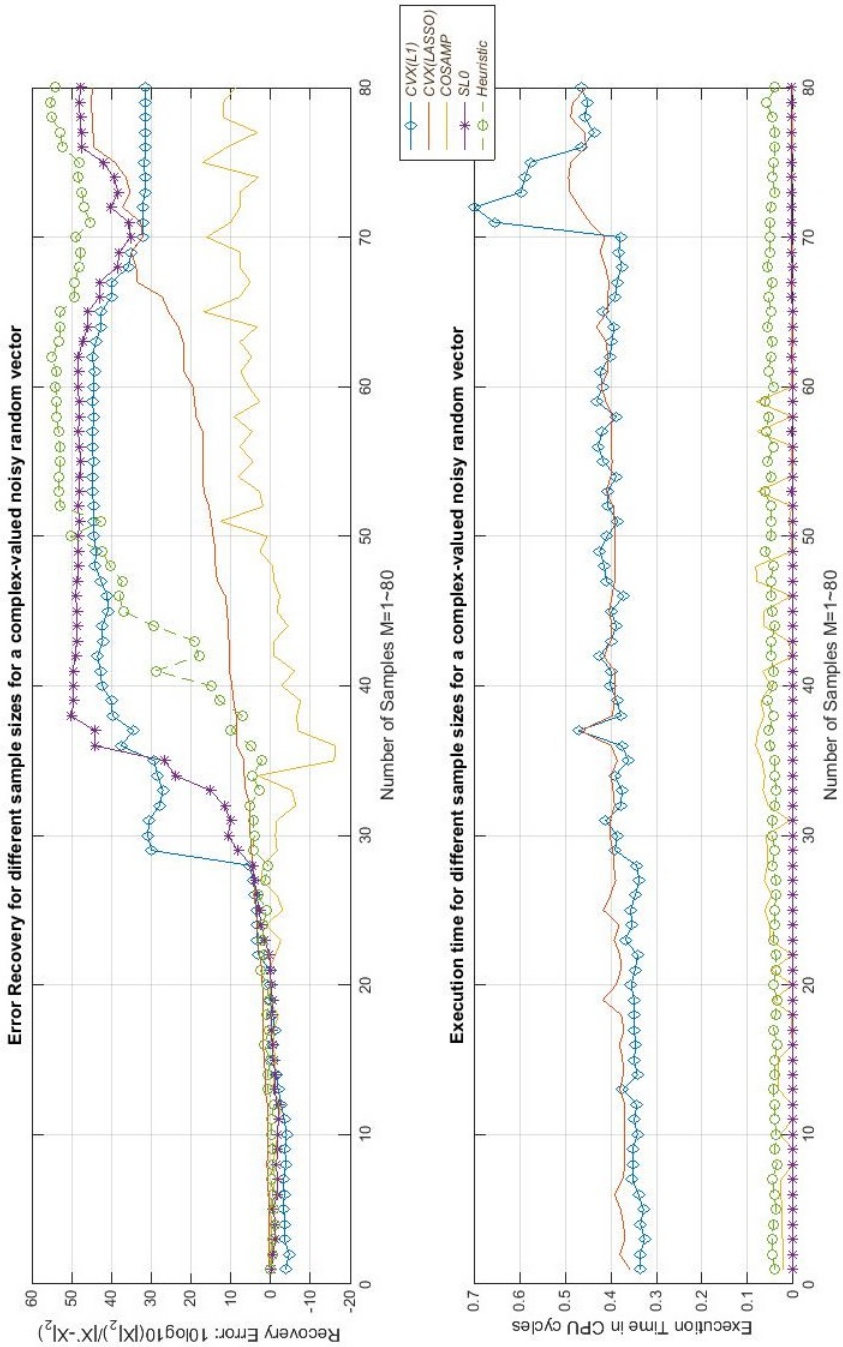

Figure 9.14: Plot of benchmark results showing average running time, in CPU cycles, and recovery error, in relative SNR, of different sparse recovery methods versus noisy sample sizes (with noise levels $[0.1 - 0.01]$). A 70-element complex-valued random vector, of 15 non-zero entries with values between zero and one drawn from the Normal distribution, was created and sampled using different sample sizes, namely $M = 1, 2, \ldots, 80$. The parameters of sparse recovery methods are: 18 iterations for Lasso, 80 iterations for COSAMP, $T = 20$ iterations, $S = 10$ swarms, σ decrease factor $\beta = 0.5$ and initial $\sigma^{(0)} = 2\|X\|_{l_\infty}$ for l_0 heuristic (experiment6.m).

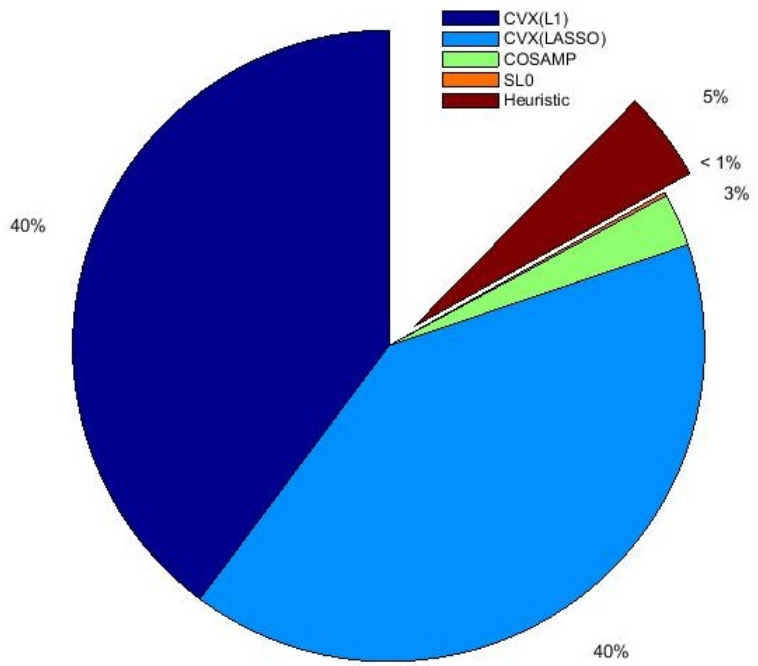

Figure 9.15: Plot of benchmark results showing average execution time of all sparse recovery methods for all the sample sizes $M = 1, 2, \ldots, 80$ (measured in CPU cycles), for a 70-element complex-valued random vector, of 15 non-zero entries with values between zero and one. The parameters of sparse recovery methods are: 18 iterations for Lasso, 80 iterations for COSAMP, $T = 20$ iterations, $S = 10$ swarms, σ decrease factor $\beta = 0.5$ and initial $\sigma^{(0)} = 2\|X\|_{l_\infty}$ for l_0 heuristic (experiment6.m).

9.4.7 The size of the Sensing matrix

In this experiment we investigate the effect of the size (dimension) of the Sensing matrix on the performance and justify the scalability of the l_0 heuristic. In particular, we analyse the effect of increasing the dimensions of the Sensing matrix so as to achieve an over-complete system of linear equations as samples. It is expected that as the dimension of the Sensing matrix increases and exceeds the dimensions of the original (sampled) vector, the sparsest solution may no longer be unique since RIP/UUP properties are only conjectured to work [189]. The theoretical results for the under-determined case state that the sparsest solution is unique if $M \approx K \log_2 N$, where M is the number of measurements and K is the sparsity level of the original vector of N elements, with $K \ll M \ll N$. In practice, from the previous experimental results (Experiments 5 and 6) we have seen that efficient sparse recovery is achieved when $M > 2K$, though the theoretical results state that efficient sparse recovery using the l_0 norm based problem is possible for $M = K + 1$ [41–43]. However, this is not possible in practice due to the complexity of the l_0 norm based problem, which requires combinatorial search. For this purpose we need to approximate the l_0 norm with a smoother and easier to differentiate function $f(\|X\|, \sigma)$, which requires more samples than l_0 norm based problem, but less than l_1 and l_2 norm based problems (see Experiments 5 and 6 for further details).

Consider a system of linear equations of N unknowns and M equations defined as:

$$Y_{M\times 1} = C_{M\times N} X_{N\times 1} + e_{M\times 1}, \tag{9.36}$$

where e is a small unknown error in the sampling process with $0 \leq \|e\|_{l_2} \leq 1$, C is the Sensing matrix, Y is the collected samples vector and X is the unknown vector we wish to recover. From the theory of Linear Algebra, we have three distinct cases [28, 33, 159]:

1. $M = N$: Square system where C has full rank, $Y \neq 0$ and unique solution $X = C^{-1}Y$.

2. $M > N$: Over-determined system, where the best solution is known to be the least-squares solution calculated as min $\|Y - CX\|_{l_2}$.

3. $M < N$: Under-determined system, with singular C and estimation of min $\sup(X) = \sharp\{1 \leq i \leq N : X_i \neq 0\}$ from Y. The solution with the fewest non-zero ele-

ments (less than M due to the null-space of C) is the sparsest solution derived from $\min \ \|X\|_{l_0}$ s.t. $CX = Y$.

Under this scheme (lower and higher dimensional spaces), we will now investigate how sparse recovery can be achieved using the l_0 heuristic. We will assume that the Sensing matrix C is randomly generated with values between zero and one, with and without normalised entries (the l_2 norm of each column sums to one).

Figure (9.16) presents the sparse recovery of a purely random signal in a form of vector X, which is initially unknown, under different sample sizes, namely $M = 1, 2, \ldots, 200$. We will also assume that the initial signal is of a fixed length of 100 unknown elements and with unknown support (sparsity level). The samples vector Y and the Sensing matrix C are real-valued vectors with random values between zero and one, generated using the standard distribution. The parameters of the l_0 heuristic are $T = 200$ iterations, $S = 12$ swarms, σ decrease factor $\beta = 0.5$, thres $= 10^{-10}$ as a sparsity level threshold for the calculation of the sparsity level K using the absolute value of back-projection $|C^T Y|$, i.e. $K = \text{length}(\text{find}|C^T Y|_{l_0} > \text{thres}))$, minimum and maximum values for the recovery of X as $\min(X) = \min(|C^T Y|)$ and $\max(X) = \|C^T Y\|_{l_\infty}$, while initial σ value is set as $\sigma^{(0)} = 2\|C^T Y\|_{l_\infty}$. For the calculation of the recovery error in the experiments, the relative recovery error was used based on the samples vector, $\|C\hat{X} - Y\|_{l_2}/\|Y\|_{l_2}$.

Note that as the sample size increases (up to 20 samples), the recovery error decreases (up to 10^{-1}), which is expected since more samples contribute to the stable recovery of the initial compressed vector. However, for sample sizes greater than 60 the recovery error remains the same as more samples do not improve the quality of the solution. This stable relatively high error after 20 samples can be attributed to the fact that no prior knowledge of the structure (minimum/maximum values and sparsity level) of the initial vector is known, since both Y and C are completely random and not related (Y is not derived from CX). If prior knowledge of the initial vector's structure was known then the recovery error would steadily decrease beyond 10^{-1} achieving values less than 10^{-7} for samples more than 80, which is the case in Figure (9.17), where the sparsity level and minimum/maximum values of the original vector are known. Also notice that the use un-normalised Sensing matrix C is preferable for sample sizes less than 10, while for sample sizes greater than 10 the use of the normalised Sensing matrix is preferred due to slightly better results. However, in general, it

can be seen that if no prior knowledge of the initial vector is available then efficient recovery is not possible in both under-sampled and over-sampled cases.

Figure (9.17) presents sparse recovery of a real-valued random vector of 80 elements with known values between zero and one, randomly generated using the Normal distribution. The sparsity level has been chosen as 10% ($K = 8$), while the samples size ($M = 1, 2, \ldots, 100$) has been generated randomly with values between zero and one, using the Normal distribution with and without normalised columns. The l_0 heuristic's parameters are $T = 70$ iterations, $S = 12$ swarms, σ decrease factor $\beta = 0.5$ and initial $\sigma^{(0)} = 2\max(X)$ (twice the maximum entry of initial vector). The relative recovery error used in this experiment was calculated using MSE, $\|\hat{X} - X\|_{l_2}/\|X\|_{l_2}$, as a measure of the quality of the estimate $\hat{X}$. It can be easily seen that as the number of samples increases the recovery error decreases, in contrast to Figure (9.16), since prior knowledge of the initial vector is available during the recovery process. Notice that efficient recovery (error less than 10^{-1}) is achieved after 50 samples, due to the small number of iterations (70) chosen for the l_0 heuristic. If the number of iterations was chosen to be more than 150 then efficient recovery (error less than 10^{-1}) would have been achieved after 30 samples. In general, we can see that the recovery error keeps decreasing as the number of samples increases, with recovery error 10^{-5} for 50 samples and 10^{-8} for more than 70 samples. Also notice that the use of un-normalised entries in the Sensing matrix (its columns are not unit normed) achieves lower recovery error for sample sizes between 20 and 70 (under-determined case). For more than 70 samples and particularly for the over-determined cases of more than 80 samples both the use of normalised and un-normalised entries in the Sensing matrix C achieve the same results (10^{-8} as a recovery error). Therefore, in under-determined cases Sensing matrices with un-normalised entries are preferred, while in over-determined cases the use of normalised or un-normalised entries for Sensing matrix C do not affect the efficiency of the recovery using the l_0 heuristic. As a final note, the small fluctuations in the recovery error for samples between 1 and 40 samples can be attributed to the stochastic nature (randomness) of both the measurements collection and solution generation of the l_0 heuristic.

Figure (9.18) presents sparse recovery of a weakly sparse complex-valued randomly generated vector, from different noisy sample levels $M = 1, 2, \ldots, 70$. The collection of noisy samples can be modelled as $Cf(t) + \epsilon = Y$, where $f(t) = exp(-3ut)$ is the complex-valued signal of 50 elements with time domain $t = 1, 2, \ldots, 50$ and $u = 0.1 + 0.5i$. Y is the samples

vector representing the noisy measurements collected using the Sensing matrix C, with and without normalised entries, randomly generated from the Normal distribution with values between zero and one. The ϵ parameter is a small additive bounded error which is randomly generated $0 \leq \|\epsilon\|_{l_2} \leq 1$ and represents noise as a sampling imprecision. Here we assume that the vector $f(t)$ is strictly complex, which means that its non-zero elements have both real and imaginary parts, so as to decompose it into real and imaginary parts in the recovery process. Also, similarly to other sparse recovery approaches, such as the TV minimisation, the l_0 heuristic samples and recovers the real and imaginary parts of the signal separately, assuming the same pattern of sparsity. This is an important assumption on the design of the l_0 heuristic and how the recovery of complex-valued signals is achieved.

The l_0 heuristic's parameters are $T = 100$ iterations, $S = 12$ swarms, σ decrease factor $\beta = 0.5$ and initial $\sigma^{(0)} = 2\max(f(t))$ (twice the maximum entry of initial vector). Since the initial signal is weakly sparse we use thres $= 10^{-50}$ as a sparsity level threshold for the calculation of $K = \text{length}(\text{find}|f(t)|_{l_0} > \text{thres}))$. This threshold has been chosen to simulate the sparsity of the vector in the collection of its noisy measurements. Smaller value for the threshold will decrease both sparsity level and recovery error (at the expense of more samples), as any value smaller than the threshold is ignored in the recovery process. On the other hand, higher value for the threshold will increase both sparsity level and recovery error, requiring less samples. The relative recovery error was calculated using SNR defined in Equation (9.11), as a measure of the quality of the estimate $\hat{f(t)}$. Note that from logarithm expansion rule, SNR ratio can be re-written as $10\log_{10}(\|f(t)\|_{l_2}) - 10\log_{10}(\|f(t) - \hat{f(t)}\|_{l_2})$, which implies that $SNR \geq 0$ if $10\log_{10}(\|f(t) - \hat{f(t)}\|_{l_2}) < 0$ since $10\log_{10}(\|f(t)\|_{l_2}) \lesssim 0$. Therefore, efficient recovery with de-noising (removing small error) is possible for sample sizes greater than 50 samples where the SNR is positive (equal or greater than the signal length). This is expected as very low sparsity levels (we ignore any value smaller than 10^{-50}) require more samples for efficient recovery. Small fluctuations within the region of 5 and 50 samples in the plot can be attributed partially to the stochastic nature of the l_0 heuristic and partially to the fact that small noise levels $\epsilon \approx 0.01 - 0.05$ have not been removed from the recovery process and thus been kept in the estimate $\hat{f}(t)$ after recovery. Again, similarly to Figure (9.16) the use of un-normalised Sensing matrix C is preferable for sample sizes greater than 50, while for sample sizes smaller than 50 the use of both the normalised and the un-normalised Sensing matrix derive the same results.

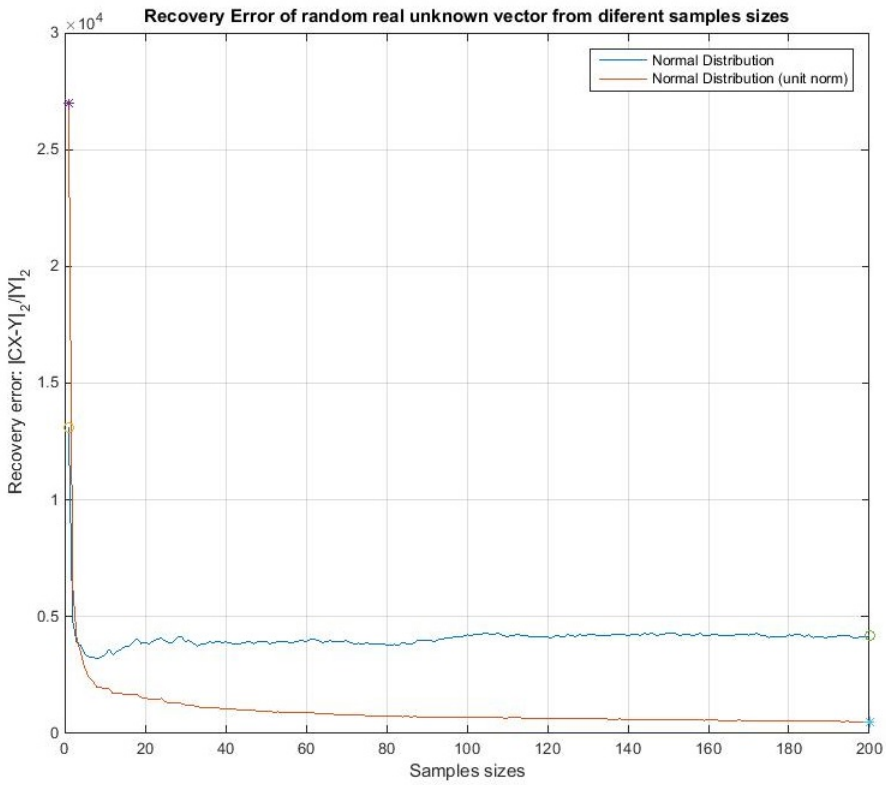

Figure 9.16: Sparse recovery of a completely random signal X under different sample sizes $M = 1, 2, \ldots, 200$. The signal is represented in a form of vector with unknown real-valued initial values but with known length of 100 elements. The samples vector Y and the Sensing matrix C are real-valued vectors with random values between zero and one, generated using the standard distribution. The parameters of the l_0 heuristic are $T = 200$ iterations, $S = 12$ swarms, σ decrease factor $\beta = 0.5$, thres $= 10^{-10}$ as a sparsity level threshold for the calculation of K using the absolute value of back-projection $|C^T Y|$, i.e. $K = \text{length}(\text{find}|C^T Y|_{l_0} > \text{thres}))$, minimum and maximum values for the recovery of X as $\min(X) = \min(|C^T Y|)$ and $\max(X) = \|C^T Y\|_{l_\infty}$, while initial sigma value is set as $\sigma^{(0)} = 2\|C^T Y\|_{l_\infty}$. The relative recovery error was calculated using the samples vector as $\|C\hat{X} - Y\|_{l_2}/\|Y\|_{l_2}$ (experiment7.m).

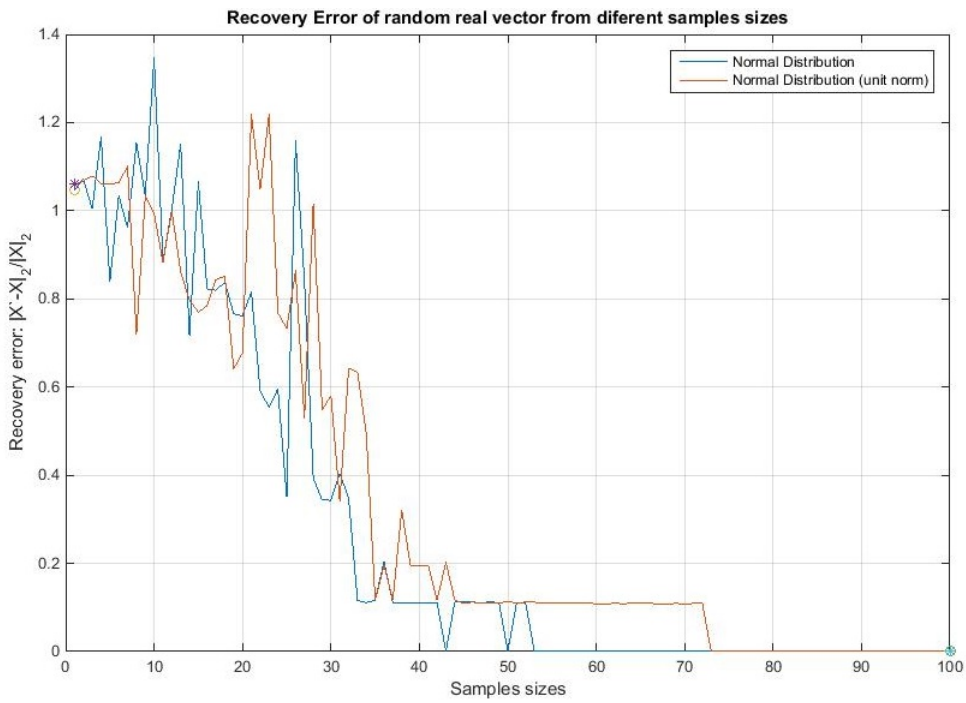

Figure 9.17: Sparse recovery of a real-valued random vector of 80 elements with values between zero and one, randomly generated using the Normal distribution. The sparsity level has been chosen as 10% ($K = 8$), while the samples size ($M = 1, 2, \ldots, 100$) has been generated randomly with values between zero and one, using the Normal distribution with and without normalised columns. The l_0 heuristic's parameters are $T = 70$ iterations, $S = 12$ swarms, σ decrease factor $\beta = 0.5$ and initial $\sigma^{(0)} = 2\max(X)$. The relative recovery error was calculated using MSE, $\|\hat{X} - X\|_{l_2}/\|X\|_{l_2}$, as a measure of the quality of the estimate $\hat{X}$ (experiment7.m).

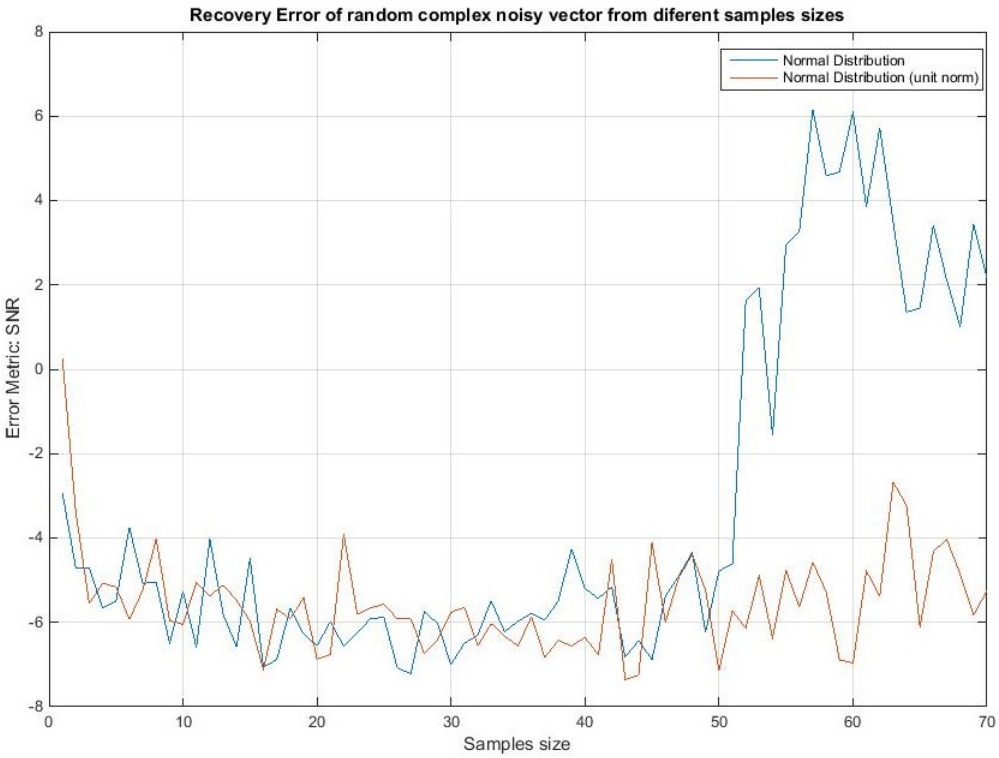

Figure 9.18: Sparse recovery of a weakly sparse complex-valued randomly generated vector, from different noisy sample levels $M = 1, 2, \ldots, 70$. The collection of noisy samples can be modelled as $Cf(t) + \epsilon = Y$, where $f(t) = exp(-3ut)$ is the complex-valued signal of 50 elements with time domain $t = 1, 2, \ldots, 50$ and $u = 0.1 + 0.5i$. Y is the samples vector representing the noisy measurements collected using the Sensing matrix C, with and without normalised entries, randomly generated from the Normal distribution with values between zero and one. The ϵ parameter is a small additive bounded error which is randomly generated $0 \leq \|\epsilon\|_{l_2} \leq 1$ and represents noise as a sampling imprecision. The l_0 heuristic's parameters are $T = 100$ iterations, $S = 12$ swarms, σ decrease factor $\beta = 0.5$ and initial $\sigma^{(0)} = 2\max(f(t))$. Since the initial signal is weakly sparse we use thres $= 10^{-50}$ as a sparsity level threshold for the calculation of $K = \text{length}(\text{find}|f(t)|_{l_0} > \text{thres}))$. The relative recovery error was calculated using SNR defined in Equation (9.11), as a measure of the quality of the estimate $\hat{f(t)}$ (experiment7.m).

9.4.8 Different Sensing matrices and design

In this experiment we will test the performance of the l_0 heuristic against different sampling matrices which do not satisfy UUP and RIP properties. In all the previous experiments we used sampling matrices with elements generated randomly using the Normal distribution. Normal, Rademacher and database friendly distributions are the only known distributions which satisfy RIP/UUP conditions with high probability [39, 42, 215]. This is the main reason why sampling matrices are designed using these probability distributions. Using these probability distributions we are able to express sparsely any vector (signal) of high dimensional space into a low dimensional space and be still able to reconstruct them by finding the sparsest or the most compressible estimate using only a small number of measurements [41, 50, 189]. Based on these conditions sparse recovery methods, such as LASSO, OMP, AIHT, IRLS and l_1 minimisation principle, are able to recover the sparsest solution from an under-determined system of linear equations [40, 41, 43]. In fact, sampling matrices that satisfy UUP and RIP properties are a generalisation of rectangular orthogonal or orthonormal matrices, whose columns are unit normed [189].

However, all these cases are too ideal for efficient recovery of a vector, as most of the times we usually do not know how the available measurements were collected. Also, there is no particular theoretical framework for the construction of these matrices since they are all probabilistic in nature and thus there is no mathematical guarantee that they will always work [50, 189]. In fact, it has been shown that the search and test for the most suitable among all matrices in a given class or distribution that satisfies either RIP or UUP properties requires an exponential time algorithm [40, 42, 189]. Moreover, it has also been proven that the recovery error of a vector sampled using them is exponentially small based on the size of the sampled vector and for relatively small noisy levels [39, 41, 43]. In cases where the level of noise is very high the failure rate is also very high [189]. The aim here is to test the recovery of a sparse vector for under-sampling ratios by using different distributions to construct the sampling matrix and measure how this affects the performance of the recovery. These numerical simulations are very important in order to give some further insights about the practical behaviour of these Sensing matrices.

The probability distributions used for the design of the sensing matrix in this experiment are: Normal (with and without unit norm columns), Uniform, Bernoulli, Exponential,

Geometric, Poisson and Pareto distributions. In all these cases the values of the sampling matrix C elements were between the interval $[0, 1]$. Assuming a random variable c_i (i-th element of matrix C) then the probability density function ($Pr(c_i)$) of the previous probability distributions can be defined as follows [108, 115, 140, 208]:

1. Uniform (continuous) probability distribution as a random number generator on the interval $[0, 1]$ with probability density function:

$$Pr(c_i) = \frac{1}{b-a}, \quad a \leq c_i \leq b, \tag{9.37}$$

   where $a = 0$ and $b = 1$; with mean $(a + b)/2 = 1/2$ and variance $(b - a)^2/12 = 1/12$.

2. Bernoulli (zero-one) probability distribution as a random number generator of 1 with probability $p = 1/2$ and 0 with probability $1 - p = 1/2$:

$$Pr(c_i = 0) = Pr(c_i = 1) = 1/2. \tag{9.38}$$

3. Normal (Gaussian) continuous probability distribution as a random number generator on the interval $[0, 1]$ with mean $\mu = 0$ and standard deviation $\sigma = 1$ and probability density function:

$$Pr(c_i|\mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp(\frac{-(c_i - \mu)^2}{2\sigma^2}), \tag{9.39}$$

   with and without unit norm entries (l_2 norm sums to one for each column of C).

4. Exponential (continuous) probability distribution as a random number generator on the interval $[0, 1]$ with parameter $\lambda = 0.1$ and probability density function:

$$Pr(c_i) = \lambda \exp(-\lambda c_i). \tag{9.40}$$

   It has mean $1/\lambda = 10$ and variance $1/\lambda^2 = 100$.

5. Geometric (discrete) probability distribution as a random number generator on the

interval $r \in [0,1]$ with probability of success $p = 1/2$ and probability mass function:

$$Pr(c_i = r) = p(1-p)^{r-1}, \quad r = 0,1 \quad p = 1/2 \tag{9.41}$$

It has mean $1/p = 2$ and variance $(1-p)/p^2 = 2$.

6. Poisson (discrete) probability distribution as a random number generator on the interval $r \in [0,1]$ with parameter $\lambda = 5$ and probability mass function:

$$Pr(c_i = r) = \exp(-\lambda)\lambda^r/r!, \quad r =\in [0,1]. \tag{9.42}$$

It has mean and variance both equal to $\lambda = 5$.

7. Pareto (continuous) distribution as a random number generator on the interval $r \in [0,1]$ with parameters $a = 10/13 = 1/1.3$, $b = 1$ and probability density function:

$$Pr(c_i) = ab^a/c_i^{a+1}, \quad a = 10/13, \quad b = 1. \tag{9.43}$$

It has mean $ab/(a-1) = -10/3$ and variance $ab^2/((a-1)^2(a-2)) = -845/72$.

Figure (9.19) presents the sparse recovery of a 100 element real-valued vector of 10% sparsity, which is randomly generated with values between zero and one (using the Normal distribution) under different sample sizes ($M = 11, 12, \ldots, 200$) drawn from different probability distributions (normal, uniform, zero-one, exponential, geometric, poisson and pareto). The efficiency of the sparse recovery was measured using the relative recovery error metric defined in (9.10), namely $\|X - \hat{X}\|_{l_2}/\|X\|_{l_2}$, where X and $\hat{X}$ is the original and the recovered vector respectively. The l_0 heuristic parameters used for this experiment are $T = 150$ iterations, $S = 12$ swarms, σ decrease factor $\beta = 0.5$ and initial $\sigma^{(0)}$ twice the maximum element of the initial random real-valued vector. Notice that efficient recovery using the l_0 heuristic is possible for more than 40 measurements regardless of the choice of the probability distribution used for the collection of samples. For 40 measurements the recovery error is less than 10^{-5}, while as the number of measurements increases the error decreases with smallest value of 10^{-20} for 200 measurements, similar for all the probability distributions used.

It is clear that using the l_0-norm based optimisation principle we can recover any real-valued sparse vector from its random measurements, using different probability distributions

and without any significant changes in the recovery error. For example, the recovery error of a 100 element real-valued test vector with 10% sparsity using 60 samples is the same, approximately 10^{-8}, using any of the probability distributions presented here. Note that among these probability distributions only Normal distribution satisfies the RIP/UUP properties which are required for efficient sparse recovery based on the l_1 and l_2 norms. As a brief reminder RIP/UUP properties state that $\|CX\|_{l_2} \approx \|X\|_{l_2}$, for any sparse vector X given that C follows RIP/UUP, in order for the l_0-norm based optimisation principle to be relaxed by either its convex analogue, namely the l_1-norm based optimisation principle, or the re-weighted l_2-norm based optimisation principle and its variations.

Another important aspect is that the number of random measurements required for efficient recovery is not affected by the way we collect them. Therefore, 40 samples appear to be the threshold for efficient recovery despite the choice of the collection of samples (i.e. probability distribution). However, note that there are no generalised theoretical guarantees, treated in the literature, for efficient recovery of any sparse vector using the l_0-norm based optimisation principle for under-sampled ($M \leq 100$) or over-sampled $M \geq 100$ measurements, drawn from distributions which do not follow RIP/UUP properties. As a final note, the small fluctuations of the recovery error for smaller than 40 samples, in all the probability distributions, can be attributed to the randomness of the collection of samples and the stochastic step of the solution generation of the l_0 heuristic.

9.4.9 Experiments in transform domains

In this experiment we study how the performance of the l_0 heuristic is affected by different transform domains, namely Discrete Fourier, Discrete Cosine and Discrete Walsh-Hadamard transforms. These Unitary transforms (orthogonal transforms) are mainly used to represent either smooth or locally periodic behaviours in signals/images and thus allow the necessary operations to be efficiently performed. This compressibility constitutes the foundation of transform coding which is commonly used in many compression standards in audio and images, such as MP3, AAC and JPEG [186]. Obviously, the best transform domain is the one that leads to the sparsest representation which is highly based on the nature of the signal/image, or more precisely on the trade-off between the computation time and the complexity of the analysis (the size of the dictionary of the transform) [132, 186, 215].

For simplicity reasons we will restrict our experiments in only these three transform

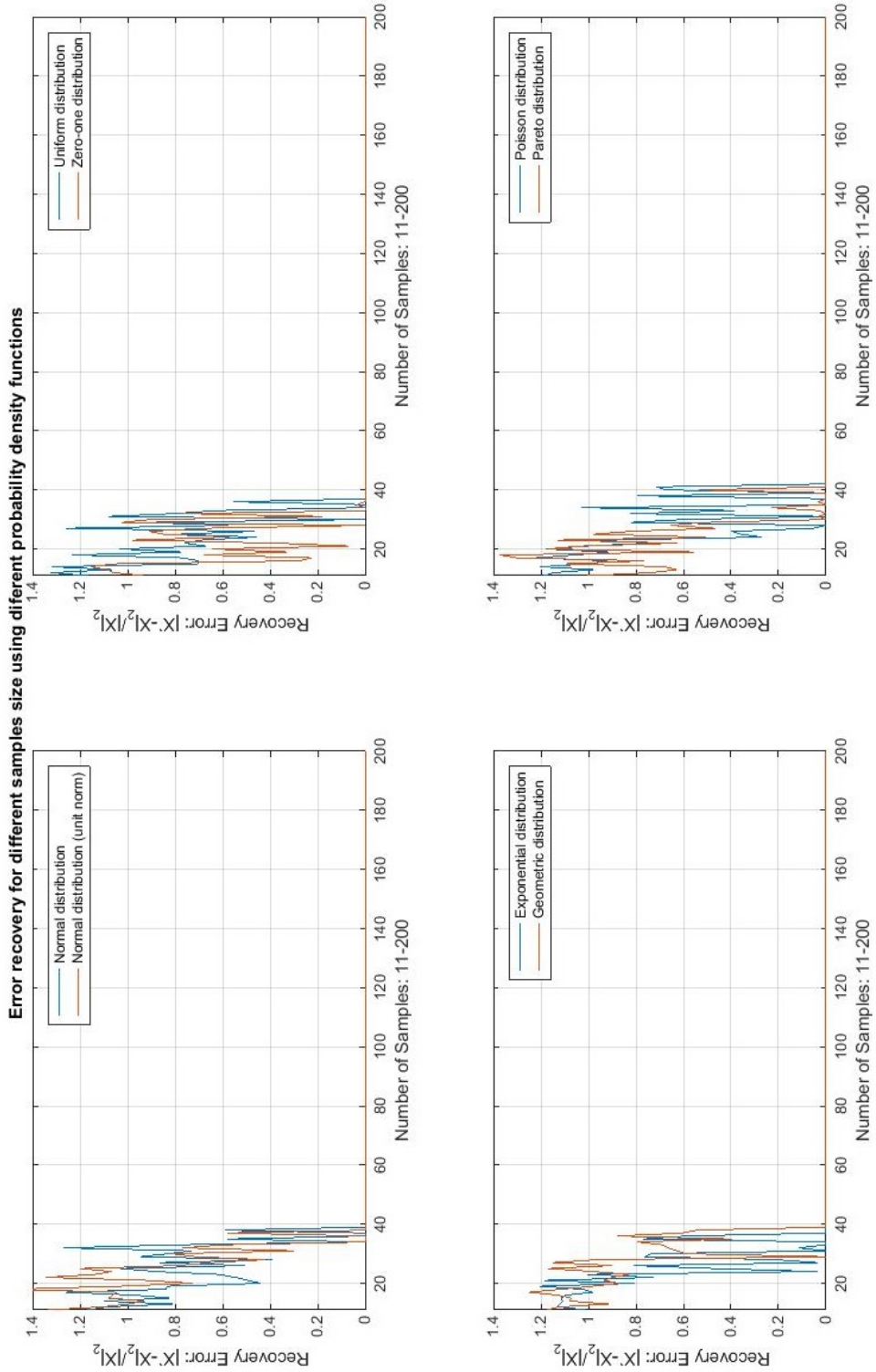

Figure 9.19: Sparse recovery of a 100-element real-valued vector with 10 non-zero entries randomly generated and sampled against different levels ($M = 11, 12, \ldots, 200$), using different probability distributions; normal (with and without unit norm columns), Uniform, Zero-one, Exponential, Geometric, Poisson and Pareto distributions.The l_0 heuristic parameters are $T = 150$ iterations, $S = 12$ swarms, σ decrease factor $\beta = 0.5$ and initial $\sigma^{(0)}$ twice the maximum element of the initial vector (experiment8.m).

domains, namely DFT, DCT and FWHT applied in signals, as the most representative ones and we will investigate how the sparsity of a test signal is affected in contrast to its original domain of representation. We also need to note that we will not consider cases where a transform is used for general signal enhancement techniques, while only the generalised Discrete Fourier transform will be used here (for further details see Appendix B). Wavelet, Ridgelet and Curvelet transforms which are natural extensions of Fourier transform (i.e. they also support angular alignment information, the length of the alignment and multiple scales) will not be covered here as they are beyond the scope of this thesis. For further details on these relatively new transforms as a natural extension of the traditional Wavelets together with their geometric interpretation can be found in [132, 186].

As a quick reminder we will now briefly describe the computations of the forward and inverse transforms used in this experiment, namely the Discrete Fourier transform (DFT), the discrete cosine transform (DCT) and the Fast Walsh-Hadamard transform (FWHT). The Discrete Fourier transform (DFT) $\mathcal{F}(f) = \hat{f} : \mathbb{R}^N \to \mathbb{C}^N$ of a finite-dimensional signal in time domain $f(t) \in \mathbb{R}^N$ can be defined as [94, 132, 186]:

$$\hat{f}(\omega) = \sum_{t=1}^{N} f(t) e^{(-2\pi i \omega t)/N}, \tag{9.44}$$

where $\omega = 1, ..., N$ and $\hat{f}(\omega)$ is the transformed signal (spectrum). The inverse Fourier transform given the Fourier coefficients $\hat{f}(\omega)$ for all frequencies $\omega \in \mathbb{Z}_N$ can be defined as [94, 132, 186]:

$$f(t) = \frac{1}{N} \sum_{\omega=1}^{N} \hat{f}(\omega) e^{(2\pi i \omega t)/N} \tag{9.45}$$

for all frequencies $\omega = 1, ..., N,$, where $f(t)$ is the signal we want to recover.

The Discrete Cosine transform (DCT) of a finite-dimensional signal in time domain $f(t) \in \mathbb{R}^N$ can be defined as a set of positive values as [94, 132, 186]:

$$f(k) = w(k) \sum_{t=0}^{N-1} f(t) \cos(\frac{\pi k(2t+1)}{2N}), \tag{9.46}$$

where $k = 0, 1, 2, \ldots, N-1$, N is the signal length and $w(k)$ is a normalisation constant:

$$w(k) = \begin{cases} \sqrt{\frac{1}{N}} & k = 0, \\ \sqrt{\frac{2}{N}} & 1 \leq k \leq N-1. \end{cases} \tag{9.47}$$

The inverse Discrete Cosine transform is defined as [94, 132, 186]:

$$f(t) = \sum_{k=0}^{N-1} w(k) f(k) \cos(\frac{\pi k(2t+1)}{2N}), \tag{9.48}$$

Finally, the Walsh-Hadamard transform ($y(k)$), also known as Hadamard transform, is a non-sinusoidal, orthogonal transform, very similar to Fourier transform [184, 207]. This transform decomposes a finite-dimensional signal in time domain $f(t) \in \mathbb{R}^N$ into a set of basis functions, called Walsh functions, which are rectangular or square waves with values of $+1$ or 1 [94, 184, 207]. These return values are called sequency values, as a more generalized notion of frequency [94, 184, 207]. Each Walsh function has a unique sequency value, used for the recovery of the original signal. The FWHT and IFWHT are defined as [94, 184, 207]:

$$y(k) = 1/N \sum_{t=0}^{N-1} w(k,t) f(t), \quad k = 0, 1, \ldots, N-2 \tag{9.49}$$

$$f(t) = \sum_{k=0}^{N-1} w(k,t) y(k), \quad t = 0, 1, \ldots, N-1 \tag{9.50}$$

where k values represent the intervals for the calculation of the Walsh functions $w(k,t)$, whose values $w(k)$ depend on time $t = 0, 1, \ldots, N-1$. Walsh functions are allowed to have only two different values, namely -1 and 1, based on the input (signal) values at fixed intervals and with initial state ($w(0,t) = 1$) always $+1$, satisfying orthogonality (see Section 2.3 and Appendix B for more details). For example [184], the first four values of Walsh functions on a time $t \in [0, 1)$ are: $w(0) = 1$ on $[0, 1)$, $w(1) = 1$ on $[0, 1/2)$ and $w(1) = -1$ on $[1/2, 1)$, $w(2) = 1$ on $[0, 1/4) \cup [1/2, 3/4)$ and $w(2) = -1$ on $[1/4, 1/2) \cup [3/4, 1)$, $w(3) = 1$ on $[0, 1/4) \cup [3/4, 1)$ and $w(3) = -1$ on $[1/4, 3/4)$. Walsh functions exhibit many important

and useful properties in the area of signal processing and analysis. For further details see [184, 207].

Figure (9.20) represents a $N = 1024$ element real-valued signal with two cosines in time domain $t = 1, 2, \ldots, N-1$. The red asterisks represent the position of the 300 samples to be collected in the transform domain. The signal is of the form:

$$f(t) = \cos(2\pi(12/1024)t) + \cos(2\pi(39/1024)t), \tag{9.51}$$

Notice that the signal $f(t)$ is not sparse in time domain, as almost all its values are non-zero. Figure (9.21) now represents the original signal $f(t)$ under different transform domains (only real part of $f(t)$ in a form of absolute values), namely real-valued domain (in DCT), frequency domain (in DFT) and sequency domain (in FWHT). Notice that in DFT transform we can clearly see four peaks (sparsity level is four), while in FWHT there are several smaller peaks which make the original signal weakly sparse in FWHT domain. The same problem appears in DCT, where the elements in positions between 100 and 550 are close to zero but not clearly zero, which make the original signal approximately sparse in DCT domain. Figure (9.22) presents the collection of 300 samples (absolute value), under different transform domains (DCT, DFT and FWHT), to be used in sparse recovery of the signal $f(t)$ by the l_0 heuristic. These samples are collected using normal un-normalised distribution with zero mean and one variance.

Figure (9.23) presents the sparse recovery of the original signal $f(t)$ using under-sampled measurements from DCT, DFT and FWHT. The parameters of the l_0 heuristic are $T = 300$ iterations, $S = 12$ swarms, σ decrease factor $\beta = 0.5$ and initial $\sigma^{(0)} = 2\|f(t)\|_{l_\infty}$ (twice the maximum element of the initial signal). For the calculation of the recovery error we used the relative recovery error metric defined in Equation (9.10), namely $\|f(t) - \hat{f}(t)\|_{l_2}/\|f(t)\|_{l_2}$, where $f(t)$ and $\hat{f}(t)$ is the original and the recovered signal of the corresponding transform domain. However, since the signal's values are complex in frequency domain, we separated the complex signal using magnitude R and phase angle θ, sampling in magnitude which has real values and then use phase angle to project back to complex values where the inverse transform is applied for obtaining the signal in time domain again. For a complex-valued signal $f(t)$ the magnitude R (with real values) and the phase angle θ (measured in radians

with values $\pm\pi$) are given as [94, 157, 205]:

$$f(t) = R\exp(i\theta), \tag{9.52}$$

where $R = |f(t)|$ represents the absolute values of $f(t)$ in vector format and $\theta = \text{angle}(f(t))$ is the phase angle of the signal.

Notice in Figure (9.23) the difference in sparsity levels between the different transform domains, which actually affect the efficient recovery of the signal $f(t)$ from its partial measurements. Also note that the recovered signal, using DFT, is very close to the original, though it has been compressed to one third of its size. This can be attributed to the high levels of sparsity in terms of storing the transform coefficients of the signal in the frequency domain. In the other two cases, namely in FWHT and DCT we can notice artifacts in the recovery of the signal (high recovery error) which can be attributed to the low level of sparsity in these transform domains. In general, efficient recovery for highly incomplete measurements ($M = 300$) can only be achieved under high levels of sparsity using the l_0 heuristic as a sparse recovery method. In fact, the l_0 heuristic does not support efficient recovery of weakly sparse vectors under highly under-sampled measurements. This assumption is similar to other sparse recovery methods including l_1, l_2 minimisation principles, OMP and LASSO methods. In weakly sparse vectors more samples are required for efficient recovery due to the fact that more non-zero values are mapped to zero during sampling (null space of the sampling matrix C).

9.4.10 Application of filters

In this experiment we investigate how the use of different types of filters can enhance the performance of the l_0 heuristic in sparse vector recovery under noisy under-sampled and over-sampled random measurements. As a reminder filter is simply a function which can be applied after the sparse vector is recovered so as to remove undesired elements obtained during the sampling process, such as random variations and other inaccuracies attributed to noise (noisy channel of measurements, noisy environment, malfunctions in the sampling process, etc.). One of the simplest form of filters, proposed by Donoho and Johnstone in 1994, is the median absolute deviation where every element X_i of a vector X known to be

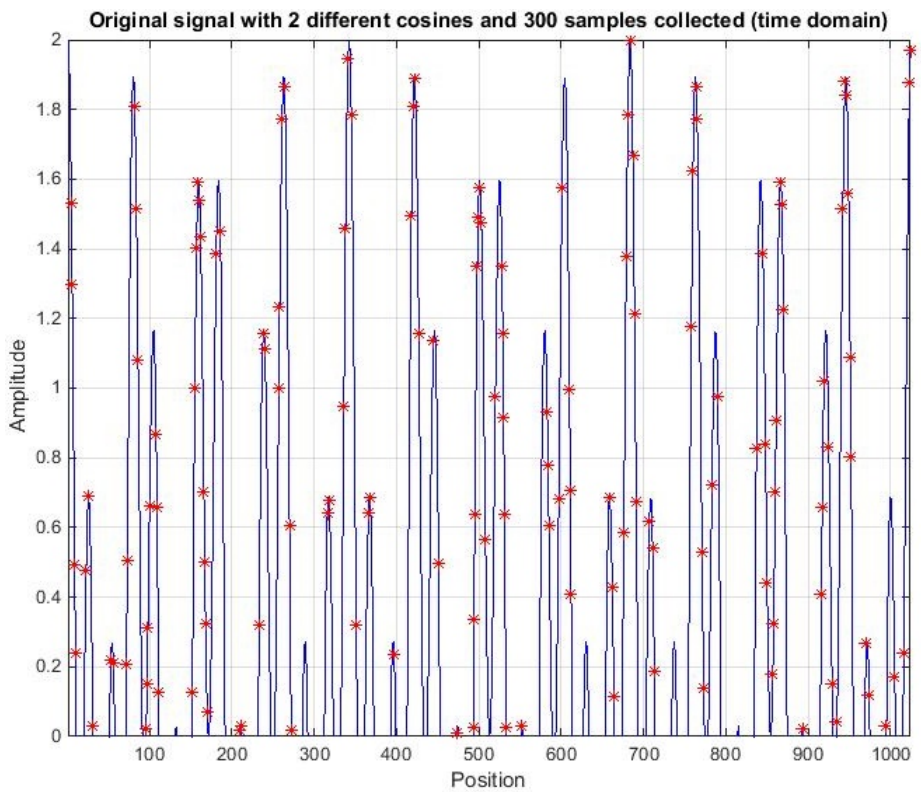

Figure 9.20: Plot of the original signal of $N = 1024$ elements length of the form $f(t) = \cos(2\pi(12/1024)t) + \cos(2\pi(39/1024)t)$ represented in time domain $t = 1, 2, \ldots, N-1$. The red asterisks represent the position of the collection of 300 samples in the frequency domain. The parameters of the l_0 heuristic are $T = 300$ iterations, $S = 12$ swarms, σ decrease factor $\beta = 0.5$ and initial $\sigma^{(0)} = 2\|f(t)\|_{l_\infty}$ (twice the maximum element of the initial vector). Note that in time domain the signal $f(t)$ is not sparse (experiment9.m).

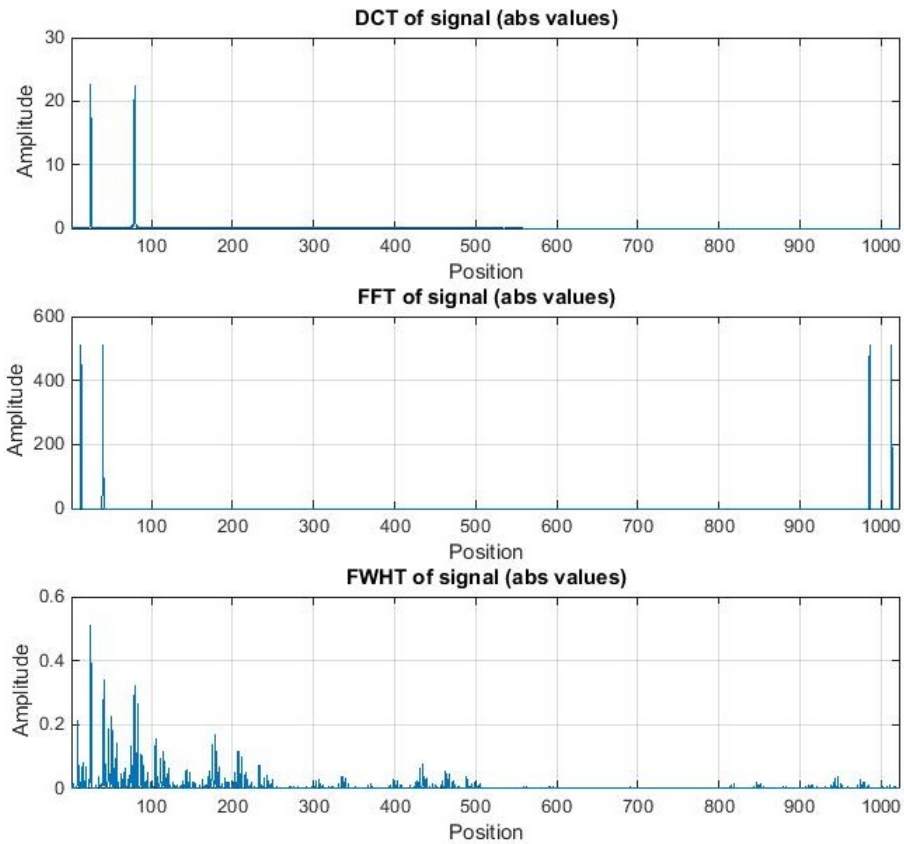

Figure 9.21: Representation of the original signal $f(t)$ in transform domain using different unitary transforms, DCT, DFT and FWHT (only real part of signal in a form of absolute values). Notice that in DFT transform we can clearly see four peaks (sparsity level is four), while in FWHT there are several smaller peaks which make the original signal weakly sparse in FWHT domain. The same problem appears in DCT, where the elements in positions between 100 and 550 are close to zero but not clearly zero, which make the original signal approximately sparse in DCT domain (experiment9.m).

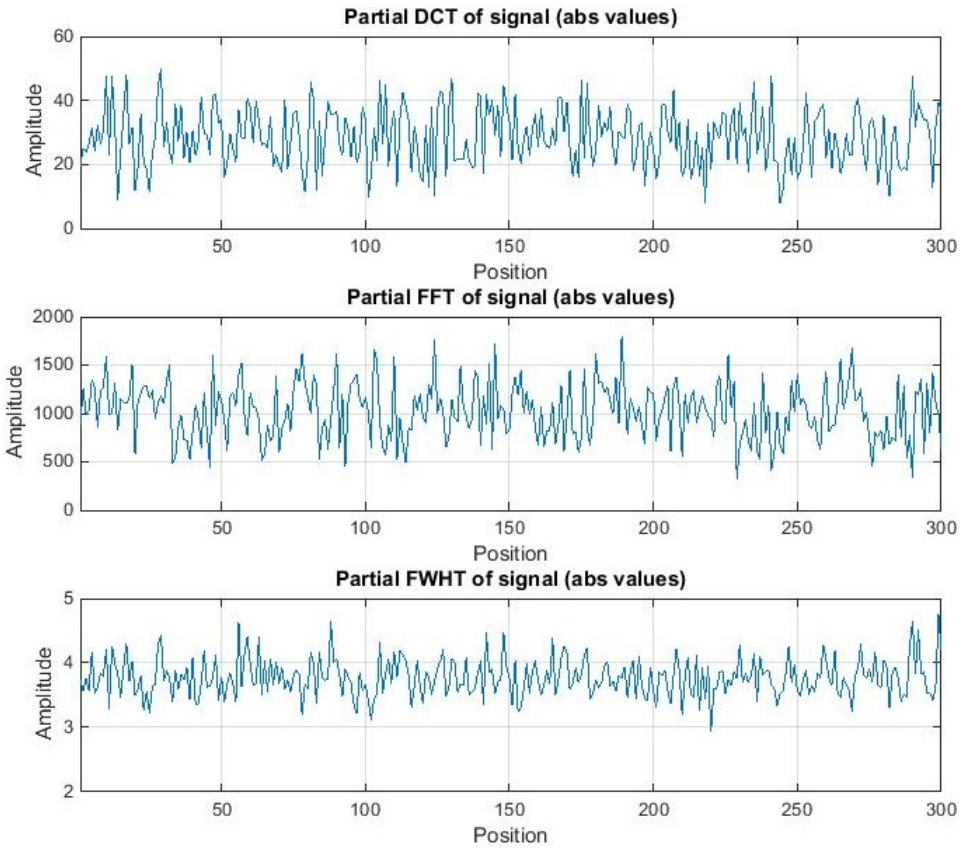

Figure 9.22: Representation of samples collection ($M = 300$) under different transform domains, namely DCT, DFT and FWHT. These samples will be used for the recovery of the original signal under different transform domains (real-valued, frequency and sequency domain respectively) using the l_0 heuristic (experiment9.m).

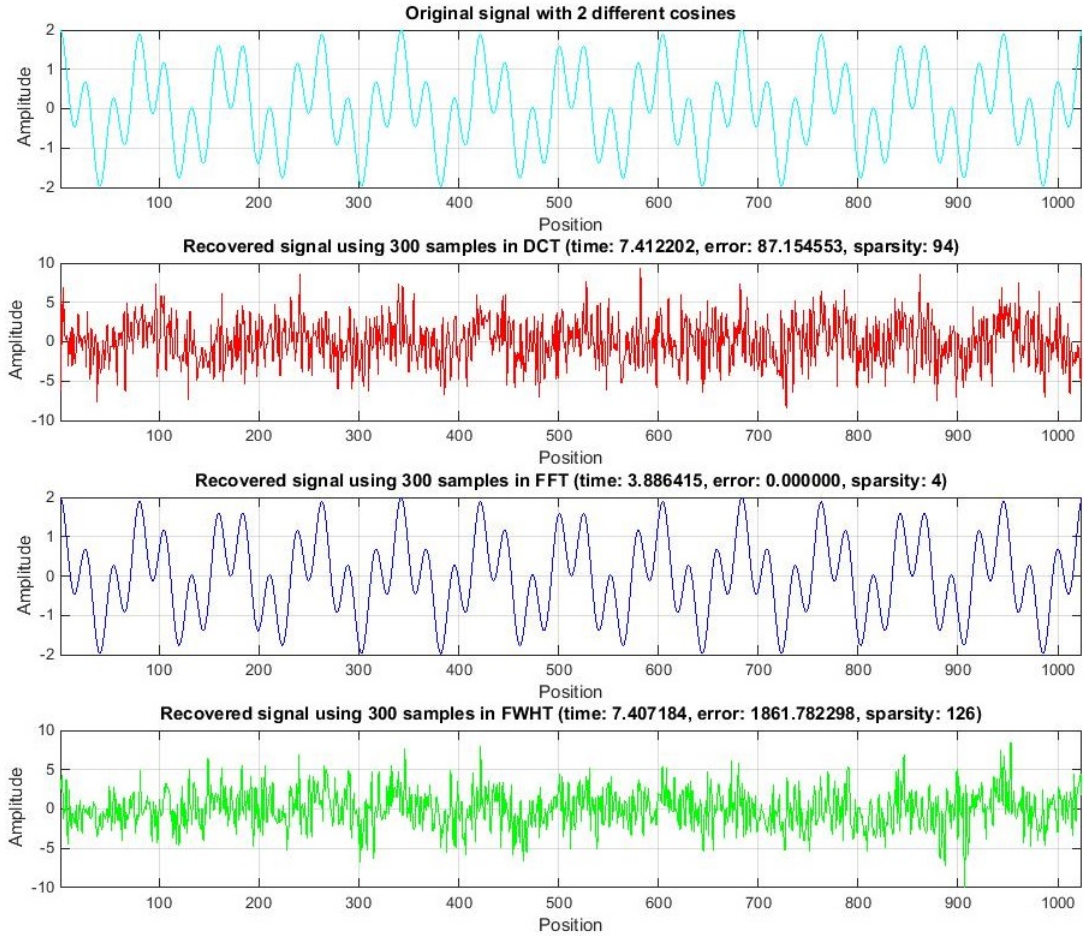

Figure 9.23: Sparse recovery of the original signal using highly under-sampled (partial) measurements from DCT, DFT and FWHT. Notice the difference in sparsity levels between the different transform domains, which actually affect the efficient recovery of the signal from its partial measurements in the frequency domain. Using DFT we achieve exact recovery due to the high level of sparsity using this transform. In the other two cases on FWHT and DCT we can notice small artifacts in the recovery of the signal which can be attributed to the low level of sparsity in the FWHT and DCT domain (experiment9.m).

sampled under noise is recalculated using its median value as a smoothing filter [132, 186]:

$$X_i = X_i - \text{median}(\|X_i - \text{median}(X_i)\|_{l_1})/0.6745 \tag{9.53}$$

Note the value 0.6745 which is a small constant used in the filter and was found experimentally by the authors based on the coefficients obtained from the wavelet decomposition structure. Here, we will use a similar approach with the mean absolute deviation, since through experimentation the median absolute deviation has been found to fail to improve the l_0 norm based heuristic results. The mean absolute deviation filter can now be defined as:

$$X_i = X_i - \text{mean}(\|X_i - \text{mean}(X_i)\|_{l_1}) \tag{9.54}$$

Notice that in this case the value 0.6745 has been omitted as it does not improve the performance of the filter. Another similar approach that we will use for comparison purposes is a simple average filter defined as [157, 210]:

$$X_i = X_i * X_i / X_{\max}, \tag{9.55}$$

where $X_{\max}$ is the element of vector X with the maximum value among all its elements.

An alternative and very popular technique mainly used in sparse recovery methods under the presence of noise is the hard and soft thresholding [86, 132, 186]. In hard thresholding we pick coefficients with magnitudes greater than a specific threshold, while in soft thresholding we subtract the coefficient from the threshold if it is greater than a threshold. In particular the hard thresholding technique works as follows:

$$X_i = \begin{cases} X_i & \text{if } |X_i| \geq \text{threshold} \\ 0 & \text{otherwise} \end{cases} \tag{9.56}$$

On the other hand the soft thresholding technique works as follows:

$$X_i = \begin{cases} \text{sign}(X_i)(X_i - \text{threshold}) & \text{if } |X_i| \geq \text{threshold} \\ 0 & \text{otherwise} \end{cases} \tag{9.57}$$

In the experiments, for comparison purposes, we will use a soft thresholding technique where we threshold (keep) significant coefficients (X_i) with threshold value 0.1, which is the maximum value of the additive noise used in the experiment. Similar approaches suggested in [132, 186] is to set threshold value as $r\sigma^2$ or even $\sigma\sqrt{2\log M}$, where $r \in [1, 3]$ a small number based on the variance of the additive noise of the sampling model σ and M the number of (noisy) measurements collected. The latter is a commonly used approach for uncorrelated additive noise, under an over-complete transform domain (i.e. Unitary transform) as a representation, introduced by Donoho and Johnstone. However, this latter value has been argued that it is too large as a threshold in most numerical experiments (which is also the case here) and thus several dictionary adapted threshold estimators have been introduced [132].

Figure (9.24) presents sparse recovery of a real-valued sparse vector following power decay law, under different noisy sample sizes $M = 1, 2, \ldots, 180$, using different types of filters; no-filter, moving average, mean absolute deviation and soft-thresholding. The measurements collection model used is $Y_{M\times 1} = C_{M\times N}X_{N\times 1} + \epsilon_{M\times 1}$, where Y is the randomly collected noisy measurements vector, C is the Sensing matrix, randomly generated using the Normal distribution with values between zero and one, while X is a sparse vector with real values between zero and one based on the power decay model $X = \exp(-tk)$, where $k = 0.84$ is a small parameter value affecting sparsity and $t = 0, 1, \ldots, 99$ is the time parameter defining the length of the vector $N = 100$. Notice that the maximum number of samples collected is 180, which define an over-determined case of sampling. Also the ϵ vector depicts the randomly generated additive noise in the interval $[0.01, 0.1]$ defined as $\epsilon = 0.01+(0.1-0.01)r$, with $0 \leq r \leq 1$ a small random number using the Normal distribution. The l_0 heuristic's parameters are $T = 50$ iterations, $S = 12$ swarms, σ decrease factor $\beta = 0.5$ and initial $\sigma^{(0)} = 2\|X\|_{l_\infty}$. The choice of small number of iterations has been chosen so as to efficiently compare different types of filters after the use of the l_0 heuristic and how these filters can improve the heuristic's performance. Towards the same objective a sparsity threshold thres $= 10^{-5}$ is used for the calculation of the sparsity level of X, namely $K = \text{length}(\text{find}\|X\|_{l_0} > \text{thres}))$. The relative recovery error was calculated using SNR defined in Equation (9.11), as a measure of the quality of the estimate $\hat{X}$ under noisy measurements.

Notice in Figure (9.24) that for sample sizes less than 110 the moving average and soft-thresholding filters after the use of the l_0 heuristic improve the estimate from the noisy measurements. In particular, the moving average achieves the best results with SNR between

5 and 10, while soft-thresholding filter achieves slightly worse results with SNR between 3 and 5. On the other hand, in over-determined cases of more than 110 samples, the use of any filter degrades the quality of the estimate, since the application of the l_0 heuristic without filters achieves three times better SNR ratio with values between 35 and 40. Also notice the SNR ratio of the 50-th and 110-th sample where the l_0 heuristic performs two to three times better than any filter, which can be attributed to the stochastic nature of the l_0 heuristic in the generation of the initial and new solutions.

In general, there is a tradeoff between increased compression and increased noise. In particular, for highly under-determined samples, even small bounded additive noise can cause a lot of distortion in the estimate as very few coefficients (elements) from the original vector are stored, fact which causes loss of peaks. For over-determined samples small bounded additive noise does not affect efficient recovery. This is expected as in CS we acquire only a small number of measurements in contrast to over-determined cases where we measure the vector entirely. In fact, it has been argued in [39] that the measurement matrix has to be adjusted so as to satisfy the measurement error bound as a tighter assumption in its design for robust noisy measurements. Also in [215] the authors argue that the pair (Δ, C) is ϵ-stable if $1/\epsilon\|X\|_{l_2} \leq \|CX\|_{l_2}$ for a small constant ϵ, which represents the quantisation noise acquired in some measurement interval. Notice that as $\epsilon \to 1$ then the Sensing matrix C must satisfy the lower bound with $\delta_K = 1 - 1/\epsilon^2 \to 0$. Therefore, in order to reduce the impact of noise in the recovered estimate of a vector we need adjust C so as it satisfies the lower bound of RIP with a tighter constant. However, this is not always easy in practice since in real noisy cases the noise is random and usually dependent of C. Indeed, the use of a filter for removal of small levels of noise, which does not require this prior-knowledge, together with the use of l_0 heuristic, which does not require RIP/UUP properties, seems an important improvement in sparse recovery using the l_0 norm based principle.

9.5 Experimental results in images

9.5.1 Experiments description

In this Section we measure the performance of the l_0 heuristic as a sparse image recovery method in three 128×128 synthetic/test images (Phantom, Circle and Checkerboard), which have been extensively used for testing purposes in image recovery techniques (see [7, 45, 100,

Figure 9.24: Sparse recovery of a real-valued sparse vector of 100 elements with values between zero and one, under different noisy sample sizes $M = 1, 2, \ldots, 180$. the measurement matrix C has been created randomly using the Normal distribution with element values between zero and one. We also assume a small additive noise, in the interval $[0.01, 0.1]$, in the measurements collection model which depicts impressions or mistakes in the sampling process. The l_0 heuristic's parameters are $T = 50$ iterations, $S = 12$ swarms, σ decrease factor $\beta = 0.5$, initial $\sigma^{(0)} = 2\max(X)$ and sparsity threshold thres $= 10^{-5}$ used for the calculation of the sparsity level of the initial real-valued sparse vector $K = \text{length}(\text{find}|X|_{l_0} > \text{thres}))$. The relative recovery error was calculated using SNR (experiment10.m).

199]). The selection of these images with good quality of curved edge-like structure was made mainly due to the fact that they exhibit a similar mix of smooth but different in appearance regions as an efficient way to test competing sparse recovery methods under both noiseless and noisy sampling environments (measurements). Here, we also adopt input images which are complex valued, rather than being purely real, as a more realistic case (e.g. by using a unitary transform to enhance sparsity and collect the measurements). Note that the samples collected from the three images are used as the same input data for all methods, in order to recover the original image in spacial domain.

The use of the complex-valued domain requires some amendments in the l_0 heuristic so as to handle complex, phase-encoded data. Like [42, 75, 86] we will consider the recovery of a complex-valued image f (in frequency domain) by recovering its real and imaginary parts separately. To achieve this, we also assume that the real and imaginary parts of f are both sparsely represented in the frequency (complex-valued) domain (i.e. both real and imaginary parts share the same sparsity pattern). Keeping and arranging these real and imaginary parts separately into column vectors we effectively have a purely real over-complete transform, which can then be applied simultaneously to the real and imaginary parts of the image f [168, 179]. This approach is also common in TV minimisation [16, 38, 43, 170], defined in Section 3.7, though sometimes this process leads to unphysical blocking artifacts, which limits the quality of the sparse recovery method [168].

The sampling process of a complex-valued image f can now be modelled as [168, 170]:

$$Y_{M\times N} = Fu_{M\times N} f_{N\times N} + e_{M\times N}, \tag{9.58}$$

where Y represents the (complex-valued) measurements collected from the k-sparse representation of the complex-valued image f using the under-sampled Fourier transform Fu. In essence, $Fu = \Phi\Psi$, where Φ is the linear operator (sampling matrix) which maps the image f to a set of M measurements and Ψ is the sparsifying matrix (orthonormal basis) representing the Unitary transform, FFT in this case. The term $0.01 < e < 0.1$ represents small additive noise as a sampling imprecision, which is generated following the normal distribution and is based on the samples Y size. This process of collecting sub-sampled measurements in frequency domain simulates the samples acquisition technique for MRI images (sub-sampled Fourier domain measurements) or more generally the distribution of velocity of liquid or

gas flows, used in the classification and prediction of flow behaviour [169, 179]. If we now decompose image f into its real (Real(f)) and imaginary (Imag(f)) components the sparse measurement model becomes:

$$Y_{M\times N} = Fu_{M\times N}\{\text{Real}(f) + i\text{Imag}(f)\}_{N\times N} + e_{M\times N}, \tag{9.59}$$

where $f = \text{Real}(f) + i\text{Imag}(f)$. Note that we still consider incoherent measurements in images where the domain of representation is the Fourier transform (FFT) for enhancing sparsity. Also the number of measurements for efficient recovery highly depends on the structural content of the image (number of significant non-zero entries in the frequency domain) than its resolution [42, 43]. Under these assumptions, sparse recovery can be achieved as a minimisation problem of the absolute value of the sum of magnitudes of the image f as a search for the sparsest image representation in the domain Ψ which fits the measurements Y. The choice of the minimisation problem and the corresponding norm in the domain of image representation (Fourier domain) depends on the choice of the sparse image recovery package used (see Chapter 8 for further details).

In the following two experiments we will compare the performance of the l_0 heuristic in terms of recovery error and time complexity against l_1-Magic, TWIST and FOCUSS-cndl packages. CPU cycles have been used as a rough estimate of the execution time. The relative recovery error metric (MSE) defined in (9.6) and the Peak Signal-to-Noise ratio (PSNR) defined in (9.12) have been used for the noiseless and noisy collected measurements respectively. The parameters of the sparse recovery methods are as follows (the same for both noisy and noiseless measurements). For l_1-Magic parameters are $lbtol = 10^{-1}$ as the duality gap, $\mu = 2$ as the factor to increase the barrier constant at each iteration, $slqtol = 10^{-8}$ as a tolerance level for the iterations, $slqmaxiter = 600$ as the number of log barrier iterations and the back-projection $\hat{f} = Fu^TY$ as the method's starting feasible point. The FOCUSS-cdl parameters are $t = -1$ as a parameter to allow normal convergence, ps="true" to indicate that the samples are positive, $\lambda = 2.0e-3$ as a regularisation parameter limit, $iter = 15$ as the maximum number of iterations, $p = 1$ for the norm $d_p(.)$ used and $\hat{f} = Fu^TY$ as the method's starting feasible point. The TwIST parameters are $\lambda = 10^{-2}$ as the regularisation parameter of the TV based objective function and $\hat{f} = Fu^TY$ as the method's starting feasible point. The l_0 heuristic's parameters are $S = 10$ swarms, σ decrease factor $\beta = 0.5$

and initial $\sigma^{(0)} = 2\max(f)$. The number of iterations T and the sparsity threshold thres, used for the calculation of the sparsity level of the initial image $K = \text{length}(\text{find}\|f\|_{l_0} > \text{thres}))$, are calculated based on the number of measurements collected, expressed in radial lines LL. The desired number of iterations and the sparsity threshold are defined respectively as $T = 750$ and thres $= 10^{-2}$ for $LL < 30$, while for $LL > 30$, $T = 1000$ and thres $= 10^{-3}$. The number of radial lines LL chosen for the experiments are $LL = 8, 12, 28, 42, 82$ which represent $1001, 1489, 3353, 4869, 8657$ samples in the FFT domain respectively.

Note that due to the very large scale of the problem, a matrix handle to a function that computes the products of the form Fuf and Fu^TY has been implemented in Matlab $R2014b$. This approach does not directly affect the calculations of the very large under-sampled dictionary Fu, but this avoids its direct processing which will cause "out of memory" error [73]. In fact, direct calculation of $(Fu^TFu)^{-1}$ is not possible and thus the regularised pseudo-inverse defined in (6.41) of Chapter 6 cannot be achieved. Therefore, an efficient variation, namely $Fu^TY + R_iFu^T(FuFu^TY - Y)$, has been chosen for the generation of the initial solution, where R_i is a vector of numbers, different for every swarm and randomly generated using Gaussian distribution between 0 and 1.

Finally, a median filter is used as an optional post-processing step after the use of the l_0 heuristic both for noisy and noiseless measurements. The median filter is very similar to a moving average filter and can be used to remove small additive (spike) noise from an image or background. However, in this case a median is calculated, instead of a mean, by sorting all the values of the reconstructed image from lower to higher levels of intensity, and then taking the value in the centre of the image. In general, a median filter considers each pixel in the image in turn and calculates its value based on its neighbouring pixels values (replace it with the median of those values). This is achieved by firstly sorting all pixel values surrounding the neighbourhood (window) into numerical order and then replacing the pixel being considered with the middle pixel value. If two values are in the middle, their average is calculated. Median filter is more robust than average filter since unrepresentative/extreme pixel values do not affect the median value significantly. It is also efficient on preserving sharp edges than mean filter as it does not create unrealistic values in edges for an image. In Matlab, a sorting algorithm is initially used for sorting image's pixels and then the median of each pixel is calculated using its 5 pixel neighbours by matrix convolution (the medfilt2() function in Matlab uses ordfilt2() function to perform the filtering; see Matlab online documentation

and Appendix B for more details on convolution).

9.5.2 Experiments without noise

In this experiment we compare the performance of the l_0 heuristic against other competing sparse image recovery methods, namely l_1-Magic, TWIST and FOCUSS. For this purpose three synthetic test images, Phantom, Circles and Checkerboard, of 128×128 dimensions have been used under different measurement sizes, expressed in radial lines as a sampling pattern in the frequency (FFT) domain. Table (9.3) compares the performance of these sparse image recovery methods in terms of execution time t, measured in CPU cycles, and recovery error, to evaluate the recovered image quality, measured as relative MSE defined in Equation (9.10). It is important to pinpoint again that we treat $N \times N$ images as $N : N^2$ vectors for simplicity and efficiency in the calculations. Another important note is that since the underlying images are real, the sampling pattern presented here returns the real and imaginary part of the $2D$ FFT on the upper half plane of the (Fourier) frequency domain, as shown for example in Figures (9.25) and (9.26). All the solvers used in recovery methods are iterative, where each iteration requiring one application of the partial Fourier matrix Fu. Moreover, based on the structure of the given test images, a few frequency coefficients (magnitudes) are enough to capture most of the image energy, as most of such types of images are highly compressible. Following this aspect, the l_0 heuristic's performance (in accordance with all the other sparse recovery methods) is increasing as the number of measurements increases and deteriorates as the measurements decrease, which is expected as fewer measurements result in loss of image quality and thus loss of substantial information (i.e. aliasing in the reconstruction).

In general, we can see that the l_0 heuristic is found to have significantly better performance ($e \in [0.23, 0.28]$) with smaller run times than l_1-Magic for highly under-sampled measurements (less than 12 radial lines) for the Phantom and Checkerboard images. However, in all cases TWIST is the quickest sparse recovery method but without good quality solutions. As the number of measurements increases the quality of sparse recovery of the l_0 heuristic increases but in such cases l_1-Magic outperforms all the other methods (since UUP/RIP conditions hold). Notice two special cases here; in the binary circles image where the l_0 heuristic performs slightly worse than l_1-Magic and in the Checkerboard image sampled using 82 radial lines, where the l_0 heuristic outperforms all the other recovery methods.

In fact in this case the difference between the recovered and the original image is hardly noticeable. On the other hand, the application of a median filter does not seem to significantly improve the quality of the recovered image, in contrast to the next experiment of noisy measurements, where the median filter efficiently removes small additive noise and improves the image quality.

The results of sparse recovery of Phantom and Checkerboard images for 12 radial lines are presented in Figures (9.25) and (9.26). Starting from left to right and from top to bottom, the original image is presented, the sampling pattern (i.e. measured locations or number of lines through origin in $2D$ Fourier plane), the Back-projection (Fu^TY), which represents direct recovery from partial measurements, and the recovered image (estimate) using the methods l_1-Magic, FOCUSS, TWIST and l_0 heuristic, with and without the application of a median filter, respectively. The l_0 heuristic achieves not only smaller recovery error but also manages to recover the geometry of the image, including its boundaries and its separation from the background. This is possible due to the pixel based recovery adopted by the l_0 heuristic in contrast to region based recovery used in TV-norm based methods TWIST and l_1-Magic. The l_0 heuristic manages to converge smoothly towards good quality sparse solutions, despite the non-convexity of the problem space (using a function to approximate the l_0 norm). This efficiency can be attributed to the weighted approach of backward projection step together with the iterative hard-thresholding operator P_K which encourage sparsity of the solution. Also note that the l_0 heuristic has the second highest execution time after the l_1-Magic method.

The l_0 heuristic is an iterative process which keeps improving the current best solution through iterations, provided that there is a sufficient number of measurements (i.e. we do not have highly under-determined cases, this aspect is explained in detail in Section 9.4.2). Therefore, if we provide a smaller number of iterations, say $T = 50$ instead of $T = 750$, we will have very small execution time at the expense of lower quality of the final solution. In general, there is a balance between the number of measurements, sparsity level of the object sampled and the required number of iterations for good quality of solutions. The "ideal" number of required iterations can only be found experimentally, but in general the smaller the sparsity and the larger the measurements, more iterations can improve significantly the final (optimal) solution.

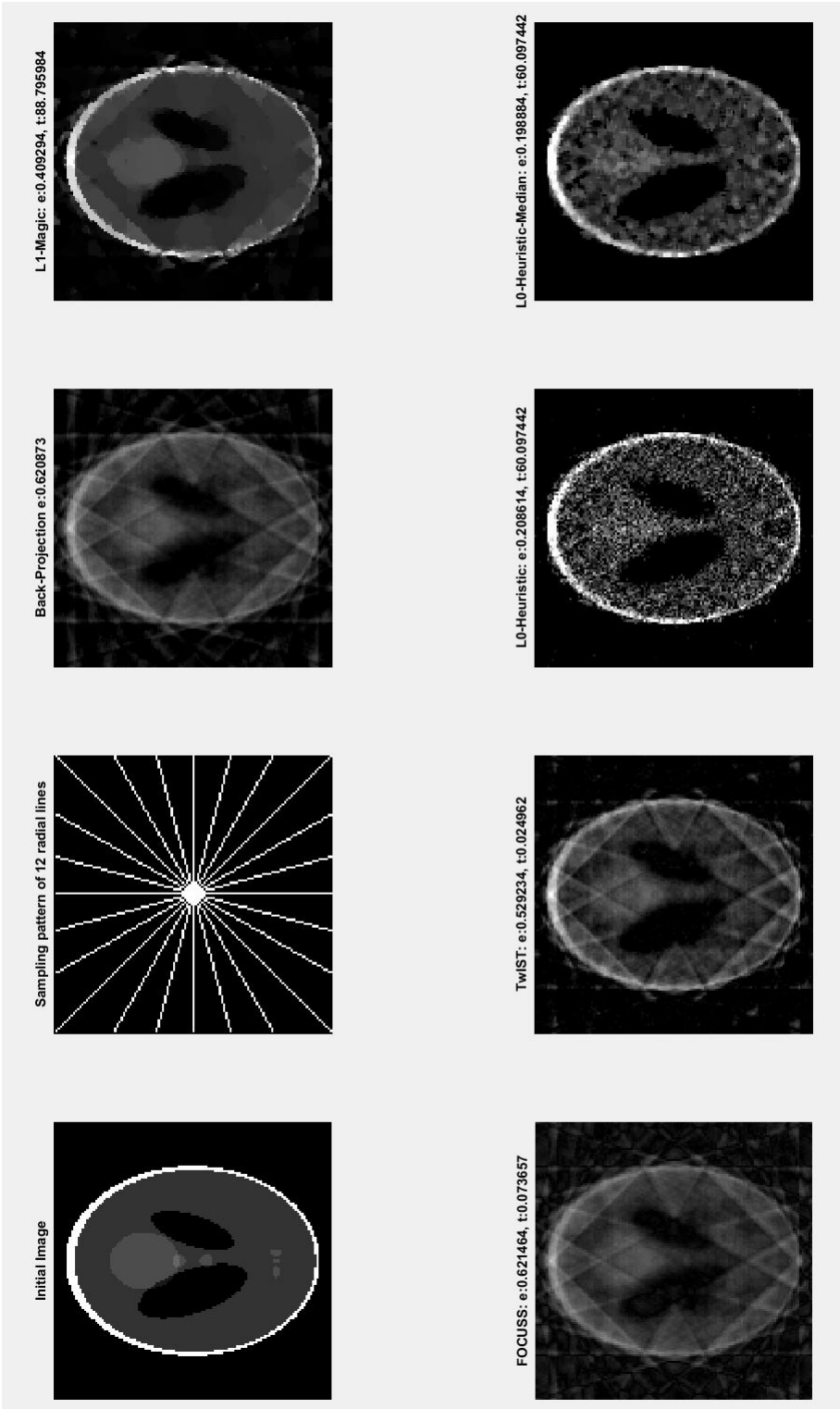

Figure 9.25: Sparse recovery of the Phantom image from 12 radial lines without noise. The sparse recovery methods compared are l_0 heuristic, l_1-Magic, TWIST and FOCUSS. The relative MSE has been used as an error metric e, while CPU cycles have been used as a metric for the execution time t of each method. Back-Projection ($F_u^T Y$) is used for comparison as a direct recovery from the highly under-sampled measurements (experiment11.m).

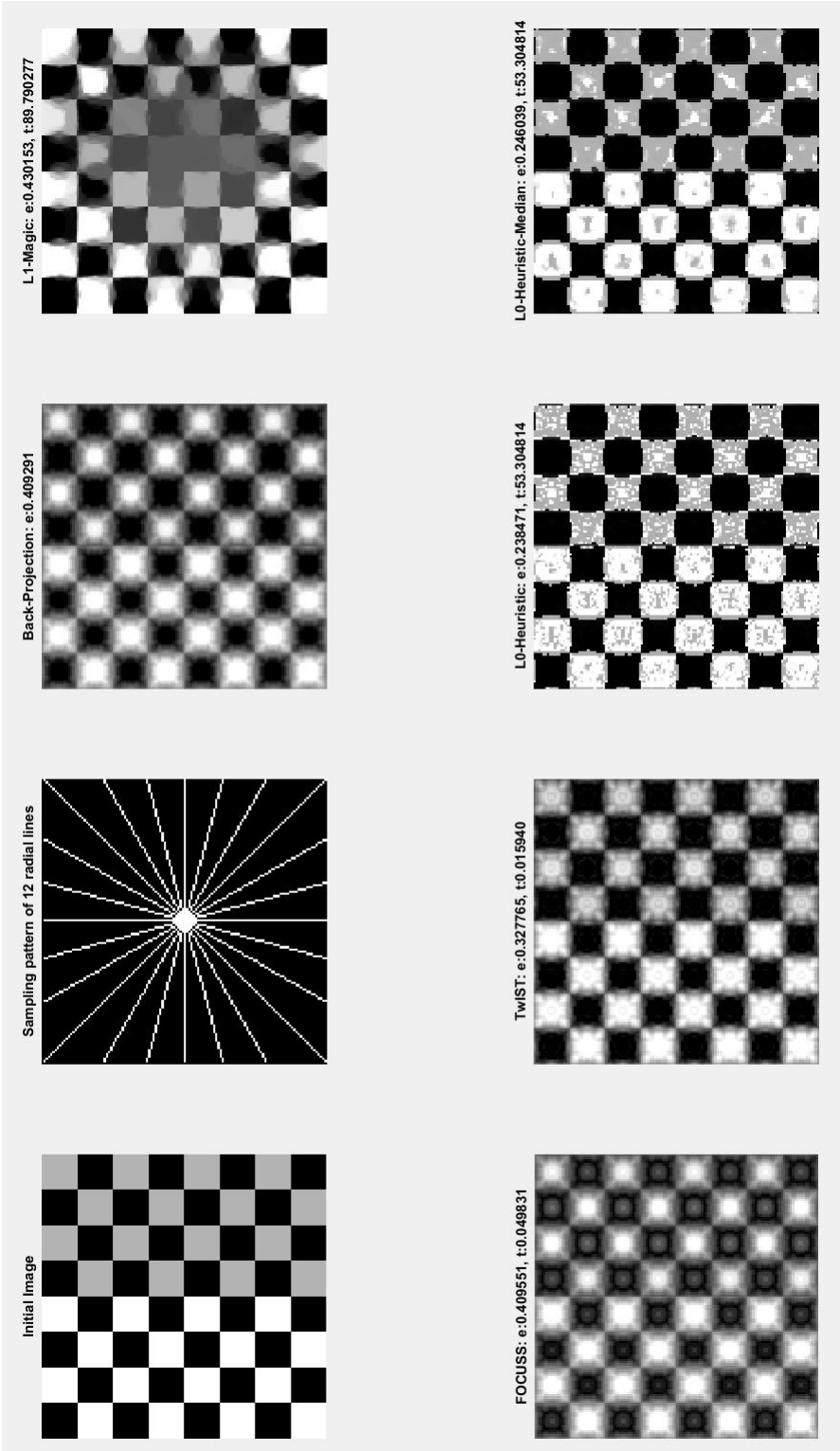

Figure 9.26: Sparse recovery of the Checkerboard image from 12 radial lines without noise. The sparse recovery methods compared are l_0 heuristic, l_1-Magic, TWIST and FOCUSS. The relative MSE has been used as an error metric e, while CPU cycles have been used as a metric for the execution time t of each method. Back-Projection (Fu^TY) is used for comparison as a direct recovery from the highly under-sampled measurements (experiment11.m).

9.5.3 Experiments with noise

In this experiment we compare the performance of the l_0 heuristic against other competing sparse image recovery methods, namely l_1-Magic, TWIST and FOCUSS. For this purpose three synthetic test images, Phantom, Circles and Checkerboard, of 128×128 dimensions have been used under different noisy measurements, expressed in radial lines as a sampling pattern in the frequency (FFT) domain. This simulation follows the data acquisition and noise levels pattern commonly followed by many real imaging devices, which collect measurements along radial lines at relatively few angles, such as MRI and PET images [169, 170]. As we discussed in Section 5.2, image noise represents a random variation in brightness or color information in images, which is not present in the original image. It is an undesirable by-product of image capture that adds extra information and thus changes the light intensity of the image.

In this experiment we will add a small randomly generated number $0.01 < \|e\|_{l_2} < 0.1$ during the sampling process of the test images as a realistic simulation of sampling (hardware) imperfections. Note that in grey-level images, darker regions are less prone to noise than brighter regions by convention (in academic papers). The light intensity of a pixel is expressed within a given range of values, with the smallest value always representing black colour and the highest value the white colour. All the fractional values in between represent the tones of grey. Therefore, brighter regions have a stronger signal (i.e. higher values) due to more light, resulting in a higher overall PSNR. It is also expected that small additive noise will make the images increasingly smooth at similarly low (i.e. smoother) level areas. However, based on the structure of the synthetic test images, which are highly compressible, the effects of the small additive noise on images are expected to be eliminated as the number of measurements increases, for all sparse image recovery methods.

Table (9.4) compares the performance of the sparse image recovery methods in terms of execution time t, measured in CPU cycles, and recovery error e, to evaluate the recovered image quality, measured as PSNR defined in Equation (9.12). In general, we can see that the l_0 heuristic is found to have slightly better performance ($e \in [17, 27]$) with smaller run times than l_1-Magic for most of the radial lines in Phantom and Circles images. However, in almost all cases TWIST is the quickest sparse recovery method but with some good quality solutions only in the sparse recovery of the Checkerboard image. In the recovery of Checker-

board image, the l_0 heuristic achieves slightly worse or equally the same quality solutions as l_1-Magic. In general, as the number of measurements increases all sparse recovery methods achieve better results (their PSNR increases), though the performance of the l_1-Magic does not increase significantly for more than 28 radial lines when the necessary UUP/RIP conditions for noisy cases hold. Notice the case of sparse recovery of Circles and Checkerboard images from 82 radial lines, where the l_0 heuristic with a median filter over-performs all the other sparse recovery methods. In fact, from Figures (9.28) and (9.27) the difference between the recovered and the original image is hardly noticeable. In essence, the application of a median filter seems to significantly improve the quality of the recovered image and its corresponding PSNR metric, in contrast to the previous experiment of sparse recovery from noiseless measurements.

Figures (9.28) and (9.27) illustrate the sparse recovery of Circles and Checkerboard image respectively, from $M = 8657$ samples (82 radial lines). This is the sampling domain in the frequency plane which represents how Fourier coefficients are sampled long the radial lines. In each experiment we observe 82×256 noisy real-valued Fourier coefficients and use different methods for sparse recovery (about 87% of the 2D Fourier coefficients are omitted). Starting from left to right and from top to bottom, we have the original image, the sampling pattern (i.e. measured locations or number of lines through origin in $2D$ Fourier plane), the Back-projection (Fu^TY), as a minimum energy recovery obtained by setting unobserved Fourier coefficients to zero, and the recovered image (estimate) using the methods l_1-Magic, FOCUSS, TWIST and l_0 heuristic, with and without the application of a median filter, respectively. The l_0 heuristic achieves not only smaller recovery error (higher PSNR) but also manages to recover the image boundaries/edges, separate it from the background and recover some smaller details within the image, such as the smaller darker circles in Circles image. This is possible due to the pixel based recovery adopted by the l_0 heuristic in contrast to region based recovery used in TV-norm based methods TWIST and l_1-Magic. As expected in this case we do not have any loss of contrast, apart from the four corners of the images again, which can be attributed to the nature of the sampling pattern expressed in radial lines. Possible ways to correct this bias in the sampling process could be achieved by applying a low pass filter (a threshold a-posterior to the coefficients so as to decrease the number of non-zeros of the solution), aspect which can be an extension of a post-processing step of the l_0 heuristic. Notice again that the l_0 heuristic is an iterative process and its execution time

depends on the desired number of iterations. Higher number of iterations achieve better estimates but increase the execution time, while smaller execution time implies lower quality of the estimate. This is the reason that the l_0 heuristic has the second highest execution time after the l_1-Magic method.

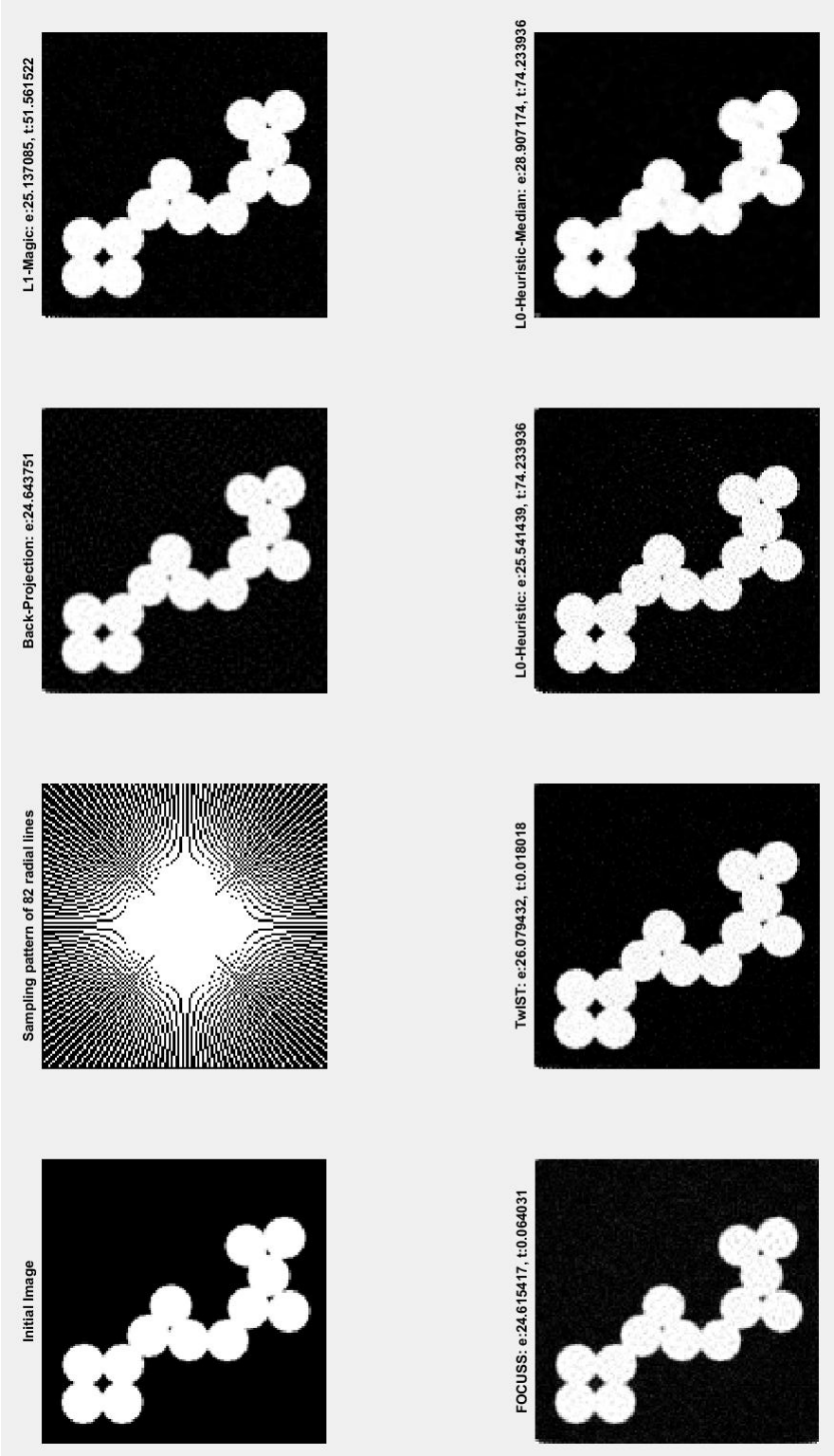

Figure 9.27: Sparse recovery of the Circles image from 82 radial lines with small additive random noise. The sparse recovery methods compared are l_0 heuristic, l_1-Magic, TWIST and FOCUSS. The PSNR has been used as an error metric e, while CPU cycles have been used as a metric for the execution time t of each method. Back-Projection ($F_u^T Y$) is used for comparison as a direct recovery from the highly under-sampled noisy measurements. Note that Circles is a binary image with only two values, 0 and 1 (experiment11.m).

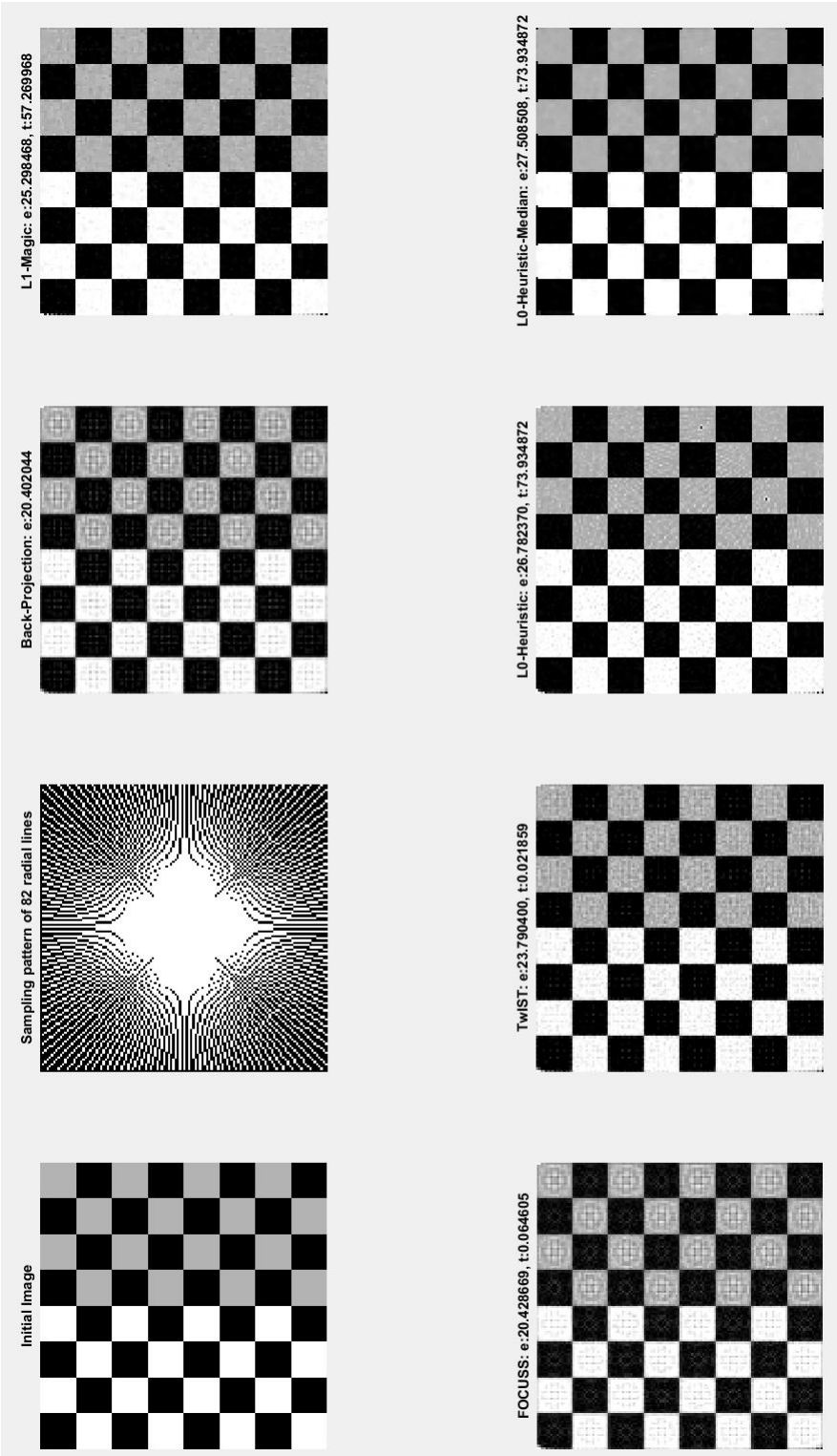

Figure 9.28: Sparse recovery of the Checkerboard image from 82 radial lines with small additive random noise. The sparse recovery methods compared are l_0 heuristic, l_1-Magic, TWIST and FOCUSS. The PSNR has been used as an error metric e, while CPU cycles have been used as a metric for the execution time t of each method. Back-Projection (Fu^TY) is used for comparison as a direct recovery from the noisy under-sampled measurements (experiment11.m).

Table 9.3: Comparison Table of sparse recovery of three images; Phantom (1), Circles (2) and Checkerboard (3). The sparse recovery methods compared are l_0 heuristic, l_1-Magic, TWIST and FOCUSS. The relative MSE has been used as an error metric e, while CPU cycles have been used as a metric for the execution time t of each method. The radial lines express the number of noiseless measurements collected, while values in bold represent the "best" execution time and recovery error (experiment11.m).

	Lines	Back-Proj.	l_0-Heur.	l_0-Heur.(Med)	l_1-Magic	FOCUSS	TWIST
(1.)	8	e : 0.66908	**e : 0.27315**	e : 0.27603	e : 0.59697	e : 0.66966	e : 0.58904
			t : 60.5003	t : 60.5003	t : 92.9086	t : 0.06909	**t : 0.04660**
	12	e : 0.62087	e : 0.20861	**e : 0.19888**	e : 0.40929	e : 0.62146	e : 0.52923
			t : 60.0974	t : 60.0974	t : 88.7959	t : 0.07365	**t : 0.02496**
	28	e : 0.48910	e : 0.18628	e : 0.16109	**e : 0.00283**	e : 0.48977	e : 0.32078
			t : 56.8526	t : 56.8526	t : 48.9117	t : 0.06782	**t : 0.02425**
	42	e : 0.40316	e : 0.11973	e : 0.09869	**e : 0.00015**	e : 0.40368	e : 0.25174
			t : 75.0842	t : 75.0842	t : 36.1695	t : 0.07441	**t : 0.02656**
	82	e : 0.25048	e : 0.09584	e : 0.09922	**e : 0.00001**	e : 0.25076	e : 0.12586
			t : 74.7238	t : 74.7238	t : 19.7147	t : 0.08505	**t : 0.02274**
(2.)	8	e : 0.40908	e : 0.09144	e : 0.06563	**e : 0.06463**	e : 0.40945	e : 0.25467
			t : 61.1168	t : 61.1168	t : 90.1312	t : 0.06822	**t : 0.02244**
	12	e : 0.32751	e : 0.04971	e : 0.05146	**e : 0.03904**	e : 0.32769	e : 0.18723
			t : 61.5674	t : 61.5674	t : 75.7745	t : 0.05742	**t : 0.02329**
	28	e : 0.20481	e : 0.01891	e : 0.03721	**e : 0.01157**	e : 0.20491	e : 0.10804
			t : 61.3615	t : 61.3615	t : 50.7954	t : 0.06985	**t : 0.05586**
	42	e : 0.16961	e : 0.01196	e : 0.03822	**e : 0.00359**	e : 0.16977	e : 0.07388
			t : 86.6883	t : 83.6883	t : 48.3419	t : 0.08883	**t : 0.04076**
	82	e : 0.08798	e : 0.00340	e : 0.04027	**e : 0.00013**	e : 0.08889	e : 0.03206
			t : 85.2393	t : 85.2393	t : 27.2485	t : 0.09242	**t : 0.02776**
(3.)	8	e : 0.41492	e : 0.35965	**e : 0.28742**	e : 0.63083	e : 0.41515	e : 0.37162
			t : 52.6925	t : 52.6925	t : 99.2223	**t : 0.05081**	t : 0.05366
	12	e : 0.40929	**e : 0.23847**	e : 0.24603	e : 0.43015	e : 0.40955	e : 0.32776
			t : 53.3048	t : 53.3048	t : 89.7902	t : 0.04983	**t : 0.01594**
	28	e : 0.27782	e : 0.15561	e : 0.18063	**e : 0.00026**	e : 0.27805	e : 0.22183
			t : 53.3320	t : 53.3320	t : 31.9543	t : 0.05187	**t : 0.01507**
	42	e : 0.24459	e : 0.12679	e : 0.17522	**e : 0.00002**	e : 0.24475	e : 0.17795
			t : 72.1827	t : 72.1827	t : 31.8566	t : 0.05317	**t : 0.01820**
	82	e : 0.14084	**e : 0.00000**	e : 0.05483	e : 0.00005	e : 0.14093	e : 0.07243
			t : 72.2845	t : 72.2845	t : 14.7561	t : 0.05821	**t : 0.02005**

Table 9.4: Comparison Table of sparse recovery of three images; Phantom (1), Circles (2) and Checkerboard (3). The sparse recovery methods compared are l_0 heuristic, l_1-Magic, TWIST and FOCUSS-cndl. The PSNR has been used as an error metric e, while CPU cycles have been used as a metric for the execution time t of each method. The radial lines express the number of noisy measurements collected, while values in bold represent the "best" execution time and recovery error (experiment11.m).

	Lines	Back-Proj.	l_0-Heur.	l_0-Heur.(Med)	l_1-Magic	FOCUSS	TWIST
(1.)	8	e : 15.6730	e : 15.5735	**e : 17.1556**	e : 16.0914	e : 15.6674	e : 16.7351
			t : 56.7291	t : 56.7291	t : 96.7619	t : 0.06845	**t : 0.02456**
	12	e : 16.2972	e : 17.7821	**e : 18.7612**	e : 18.3644	e : 16.2916	e : 17.59673
			t : 55.8783	t : 55.8783	t : 96.1028	t : 0.07023	**t : 0.06222**
	28	e : 18.2256	e : 18.9575	e : 21.6595	**e : 24.8080**	e : 18.2257	e : 21.2552
			t : 62.1647	t : 62.1647	t : 81.1179	**t : 0.07097**	t : 0.07105
	42	e : 19.6777	e : 18.7847	e : 23.0448	**e : 25.2234**	e : 19.6848	e : 22.6748
			t : 84.5506	t : 84.5506	t : 77.3003	t : 0.07368	**t : 0.06918**
	82	e : 22.5738	e : 20.6052	**e : 25.5548**	e : 25.2163	e : 22.6105	e : 24.9581
			t : 82.2985	t : 82.2985	t : 54.7971	t : 0.08586	**t : 0.02148**
(2.)	8	e : 14.4853	e : 21.4546	e : 22.5962	**e : 23.6490**	e : 14.4793	e : 18.4721
			t : 52.3849	t : 52.3849	t : 89.8635	**t : 0.05034**	t : 0.05609
	12	e : 16.3841	e : 22.7591	**e : 25.0912**	e : 24.4050	e : 16.3815	e : 20.9539
			t : 60.4316	t : 60.4316	t : 87.5857	t : 0.05623	**t : 0.02527**
	28	e : 20.2018	e : 23.8147	**e : 26.8808**	e : 25.3689	e : 20.2019	e : 24.2274
			t : 59.9151	t : 59.9151	t : 65.6527	t : 0.06328	**t : 0.01877**
	42	e : 21.5017	e : 24.9077	**e : 27.5660**	e : 25.3416	e : 21.5012	e : 25.2558
			t : 85.1067	t : 85.1067	t : 72.4995	t : 0.06799	**t : 0.01826**
	82	e : 24.6437	e : 25.5414	**e : 28.9071**	e : 25.1371	e : 24.6154	e : 26.0794
			t : 74.2339	t : 74.2339	t : 51.5615	t : 0.06403	**t : 0.01801**
(3.)	8	e : 11.9150	e : 9.62794	e : 12.2704	e : 8.32022	e : 11.9110	**e : 13.6331**
			t : 54.2372	t : 54.2372	t : 102.669	t : 0.05843	**t : 0.04863**
	12	e : 12.0262	e : 10.0646	e : 12.7152	e : 11.5270	e : 12.0216	**e : 13.9332**
			t : 54.7588	t : 54.7588	t : 95.5804	t : 0.05726	**t : 0.01458**
	28	e : 15.3132	e : 23.3275	e : 24.2670	**e : 25.5979**	e : 15.3088	e : 17.1630
			t : 54.9187	t : 54.9187	t : 80.6681	t : 0.05884	**t : 0.01664**
	42	e : 16.3322	e : 21.0658	e : 23.9796	**e : 25.1355**	e : 16.3298	e : 19.7025
			t : 73.6521	t : 73.6521	t : 73.7737	t : 0.05395	**t : 0.02030**
	82	e : 20.4020	e : 26.7823	**e : 27.5085**	e : 25.2984	e : 20.4286	e : 23.7904
			t : 73.9348	t : 73.9348	t : 57.2699	t : 0.06460	**t : 0.02185**

Chapter 10

Concluding remarks and future research

The aim of this thesis was two-folded; to survey and provide the mathematical foundations of a novel research field, called Compressed Sensing (CS) method, and also to introduce a new swarm based heuristic with a direct application into this field. CS has recently attracted considerable research, with several new application areas [100], as a framework for simultaneously sampling and compressing signals and images where the standard sampling process is not feasible or very expensive. This new sampling scheme does not follow the principle of conventional approach depicted by the sampling theorem of Nyquist-Shannon. The goal is to efficiently recover any type of signal, such as speech and image data, using what was previously considered as highly incomplete and inaccurate (under-sampled) measurements. This is an ill-posed inverse problem, which can be solved as an l_0-norm based (non-linear) optimization problem (NP-hard problem), with the aim to find the best fit which minimises the difference between the solution and the observations while satisfying all the given constraints. Several sparse recovery methods, software packages and optimisation principles have been introduced for solving this problem, which were also defined and presented in this thesis. In this thesis a novel swarm based heuristic for sparse recovery is also introduced for efficiently recovering signals and images with high probability. It is an iterative process which finds an approximation of the l_0-norm based problem, viewed as a combinatorial optimization problem, by iteratively re-weighting and minimising the difference between the solution and the

observations. In each iteration every swarm calculates and carries a slightly different feasible solution based on the current best (optimal) solution, which is necessary so as to avoid being trapped to one of the numerous local minima. In general, this l_0 heuristic achieves fast and efficient sparse recovery, compared to other conventional recovery approaches, particularly for severally under-sampled and over-sampled cases even under the presence of (small) noise, based on experimental results discussed.

Chapter 9 presented a number of experiments where the l_0 heuristic is compared with other sparse recovery methods, such as OMP, COSAMP, AIHT, LASSO, SL0, IRLS and software packages, such as L1-Magic, NESTA, CVX, FOCUSS, TWIST, in sparse signal and image recovery. The performance of each sparse recovery method is measured in terms of execution time (in CPU cycles) and recovery error (e.g. expressed as MSE, SNR and PSNR) under different sparsity levels (from highly sparse to weakly sparse vectors), samples sizes (from highly under-sampled to over-sampled cases), types of Sensing matrices (Uniform, zero-one, Normal, exponential, geometric, Poisson, Pareto), Unitary transforms (FWHT,DFT,DCT), different levels of noise during the sampling process and different synthetic test images. The l_0 heuristic is able to recover real and complex-valued sparse vectors under linear time complexity, in contrast to other recovery methods, such as LASSO, l_1 minimisation principle, NESTA and l_1-Magic, which require quadratic time complexity.

The l_0 heuristic is particularly efficient in recovering sparse vectors with small recovery error from severely under-sampled and over-sampled measurements, on condition that the vector is sparse enough. For example, for an 70-element real-valued vector with 10 non-zero entries, efficient recovery (10^{-1}) can be achieved from less than $23-25$ measurements, using the l_0 heuristic in contrast to 30 measurements required by AIHT, IRLS, LASSO and 35 measurements required by OMP and l_1 minimisation principle to achieve the same or better level of accuracy. Efficient recovery (10^{-50}) for the same vector is also possible from over-sampled measurements ($70-80$ samples), using the l_0 heuristic, while other sparse recovery methods, such as OMP, AIHT, IRLS, NESTA, fail since the solution guarantees (UUP and RIP conditions) are only conjectured to work in such cases. However, for samples sizes between 35 and 50 the recovery error of the l_1 minimisation principle is 10^{-40} and 10^{-25} for the SL0 method, in contrast to the recovery error between 10^{-8} and 10^{-30} for the l_0 heuristic. Similar results are also possible under the presence of noisy measurements, on condition that we have bounded small additive noise and enough measurements (at least twice the sparsity

level). Sparse recovery using the l_0-heuristic is also possible using sampling matrices with entries drawn from probability distributions (e.g. exponential, geometric, Poisson, Pareto) which are not supported by the majority of sparse recovery methods, such as OMP, IRLS, AIHT, LASSO, AIHT, NESTA and l_1 Magic. These probability distributions do not impose guarantees for the stability of the derived solution as they do not satisfy UUP and RIP conditions, in contrast to Normal and Rademacher distributions used in CS. For example, a 200-element real-valued vector with 20 non-zero entries can be efficiently recovered from 50 samples, drawn from the Poisson and Pareto distributions, with 10^{-10} recovery error.

Towards this perspective the l_0 heuristic was tested to a variety of problems and cases in order to check how stable and categorically meaningful solutions we can derive from synthetic data. Its ability to perform well on such diverse inverse test problems appears to be at least the necessary condition for a heuristic to be trustworthy and robust. The experimental results discussed in Chapter 9 suggest that the l_0 heuristic is able to recover the dictionary and the sparse elements of test vectors (signals or images) with a reduction in time complexity, performance and error, in contrast to other competing sparse recovery methods (iterative, greedy and Basis Pursuit). Its advantages can be summarised as: linear time complexity, efficient recovery for severely under-sampled and over-sampled cases, weaker conditions than those required by UUP and RIP properties suggested in CS theory, problem size reduction (by thresholding K largest entries), efficiency in small random additive noise levels and geometry preservation (edges/boundaries) in sparse image recovery. Another key advantage of the l_0 heuristic is its adaptability based on the type of the problem and the desired accuracy.

We have shown in Section 9.4.2 that for highly under-sampled measurements a small number of iterations (usually twice the level of sparsity) is enough for efficient recovery, while for moderately under-sampled and over-sampled cases, as the number of iterations increases, the level of the accuracy of the estimated solution increases. Furthermore, there seems an interconnection between the number of measurements, sparsity levels and the l_0 heuristic's number of iterations. The smaller the sparsity level and the smaller the number of measurements, the smaller the number of iterations needed for efficient recovery. As the number of measurements increases for small sparsity levels, an increasing number of iterations will improve further the estimated solution. This provides high adaptability of the l_0 heuristic but requires the a-priori knowledge of the sparsity level of the vector to be

recovered. This problem, also common in other sparse recovery approaches such as OMP, AIHT, LASSO and TWIST, can be overcome using the pseudo-inverse or the back-projection from the measurements as a good estimate of the sparsity level of the under-sampled vector.

At this point, it is also important to summarise the drawbacks of the l_0 heuristic, some of which are also common to other sparse recovery methods and to CS theoretical framework in general. The main open problem in sparse recovery is still the reduction of the number of the required measurements, used by sparse recovery methods, to the theoretical bound of $K+1$ (K is the sparsity level) given in [40, 41]. This is still NP-Hard in terms of complexity [92, 146], though there is no formal condition to theoretically guarantee or refute this bound, in a similar way that RIP and UUP guarantee that the solutions of l_0 and l_1 norm based problems are equivalent [189]. Although the l_0 heuristic is able to recover a sparse vector in severely under-sampled cases, it cannot achieve this theoretical measurement bound. This can be attributed to the approximation of the l_0 norm by a smooth, easy to differentiate function, used by the l_0 heuristic. Note also that the l_0 heuristic performs equally better or worse than other approaches (l_1 minimisation principle, l_1 Magic) for moderately under-sampled measurements, where RIP and UUP conditions hold.

Furthermore, the design of sensing matrices, remains open and less regularly discussed in the literature [189]. Although it is easy to verify if a sensing matrix satisfies UUP and RIP properties, the design of such matrix is NP-Hard and no conditions for designing such matrices have been introduced [189]. This imposes two major implications; the need of randomness as an inherent step of the sampling process (satisfying the low mutual coherence property) and the restriction to small bounded additive noise levels for efficient recovery [40, 50]. These are common problems for all known sparse recovery approaches including the l_0 heuristic. Towards this research direction the use of different types of filters (hard or soft thresholding, median or moving average filters) can be an important improvement in the performance of the l_0 heuristic for sparse vector recovery from noisy under-sampled measurements.

A closely related open problem is the design of efficient sparse recovery methods which are not limited by a specific choice of dictionaries. In our experiments we showed that frequency dictionaries in general yield better over-complete decompositions as transformed vectors can be represented by only a few non-zero atoms (coefficients) and thus enhance their sparsity level. In essence, the l_0 norm based problem, though efficient for smaller

sample sizes in contrast to l_1 and l_2 norm based problems, still depends on the sparsity of the vector (signal or image) and thus limited by the nature of the over-complete dictionary. In theory, using combined sparsifying transforms we could enhance the sparsity and thus the morphological diversity of a vector and as a result the l_0 heuristic's performance. The concept of morphological diversity can introduce new data modelling frameworks as a sequence of over-complete waveform dictionaries allowing us to represent and recover a vector very fast and even efficient recovery with only one (in theory) coefficient [186, 213, 215]. The core idea behind this area is that atoms in a dictionary can be modelled as the sum of components which look morphologically different using implicit fast analysis and synthesis operators [185, 186]. In practice, a few transform domains are enough to form a set of dictionaries which sparsify different types of vector features and thus enhance the sparsity level [165, 185, 186]. For instance, it has been proposed in [213] that by combining the Fourier and Wavelet Dictionaries, both stationary and localised features of signals can be well-represented (sparsifying certain types of structures). High separability (morphological diversity) of atoms in a dictionary can also enhance robustness in noise levels or model imperfections [186]. MRI images constitute a representative example where combined sparsifying dictionaries, with low mutual coherence according to RIP, yield fast and implicit recovery [165, 213]. In this case, every feature/component of the image is sparsily represented by different dictionaries, choosing the best sparsest representation among all possible combinations of sparsifying transforms. In line with the results in [165, 185, 186, 213] further theoretical analysis and simulation results are required so as to explore this aspect in terms of the l_0 heuristic's performance.

Another possible direction to overcome the problem with the frequency dictionaries is to adapt the l_0 heuristic so as to learn an over-complete representation, which can encode signals or images more efficiently than complete non-adaptive bases learnt from data, such as Fourier or Wavelet bases. In sparse representations and descriptions it is very common to deal with very large dictionaries where N is on the order of at least 10^6 [28, 215]. To be more efficient a combined dictionary learning and sparse-inverse solving method, which exploits the dictionary's structure of the signal derived, can be introduced. Both of these methods can run in parallel so as to track dictionary evolution until a convergence criterion has been satisfied (i.e. small recovery error metric). This can be based as an extension of FOCUSS method [116, 144]. One possible way to achieve this is to generate randomly encountered sampled

signals at each iteration instead of the original training set (measurements). We also have to ensure that the dictionary learning algorithm is sensitive to new data so that dictionary tracking can occur. This could be done by adaptive filtering of the current dictionary estimate corrected by new data generated, based on modeling their relationship in an iterative manner [157, 210]. A similar approach can be the use of adaptive filtering based on similarity kernels, such as Bilateral filtering based on the Euclidean distance, Kernel Regression [27] based on the spatial distance and LARK based on Geodesic distance [28] [137]. This approach can robustly obtain the signal/image structure leading to a relatively high improvement in performance for over-complete representations.

In terms of better scalability, another venue of research would be to adapt Levy flights (paths) in the generation of new solutions, as an extension of the swarm based l_0 heuristic. Direct path sampling or Levy construction refers to a stochastic interpolation method introduced the French mathematician Paul Levy in 1940. It constitutes a generalised interpolation method, where a path is created based on a local construction discipline [115, 214]. By selecting an interval $[a, b]$ a path is created by a stochastic interpolation of its two end points $x(a)$ and $x(b)$, without considering any other information outside the chosen interval. This construction is actually based on the Levy distribution which is closely related to stable distributions, along with Cauchy and Gaussian distributions, firstly introduced by Levy [115, 207]. These are reliable distributions widely used by mathematicians to explicitly model empirical data in a variety of practical problems and have many interesting mathematical properties. In particular, stable distributions are a class of probability distributions which allow skewness and heavy tails [115, 207]. Another important property they have is that the sum of two independent random variables drawn from these distributions is itself a random variable following the same probability distribution [115, 207]. Moreover, apart from the addition rule for the random variables we also have that the mean of the sum of random variable is the sum of their means and the variance of their sum is also the sum

[27] A locally adaptive regression kernel is an estimation technique to fit the given data without any assumption on the underlying distribution. The idea is to assign a weight to each observation based on the distance from a given data point using a simple non-parametric function depending only on the data given, using for example conditional probabilities.

[28] Geodetic distance is a simple distance metric between coordinates given as $d(X, Y) = \Delta X^T C \Delta X + \Delta X^T X$, where $\Delta X = [dX, dY]^T$ is the difference in spacial coordinates X, Y as $dX = X_i - X_j$ and $dY = Y_i - Y_j$ for any $\{X_i, X_j\} \in X$ and $\{Y_i, Y_j\} \in Y$ and $C = [Z^2X + 1 \;\; ZXZY \; ; \; ZXZY \;\; Z^2Y + 1]$ represents the derivatives of the coordinates, in a similar way as TV norm is defined, with $ZX = X_{i+1} - X_i$ and $ZY = Y_{i+1} - Y_i$ [137].

of their variances, as the shape of these distributions is stable or unchanged in such cases (i.e. property of normalised and centred sums of independent, identically distributed random variables) [115, 207]. A generalised version of the Levy distribution can be defined with the following probability density function [115, 214]:

$$L(x;\mu,c) = \begin{cases} \sqrt{\frac{c}{2\pi}}\exp[-\frac{c}{2(x-\mu)}]\frac{1}{(x-\mu)^{3/2}} & \text{if } 0<\mu<x<+\infty \\ 0 & \text{otherwise,} \end{cases} \tag{10.1}$$

where $\mu > 0$ is a parameter called minimum step (usually $\mu \in (0,2)$) and $c > 0$ is a scaling parameter. Obviously, when $x \to +\infty$ then [214]:

$$L(x;\mu,c) \approx \sqrt{\frac{c}{2\pi}}\frac{1}{x^{3/2}} \tag{10.2}$$

Levy random walks/paths (sampling a path using a Levy distribution) are known to be very efficient in exploring large scale search spaces [115, 214]. At this point it is also important to mention different metrics for cumulative distribution functions (cdf) on a given metric space [29] as a probability measure on random variables defined over a set. Levy distance between two cdf $f, g:\ \mathbb{R} \to [0,1]$ [108, 214]:

$$d_L(f,g) = \inf\ \{\epsilon | \forall x\ f(x-\epsilon) - \epsilon \leq g(x) \leq f(x+\epsilon) + \epsilon\} \tag{10.3}$$

Kolmogorov distance between two cdf $f, g:\ \mathbb{R} \to [0,1]$ [108]:

$$d_K(f,g) = \sup |f(x) - g(x)| \tag{10.4}$$

Prokhorov distance between two cdf $f, g:\ \mathbb{R} \to [0,1]$ [108]:

$$d_P(f,g) = \inf\ \{\epsilon > 0 | f(A) \leq G\{A^\epsilon\} + \epsilon\ \forall A \in \mathbb{R}\}, \tag{10.5}$$

for any subset $A \subset \mathbb{R}$ the closed neighbourhood of A is defined as [108]:

$$A^\epsilon = \{x \in \mathbb{R} | \inf_{y \in A} d(x,y) \leq \epsilon\} \tag{10.6}$$

[29] A metric space is a vector space with norms for calculating distances between members of the set, such as the Hilbert space. For further details please see Appendix B.

Alternatively, another possible direction is to follow the non-linear dimensionality reduction algorithms which adapt a graph embedding framework for solving undirected weighted graphs [129]. Given a data set of samples $Y = [Y_1, Y_2, \ldots, Y_m]$ in M-dimensional space we can create a $M \times M$ data-dependent similarity of affinity matrix [30] W, where the weights $w_{ij} \in [0, 1]$ are used to measure the distance between the two sample observations. A popular approach to measure neighbourhood relationships between samples y_i and y_j makes use of the heat-kernel (i.e. feature descriptor as a vector for representing pixels' local or global geometric properties for shape analysis in 3D images) namely [1, 129]:

$$w_{ij} = \exp(-\frac{\|Y_i - Y_j\|_{l_2}}{\gamma_i \gamma_j}), \tag{10.7}$$

where $\gamma_i = \|Y_i - Y_i^{kn}\|$ denotes the local scaling of data samples in the neighbourhood of Y_i and Y_i^{kn} is the k-nearest neighbor of Y_i. This result has been shown very effective in preserving local properties of the data samples with strong adaptability due to the use of the scaling parameter γ. A more thorough analysis about heats kernels on metric spaces, their properties and their applications can be found at [1, 129].

A slightly different approach to the l_0 norm based problem can be the investigation of the use of the nuclear norm (sum of singular values) of a matrix, which is often used by many heuristics for rank minimisation, matrix factorisation and completion in control theory, multi-task learning, adaptive filtering, signal processing and statistics [9, 28, 98]. Recently the role of the nuclear norm in convex heuristics has been used as an extension (reformulation) of the l_1 constrained minimisation techniques for cardinality minimisation and sparse approximation. Notable research and extensions to nuclear norm methods are discussed in [9, 13, 28, 47, 75, 98]. In particular, the problem of minimising the nuclear norm approximation problem is given as [9, 28, 124]:

$$\min_X \ \|C(X) - Y\|_*, \tag{10.8}$$

or it's penalised regularised variation (or Trace Lasso) using a small parameter $\varrho > 0$,

$$\min_X \ 1/2 \ \|C(X) - Y\|_{l_2} + \varrho \|C(X)\|_*, \tag{10.9}$$

[30] Affinity matrix represents the sum of the probabilities of all paths that can be traversed between two edges (position and direction between two edges) of a graph.

where X are the non-zero entries of a sparse matrix $C(X)$, $Y \in \mathbb{R}^{M\times N}$ is the samples matrix, $C(X) = c_1x_1+c_2x_2+\ldots+c_Nx_N$ is a linear mapping from $\mathbb{R}^N$ to $\mathbb{R}^{M\times N}$. The norm $\|.\|_*$ denotes the nuclear norm on $\mathbb{R}^{M\times N}$, which is the sum of singular values of a matrix, sometimes also known as trace norm, Ky Fan norm and Schatten norm, while the $\|.\|_{l_2}$ denotes the l_2 norm [9, 28, 98]. Note that in some cases the l_2 norm in Equation (10.9) is substituted by a loss function, which is strongly convex, so as to enhance the prediction of $C(X)$ given Y [98]. In general, both problems defined in Equation (10.8) and (10.9) are robust and stable estimators which achieve unique minimum. Indeed, the rank of a matrix X is not a continuous function and thus it is relaxed by the sum of its singular values, expressed as a norm, usually referred to as trace or nuclear norm. Many of the sparse recovery methods designed for the l_1 norm may be extended and used with the nuclear norm as a convex relaxation with a number of applications. For a detailed treatment on this area of active research see [9, 28, 98].

Usually the nuclear norm can be viewed as a convex heuristic for the rank minimisation problem (i.e. $\min \operatorname{rank}(C(X) - Y)$) which is NP-Hard [124]. This minimisation of non-zero singular values (minimisation of the number of linearly independent rows/columns) is of particular interest as a rank of a matrix indicates its linear independence. The singular values of a square matrix, say A are also the square roots of the eigenvalues of A^*A, where A^* is the conjugate transpose or conjugate adjoint of matrix A. Also, the nuclear norm of a diagonal matrix is the l_1 norm of a vector formed by its diagonal elements, namely, $\|\operatorname{diag}(X)\|_* = \|X\|_{l_1}$ for a vector X. Note also that in some problems the nuclear norm is substituted by the Frobenius norm (i.e. $\|X\|_F = \operatorname{trace}(X^TX)$), which is mathematically tractable and efficient [116, 124]. In fact, the squared Frobenius norm is used to measure the size of a matrix with the aim to choose the most appropriate scaling factor of coordinates as a linear function (e.g. $\min 1/2\|C(X) - Y\|_F$) [28, 124]. Many efficient implementations of interior point and penalty decomposition (PD) methods, which exploit the linear matrix structure in low-rank matrix approximations (as a convex regularised rank minimisation problem), have been introduced for the nuclear norm minimisation problem [126, 127]. Towards this approach it would be interesting to provide a theoretical analysis and the necessary conditions (relevant to RIP/UUP properties) for efficient sparse recovery of low rank matrices used in CS framework via nuclear norm optimisation. In particular, it could be possible to extend PD methods so as to solve the l_0 norm based problem since PD methods involve quick vector operations which are known to be efficiently applied in similar areas, namely sparse

logistic regression and sparse covariance selection [126, 127]. More details about numerical methods of minimising the nuclear norm, gradient and PD methods for smooth nuclear norm approximation and their variations can be found in [28, 124, 127].

A highly related with increasing popularity problem in statistical learning and signal processing is that of matrix completion, where data organised in a low-rank matrix can have missing entries due to e.g. limitations in the acquisition process [47, 127]. As a representative example consider the task of inferring answers in a partially filled out survey, derived from questions being asked to a collection of individuals [47]. Similar problems with scientific interest are the linear coding and distributed sensing problems. The linear coding problem can be described as a dual problem in CS, where a signal $X \in \mathbb{R}^N$ is expanded into a larger signal $AX \in \mathbb{R}^M$ where $M > N$ instead of $M < N$ to be transmitted over a noisy network [39, 48]. Even if parts of the transmitted signal are corrupted, so that the data received is of the form $Y = CX + e$ for some sparse e (representing packet loss of corruption), one can recover X efficiently in many cases. The main concept is to view e, rather than X as the signal, in which case one can convert things back to a CS problem [39, 48]. On the other hand, in distributed sensing problems, signals are considered multidimensional with the aim to recover them in time, space, etc. [84]. In such cases we assume signals can be encoded into a more complete structure through the use of an appropriate basis Ψ [84]. This collection (ensemble) of signals as a $L \times N$ matrix ($X = [X_1, X_2, \ldots, X_L] \in \mathbb{R}^N$), where signals correspond to matrix columns and rows represent values of signals in time, location, etc. can be used as an efficient extension for designing better and more robust sparse construction and recovery approaches.

A newly developed application of sparse recovery methods and CS theory in general is the area of satellite image processing and analysis, briefly discussed in Section 3.7.3. Floods, volcanic eruptions, earthquakes and meteorites/comets have long been monitored closely, with the aim to analyse them as potential threats on towns or farmlands. In particular, recent technological advances have enabled us to observe the rise of mountains, the opening of sea basins and the ebb and flow of ice, together with the measurement of the weathering of rockfaces (e.g. rock deformation, erosion and deposition) as well as the infall of interstellar dust from space, from one year to the next, continuously around the globe [58, 202]. All these developments gave impetus to Planetary Geology, Terrametry and Space Exploration as a way of measuring geology changes and understanding the underlying processes as well as their

historical perspective. Image processing and analysis was an integral part of recent planetary missions led by NASA, the European Space Agency, and other national agencies served as an understanding of the composition, internal dynamics, internal structure, thermal evolution, and surface character of both rocky and icy worlds (e.g. the Cassini probe). Volcanoes, impact craters, ice caps, dunes, rift valleys, rivers, fossil fuels and oceans are features of extra-terrestrial worlds as diverse as Mercury and Titan. Different types of filters and image enhancing techniques are now used so as to allow more light to get through the telescope as an efficient way to assign different colours to different filter exposures [58, 202]. This can ease data analysis and enhance our understanding on different planetary bodies and their recent changes subject to earthquakes, landslides and climate change. By combining extensive use of imagery, the results of laboratory experiments and theoretical modeling, geology scientists are now able to use different methods so as to better understand the very origins of our Solar System. Some primary results on experiments where CS has been used for analysing and processing satellite images can be found at [58, 173, 186].

As a particularly useful and of immediate practical interest the l_0 heuristic could be extended and used in some real life satellite images publicly available in NASA, JPL, ESA and Space Science Institute websites as TIFF, JPEG or PNG files that are widely used for comparing and research purposes. Towards this approach, the heuristic could adopt a similarity feature so as to classify and compare and then compress these images for more efficient storage and cataloguing. The cross correlation metric of the mean light intensity of an image $f(i,j)$ can be used for this purpose, namely [58, 111, 182]:

$$c = \frac{\sum_{i,j}(f_1(i,j) - f_2(i,j))}{\sqrt{\sum_{i,j}(f_1(i,j) - f_2(i,j))^2}}, \tag{10.10}$$

which measures the ratio between the covariance of the two images and the product of their standard deviation in terms of light intensity $f_1(i,j)$, $f_2(i,j)$ respectively. The correlation values range on an absolute scale from $[-1, 1]$ which gives a quantitative linear indication between the similarity of two images.

An alternative approach is to use the mutual information for this purpose. The mutual information between two image is expressed in terms of entropy [31] of images which actually

[31] Image entropy used in many compression tests can be calculated as [58, 111]: Entropy $= H =$

measures how well we can predict an arbitrary pixel in an image. For example, in a complete homogenous image we have zero uncertainty while in an image of different tones of gray level we have high uncertainty. In terms of a histogram [32] of the pixel values of an image, a large sharp peak implies low entropy as we have a homogenous image, while broad peaks yield high entropy [63, 160]. This process assumes that if the two images are similar, their entropy will be minimum as peaks of corresponding pixel values are sharpest (i.e. overlap). On the other hand, if the images are different the sharpness of the peaks will decrease and thus resulting in high entropy. Given two images $(1, 2)$ the mutual information M_{12} of the images is defined by [58, 63, 160]:

$$M_{12} = H_1 + H_2 - H_{12}, \tag{10.11}$$

with H_1 and H_2 the entropies of the two images and H_{12} is the entropy of the joint image. In this case we want to maximise the mutual information and thus to minimise the joint entropy so as to check whether the two images are matched. Note that individual entropies need to be maximised also so as to include cases where the images are shifted or rotated so as only a small region of their background overlaps. By this way we consider all cases where even small overlapping regions of the images contain enough information of the images for comparison.

Note that both cross correlation metric and mutual information can also be used in computerised axial tomography (CAT) scanning images as well. CAT incorporates FFT in order to scan and reconstruct an image of a human body from measurements collected by X-RAY beams [117]. CAT method collects information about the human body using different views in a cross-sectional plane so as to synthesize the tomogram (display of a human body in an anatomic plane at a given orientation) needed for further processing and analysis [117]. Towards this research direction, medical Imaging is another area of the most promising CS applications. For example, CS can reduce the time needed for sampling all the data in MRI angiogram, without much of details being lost. In fact, MRI angiogram, revealing the blood vessels of a patient, can be enhanced by various speeds and levels of compression. CS can

$-\sum_j P_j \log_2 P_j$, where P_j represents the histogram counts, i.e. the probability of a difference between two adjacent pixels' light intensity. For example, for two pixels we have $H = -(p \log_2 p + (1-p) \log_2 (1-p))$, where p is the probability of two adjacent pixels.

[32] The entries of a histogram denote the number of times a particular light intensity value of an image is encountered; it denotes the probability distribution of the pixel values of an image [58, 182].

retain both high resolution and contrast compared to other competing techniques, reduced sampling resolution and zero-filled sampling [42, 169, 170].

Similar area of scientific research, which is worth exploring as a potential application is the investigation on how CS and Statistical Mechanics could be introduced and combined together so as to efficiently analyse molecular interactions is the protein-DNA interactions [4]. Molecular interactions act as incoherent sensors for measuring the energy of interactions between atoms and thus the attraction between specific pairs of atoms in the protein and DNA, which formulate the signal vector. These crystal structures are usually obtained (sampled) by X-ray crystallography or nuclear magnetic resonance (NMR) spectroscopy and are publicly available for research through the Protein Data Bank [4, 150]. In contrast to conventional methods, which are heavily based on statistical correlations, on some hypothesised physical theory the new method proposed in [4] determines inter-atomic interaction energy potentials of a molecular system from randomly generated measurements. In fact, without any a-priory assumptions or extensive domain knowledge of the underlying theoretical models, it was possible to predict the energy between proteinDNA complexes with overwhelming probability of approximately 90%, compared to traditional computational methods which achieve approximately 60% [4, 150]. This achievement also has another potential application in Gene theory, where proteinDNA complexes are bound together to create new genetic material into the patient [4, 150].

Altogether the advances made by this thesis open up a number of potential novel research paths that we believe may help make rapid progress in the field of CS. Further experimentation and analysis would be also beneficial in domains where a physically or biologically meaningful sparse solutions are sought and areas with more realistic applications, such as biomedical imaging, geophysical seismic data and objects tracking. Given the increased number of published papers in the area of Compressed Sensing and its closely related areas (sparse recovery, regression and low rank correlation matrix problems), we expect many new results and heuristics to appear soon. Towards this end, further advances in these areas and their applicability in many different areas, may motivate new sparse recovery and processing methods or improvements of the most commonly applied ones, including the least-absolute shrinkage, selection operator (LASSO) and it's weighted variations, so as to outperform existing methods in terms of solution quality and complexity. This thesis can also serve as a reference that attempts to summarise and solve an alternative framework of CS with a num-

ber of applications and thus encourage further research into this unknown, yet interesting, scientific forefront. Ultimately, the results presented in this thesis highlighted new directions as well as new relations to more traditional sparse recovery methods, with the hope of serving both as a quick review of this emerging field and as a reference for researchers who attempt to implement some existing ideas in perspective of practical and theoretical aspect.

Bibliography

[1] Grigoryan A. Heat kernels on metric measure spaces. Lectures at Cornell Probability Summer School, July 2010.

[2] Carlo Alabiso and Ittay Weiss. *A Primer on Hilbert Space Theory*. Springer International Publishing, 2015.

[3] Grègoire Allaire. *Numerical Analysis and Optimization: An Introduction to Mathematical Modelling and Numerical Simulation*. Oxford University Press, 2007.

[4] Mohammed AlQuraishi and Harley H. McAdams. Direct inference of proteindna interactions using compressed sensing methods. *Proceedings of the National Academy of Sciences*, 108(36):14819–14824, 2011.

[5] Theofanis Apostolopoulos and Aristidis Vlachos. Application of the firefly algorithm for solving the economic emissions load dispatch problem. *International Journal of Combinatorics*, pages 1–23, January 2011.

[6] S. Ashkiani, M. Babaie-Zadeh, and C. Jutten. Error correction via smoothed l_0-norm recovery. *IEEE Statistical Signal Processing Workshop (SSP)*, pages 289–292, June 2011.

[7] Calibrated Imaging Lab at Carnegie Mellon University. Computer vision test images. http://www.cs.cmu.edu/afs/cs/project/cil/ftp/html/v-images.html.

[8] Bequette B. Wayne. *Process Dynamics: Modeling, Analysis and Simulation*. Prentice Hall International Series in Physical and Chemical Engineering Sciences, New York, NY, USA, 1998.

[9] Francis Bach, Rodolphe Jenatton, Julien Mairal, and Guillaume Obozinski. Optimization with sparsity-inducing penalties. In J. Shawe-Taylor, R.S. Zemel, P.L. Bartlett, F. Pereira, and K.Q. Weinberger, editors, *Foundations and Trends in Machine Learning*, volume 24, pages 1–106, 2011.

[10] Anatoly Bakushinsky, Mihail Kokurin, and Alexandra Smirnova. *Iterative Methods for Ill-Posed Problems*. De Gruyter, March 2011.

[11] Richard Baraniuk. Compressive sensing. *IEEE Signal Processing Magazine*, pages 118–120, 2007.

[12] Richard Baraniuk, Mark Davenport, Ronald DeVore, and Michael Wakin. A simple proof of the restricted isometry property for random matrices. *Constructive Approximation*, 28(3):253–263, 2008.

[13] S. Becker, J. Bobbin, and E. J. Candès. Nesta: A fast and accurate first-order method for sparse recovery. Technical report, Department of Applied and Computational Mathematics, California Institute of Technology, Pasadena, CA 91125, April 2009.

[14] Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester. Image inpainting. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '00, pages 417–424, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[15] Prof. Herman J. Bierens. Lecture notes: Introduction to hilbert spaces. http://grizzly.la.psu.edu/ hbierens/LECNOTES.HTM, 2014.

[16] José Bioucas-Dia and Mário Figueiredo. Twist website. http://www.lx.it.pt/ bioucas/TwIST/TwIST.htm.

[17] José Bioucas-Dia and Mário Figueiredo. A new twist: two-step iterative shrinkage/thresholding algorithms for image restoration. *IEEE Transactions on Image Processing*, pages 2992–3004, December 2007.

[18] José Bioucas-Dia and Mário Figueiredo. Two-step algorithms for linear inverse problems with non-quadratic regularization. *IEEE International Conference on Image Processing, ICIP2007*, pages 105 – 108, September 2007.

[19] Thomas Blumensath. Iterative hard thresholding: Theory and practice. Technical report, Institute for Digital Communications, Signal and Image Processing, The University of Edinburgh, February 2009.

[20] Thomas Blumensath. Normalised iterative hard thresholding: Guaranteed stability and performance. *IEEE Journal of Selected Topics in Signal Processing*, 4(2):298–309, 2010.

[21] Thomas Blumensath. Accelerated iterative hard thresholding. *Signal Processing*, 92(3):752 – 756, 2012.

[22] Thomas Blumensath and Mike E. Davies. Iterative hard thresholding for compressed sensing. *Applied and Computational Harmonic Analysis*, 27(3):265–274, 2008.

[23] Thomas Blumensath and Mike E. Davies. Iterative hard thresholding for compressed sensing. *Applied and Computational Harmonic Analysis*, May 2008.

[24] J. Frederic Bonnans, J. Charles Gilbert, Claude Lemarechal, and Claudia A. Sagastizabal. *Numerical Optimization: Theoretical and Practical Aspects*. Springer-Verlag, Berlin Heidelberg, New York, USA, 1997.

[25] S. Bourguignon, H. Carfantan, and L. Jahan. Regularised spectral analysis of unevenly spaced data. In *Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05). IEEE International Conference on*, volume 4, pages iv/421–iv/424 Vol. 4, March 2005.

[26] S. Bourguignon, C. Soussen, H. Carfantan, and J. Idier. Sparse deconvolution: Comparison of statistical and deterministic approaches. In *Statistical Signal Processing Workshop (SSP), 2011 IEEE*, pages 317–320, June 2011.

[27] Jake V Bouvrie and Tony Ezzat. An incremental algorithm for signal reconstruction from short-time fourier transform magnitude. In *INTERSPEECH*, 2006.

[28] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[29] Nikita Boyko, Gulver Karamemis, and Stan Uryasev. Sparse signal reconstruction: a cardinality approach, February 2010.

[30] Ronald Newbold Bracewell. *The Fourier Transform and Its Applications*. McGraw Hill, 3 edition, 2009.

[31] Richard P. Brent. Algorithm 488: A gaussian pseudo-random number generator. *Communications of ACM*, 17(12):704–706, December 1974.

[32] James D. Broesch, Dag Stranebby, and William Walker. *Digital Signal Processing: Instant Access*. Instant Access. Elsevier Science, 2008.

[33] Charles L. Byrne. *Iterative Optimization in Inverse Problems*. Chapman and Hall/CRC, February 2014.

[34] T. Cai, Tony, Lie Wang, and Guangwu Xu. New bounds for restricted isometry constants. *IEEE Transcactions on Information Theory*, 56:4388–4394, September 2010.

[35] T. Tony Cai. Orthogonal matching pursuit for sparse signal recovery. *IEEE Transactions on Information Theory*, 57:1–26, 2011.

[36] T.T. Cai, Guangwu Xu, and Jun Zhang. On recovery of sparse signals via l_1 minimisation. *IEEE Transactions on Information Theory*, 55(7):3388–3397, July 2009.

[37] E. Candès, J. Bobin, and S. Becker. Nesta: A fast and accurate first-order method for sparse recovery. http://www-stat.stanford.edu/ candes/nesta/nesta.html.

[38] E. Candès and Romberg J. L1-magic: Recovery of sparse signals via convex programming. http://users.ece.gatech.edu/ justin/l1magic/downloads/l1magic.pdf.

[39] E. J. Candès. Compressive sampling. *Proceedings of the International Congress of Mathematicians, Madrid, Spain*, 2006.

[40] E. J. Candès. The restricted isometry property and its implications for compressed sensing. *Comptes Rendus Mathematique*, 346(9-10):589 – 592, 2008.

[41] E. J. Candès and J. Romberg. Sparsity and incoherence in compressive sampling. *Inverse Problems*, 23(3):969–985, June 2007.

[42] E. J. Candès, J. Romberg, and T. Tao. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52(2):489–509, February 2006.

[43] E. J. Candès, J. Romberg, and T. Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics*, 59:1207 – 1223, August 2006.

[44] E. J. Candés and T. Tao. Near-optimal signal recovery from random projections: Universal encoding strategies? *Information Theory, IEEE Transactions on*, 52(12):5406–5425, Dec 2006.

[45] E. J. Candès, M. Wakin, and S. Boyd. Enhancing sparsity by reweighted l_1 minimization. *Journal of Fourier Analysis and Applications*, 14(5):877–905, December 2004.

[46] E. J. Candès and M. B. Wakin. An introduction to compressive sampling. *IEEE Signal Processing Magazine*, 25(2):21–30, March 2008.

[47] Emmanuel Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9:717–772, 2009.

[48] Emmanuel Candès, M. Rudelson, T. Tao, and R. Vershynin. Error correction via linear programming. In *46th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2005*, pages 668 – 681, 2005.

[49] Emmanuel Candès and Terence Tao. The dantzig selector: Statistical estimation when p is much larger than n. *The Annals of Statistics*, 35(6):2313–2351, 2007.

[50] Emmanuel J. Candès and Mark A. Davenport. How well can we estimate a sparse vector? *Applied and Computational Harmonic Analysis*, 34(2):317 – 323, 2013.

[51] L. Chaari, N. Pustelnik, C. Chaux, and J.-C. Pesquet. Solving inverse problems with overcomplete transforms and convex optimization techniques. In *Proceedings of SPIE*, volume 7446, pages 74460U–74460U–14, 2009.

[52] M. Chakraborty and R. Nath. Performance of irls algorithm in compressive sensing. In *Proceedings of the International Conference and Workshop on Emerging Trends in Technology*, ICWET '11, pages 320–323, New York, NY, USA, 2011. ACM.

[53] Tony Chan, Sung Ha Kang, and Jianhong Shen. Euler's elastica and curvature-based inpainting. *SIAM Journal of Applied Mathematics*, 63(2):564–592, 2002.

[54] K. Chang, R. Jang, and C. Iliopoulos. Music genre classification via compressive sampling. *11th International Society for Music Information Retrieval Conference (ISMIR 2010)*, 2010.

[55] R. Chartrand. Exact reconstruction of sparse signals via nonconvex minimization. *IEEE Signal Processing Letters*, 14(10):707–710, October 2007.

[56] Rick Chartrand and Wotao Yin. Iteratively reweighted algorithms for compressive sensing. *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3869 – 3872, March 2008.

[57] Chris Chatfield. *The Analysis of Time Series, An Introduction.* Chapman and Hall/CRC, 6 edition, 2004.

[58] C. H. Chen. *Signal and Image Processing for Remote Sensing.* CRC Press, second edition, 2012.

[59] X. Chen, F. Xu, and Y. Ye. Lower bound theory of nonzero entries in solutions of $l_2 - l_p$ minimization. *SIAM Journal on Scientific Computing*, 32(5):2832–2852, 2010.

[60] Emilie Chouzenoux, Anna Jezierska, Jean-Christophe Pesquet, and Hugues Talbot. A majorize-minimize subspace approach for $l_2 - l_0$ image regularization. *SIAM Journal on Imaging Sciences*, 6(1):563–591, 2013.

[61] Albert Cohen, Wolfgang Dahmen, and Ronald DeVore. Compressed sensing and best k-term approximation. *Journal of the American Mathematical Society*, 22(1):211–231, 2009.

[62] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms.* The MIT Press, third edition, 2009.

[63] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing).* Wiley-Interscience, 2006.

[64] A database for various images for benchmarking the performance of compression algorithms. Compression database; images, compressors, statistics. http://www.cs.cmu.edu/afs/cs/project/cil/ftp/html/v-images.html.

[65] Jon Dattorro. *Convex Optimization & Euclidean Distance Geometry*. Meboo Publishing, first edition, 2008.

[66] M. Davenport, P. Boufounos, M. Wakin, and R. Baraniuk. Signal processing with compressive measurements. *IEEE Journal of Selected Applications in Signal Processing (special issue on Applications of Compressive Sensing)*, 2010.

[67] M.A. Davenport and M.B. Wakin. Analysis of orthogonal matching pursuit using the restricted isometry property. *Information Theory, IEEE Transactions on*, 56(9):4395–4401, Sept 2010.

[68] Mark Davenport. Lecture notes in harmonic analysis for signal processing. http://users.ece.gatech.edu/ mdavenport/ece-8823a-spring2013/notes/.

[69] G. Davis, S. Mallat, and M. Avellaneda. Adaptive greedy approximations. *Constructive Approximation*, 13(1):57–98, 1997.

[70] Ronald A. DeVore. Deterministic constructions of compressed sensing matrices. *Journal of Complexity*, 23(4-6):918 – 925, 2007. Festschrift for the 60th Birthday of Henryk Woniakowski.

[71] Mario Di Francesco, Sajal K. Das, and Giuseppe Anastasi. Data collection in wireless sensor networks with mobile elements: A survey. *ACM Transactions Sensor Networks*, 8(1):1–31, August 2011.

[72] Atul Divekar and Okan Ersoy. Probabilistic matching pursuit for compressive sensing. Technical report, School of Electrical and Computer Engineering, Purdue University, West Lafayette, May 2010.

[73] MathWorks Online Documentation. Resolving out of memory errors. http://uk.mathworks.com/help/matlab/matlab_prog/resolving-out-of-memory-errors.html, 2015.

[74] D. Donoho and J. Tanner. Observed universality of phase transitions in high-dimensional geometry, with implications for modern data analysis and signal processing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 367(1906), October 2009.

[75] D. L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, April 2006.

[76] D. L. Donoho, M. Elad, and V. Temlyakov. Stable recovery of sparse overcomplete representations in the presence of noise. *IEEE Transactions of Information Theory*, 52:6 – 18, January 2006.

[77] David Donoho and Jared Tanner. Counting the faces of randomly-projected hypercubes and orthants, with applications. *Discrete & Computational Geometry*, 43(3):522–541, 2010.

[78] David L. Donoho. Neighborly polytopes and sparse solution of underdetermined linear equations. Technical report, Technical Report published by the Department of Statistics, Stanford University, 2005.

[79] David L. Donoho and Michael Elad. Optimally sparse representation in general (non-orthogonal) dictionaries via l_1 minimisation. *Proceedings of The National Academy of Sciences - PNAS*, 100:2197–2202, March 2003.

[80] D. Donohoand, Y. Tsaig, I. Drori, and J.L. Starck. Sparse solution of under-determined linear equations by stagewise orthogonal matching pursuit. Technical report, University of Stanford, 2006.

[81] Charles Dossal, Gabriel Peyre, and Jalal Fadili. A numerical exploration of compressed sampling recovery. *Linear Algebra and its Applications*, 432(7):1663 – 1679, 2010.

[82] M. Duarte, B. Wakin, and R. Baraniuk. Wavelet-domain compressive signal reconstruction using a hidden markov tree mode. *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Las Vegas, USA*, pages 5137–5140, 2008.

[83] M.F. Duarte and R.G. Baraniuk. Kronecker product matrices for compressive sensing. *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, pages 3650–3653, March 2010.

[84] M.F. Duarte and R.G. Baraniuk. Kronecker product matrices for compressive sensing. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pages 3650–3653, March 2010.

[85] M.F. Duarte and Y.C. Eldar. Structured compressed sensing: From theory to applications. *Signal Processing, IEEE Transactions on*, 59(9):4053–4085, September 2011.

[86] M. Elad, B. Matalon, and M. Zibulevsky. Image denoising with shrinkage and redundant representations. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 1924–1931, June 2006.

[87] Candès Emmanuel and Terence Tao. Decoding by linear programming. *IEEE Transactions on Information Theory*, 51:4203–4215, 2005.

[88] Joachim H.G. Ender. On compressive sensing applied to radar. *Signal Processing*, 90(5):1402–1414, 2010. Special Section on Statistical Signal and Array Processing.

[89] Heinz Werner Engl, Martin Hanke, and A Neubauer. *Regularization of Inverse Problems*. Springer-Verlag Berlin Heidelber Netherlands, May 2000.

[90] Quanlei Fang. *Multivariable Interpolation Problems*. PhD thesis, Virginia Polytechnic Institute and State University, Blacksburg, Virginia, July 2008. PhD Thesis in Mathematics.

[91] Simon Foucart and Ming-Jun Lai. Sparsest solutions of underdetermined linear systems via l_p minimisation for $0 < p \leq 1$. *Applied and Computational Harmonic Analysis*, 26(3):395 – 407, 2009.

[92] Dongdong Ge, Xiaoye Jiang, and Yinyu Ye. A note on complexity of l_p minimisation, February 2010.

[93] John R. Gilbert, Cleve Moler, and Robert Schreiber. Sparse matrices in matlab: Design and implementation. *SIAM Journal on Matrix Analysis and Applications*, 13(1):333–356, 1992.

[94] Signal Global Optimization and Image Processing Toolboxes. Matlab r2014b documentation. MathWorks, http://www.mathworks.co.uk/help/.

[95] Gene Golub and Charles Van Loan. *Matrix Computations*. Johns Hopkins University Press, 2715 North Charles Street, Baltimore, Maryland 21218-4319, third edition, 1996.

[96] I.F. Gorodnitsky and B.D. Rao. Sparse signal reconstruction from limited data using focuss: A reweighted minimum norm algorithm. *IEEE Transactions on Signal Processing*, 45(3):600–616, March 1997. FOCUSS, Matlab code: http://dsp.ucsd.edu/ jf-murray/software.htm.

[97] Michael Grant and Stephen Boyd. Cvx users guide. http://cvxr.com/cvx/cvx usr-guide.pdf.

[98] Edouard Grave, Guillaume R Obozinski, and Francis R. Bach. Trace lasso: a trace norm regularization for correlated designs. In J. Shawe-Taylor, R.S. Zemel, P.L. Bartlett, F. Pereira, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 2187–2195. Curran Associates, Inc., 2011.

[99] Artyom M. Grigoryan and Sos S. Agaian. *Multidimensional Discrete Unitary Transforms: Representation: Partitioning, and Algorithms (Signal Processing and Communications)*. CRC Press, 2003.

[100] Digital Signal Processing Group. Compressive sensing resources. Rice University, http://dsp.rice.edu/cs.

[101] Chen Guang-ya, Huang Xuexiang, and Yang Xiaogi. *Vector Optimization, Set-valued and Variational Analysis*. Springer-Verlag Berlin Heidelber, 2005.

[102] Guan Gui, Wei Peng, and Fumiyuki Adachi. High-resolution compressive channel estimation for broadband wireless communication systems. *International Journal of Communication Systems*, 2012.

[103] C. Guillemot and O. Le Meur. Image inpainting: Overview and recent advances. *Signal Processing Magazine, IEEE*, 31(1):127–144, 2014.

[104] J. Haupt and R. Nowak. Compressive sampling vs. conventional imaging. In *Image Processing, 2006 IEEE International Conference on*, pages 1269–1272, Oct 2006.

[105] Matthew A. Herman and Thomas Strohmer. General deviants: An analysis of perturbations in compressed sensing. *IEEE Journal of Selected Topics in Signal Processing: Special Issue on Compressive Sensing*, pages 1 – 8, April 2010.

[106] C.A. Hlavka, J.L. Dungan, and D.P. Roy. Increasing the accuracy of information from coarse-resolution satellite imagery. *American Geophysical Union*, page C892, December 2003.

[107] Paul Honeine. Entropy of overcomplete kernel dictionaries. *ACM Computing Research Repository (CoRR)*, abs/1411.0161, 2014.

[108] Peter Huber. *Robust Statistics*. Wiley Series in Probability and Statistics, New York, second edition, 2009.

[109] R. Hummel, S. Poduri, F. Hover, U. Mitra, and G. Sukhatme. Mission design for compressive sensing with mobile robots. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 2362–2367, May 2011.

[110] M. Hyder and K. Mahata. An approximate l_0 norm minimization algorithm for compressed sensing. In *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3365–3368, April 2009.

[111] Bernd Jaehne. *Digital Image Processing*. Springer, 2002.

[112] J. Kennedy and R. Eberhart. Particle swarm optimization. *Proceedings of IEEE International Conference on Neural Networks*, 4:1942 – 1948, November 1995.

[113] V. Konstantinovich, V. Vasin, and V. Tanana. *Nonlinear Model Based Process Control*. Inverse and Ill-Posed Problems Series. V.S.P. Intl Science, P.O. Box 346, 3700 AH Zeist, The Netherlands, second edition, November 2002.

[114] Peter Kraniauskas. *Transforms in Signals and Systems*. Addison-Wesley, 1 edition, 1992.

[115] Werner Krauth. *Statistical Mechanics: Algorithms and Computations*. Oxford University Press, 2006.

[116] Kenneth Kreutz-Delgado, Joseph F. Murray, Bhaskar D. Rao, Kjersti Engan, Te-Won Lee, and Terrence J. Sejnowski. Dictionary learning algorithms for sparse representation. *Neural Computation*, 15(2):349–396, February 2003.

[117] Lydia Kronsjö. *Computational Complexity of Sequential and Parallel Algorithms*. John Wiley & Sons, Inc., New York, NY, USA, 1986.

[118] Chinh La and Minh N. Do. Signal reconstruction using sparse tree representation. In *Proceedings of Wavelets XI at SPIE Optics and Photonics*, 2005.

[119] Michael Lamoureux. Tutorial on compressive sampling. *Back to Exploration, 2008 CSPG CSEG CWLS Convention Conference*, May 2008.

[120] Averill M. Law. *Simulation Modeling and Analysis*. Mcgraw Hill Higher Education, 4 edition, 2006.

[121] S. Lefkimmiatis, A. Bourquard, and M. Unser. Hessian-based norm regularization for image restoration with biomedical applications. *Image Processing, IEEE Transactions on*, 21(3):983–995, March 2012.

[122] Shancang Li, Xinheng Wang, Xu Zhou, and Jue Wang. Efficient blind spectrum sensing for cognitive radio networks based on compressed sensing. *EURASIP Journal on Wireless Communications and Networking*, 2012(1):306, 2012.

[123] S.G. Lingala and M. Jacob. Blind compressive sensing dynamic mri. *Medical Imaging, IEEE Transactions on*, 32(6):1132–1145, June 2013.

[124] Zhang Liu and Lieven Vandenberghe. Interior-point method for nuclear norm approximation with application to system identification. *SIAM J. Matrix Anal. Appl.*, 31(3):1235–1256, November 2009.

[125] Tecnick.com LTD. Testimages. http://www.tecnick.com/public/code/cp-dpage.php?aiocp-dp=testimages.

[126] Zhaosong Lu and Yong Zhang. Penalty decomposition methods for l_0-norm minimization. *CoRR*, abs/1008.5372, 2010.

[127] Zhaosong Lu and Yong Zhang. Penalty decomposition methods for rank minimisation. *CoRR*, abs/1008.5373, 2010.

[128] G. David Luenberger. *Investment Science*. Oxford University Press, 1998.

[129] D. Lunga, S. Prasad, M.M. Crawford, and O. Ersoy. Manifold-learning-based feature extraction for classification of hyperspectral data: A review of advances in manifold learning. *IEEE Signal Processing Magazine*, 31(1):55–66, 2014.

[130] Grant M. and Boyd S. Cvx: Matlab software for disciplined convex programming, version 1.22, available at: http://cvxr.com/cvx/, April 2012.

[131] S.G. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397–3415, Dec 1993.

[132] Stephane Mallat. *A Wavelet Tour of Signal Processing, Third Edition (The Sparse Way)*. Academic Press, 3 edition, 2009.

[133] George Marsaglia and Wai Wan Tsang. The ziggurat method for generating random variables. *Journal of Statistical Software*, 5(8):1–7, 10 2000.

[134] Bruno Masiero and Martin Pollow. A review of the compressive sampling framework in the lights of spherical harmonics: Applications to distributed spherical arrays. In *in Proceedings of the 2nd International Symposium on Ambisonics and Spherical Acoustics, Paris, France*, May 2010.

[135] Jeffrey J. McConnell. *Analysis of Algorithms: An Active Learning Approach*. Jone and Bartlett Publishers, World Headquarters, Jones and Bartlett Publishers, 40 Tall Pine Drive, Sudbury, MA 01776, 978-443-5000, first edition, 2001.

[136] Alfred Mertins. *Signal Analysis: Wavelets, Filter Banks, Time-Frequency Transforms and Applications*. John Wiley & Sons Ltd, West Sussex, England, 1999.

[137] P. Milanfar. A tour of modern image filtering: New insights and methods, both practical and theoretical. *Signal Processing Magazine, IEEE*, 30(1):106–128, Jan 2013.

[138] Lai Ming-Jun and Liu Yang. The null space property for sparse recovery from multiple measurement vectors. *Applied and Computational Harmonic Analysis*, pages 1–10, November 2010.

[139] Moshe Mishali and Yonina C. Eldar. From theory to practice: Sub-nyquist sampling of sparse wideband analog signals. *CoRR*, abs/0902.4291, 2009.

[140] Isi Mitrani. *Probabilistic Modelling*. Cambridge University Press, 1998.

[141] H. Mohimani, M. Babaie-Zadeh, and C. Jutten. Complex-valued sparse represntation based on smoothed l_0 norm. *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 3881–3884, April 2008.

[142] H. Mohimani, M. Babaie-Zadeh, and C. Jutten. A fast approach for overcomplete sparse decomposition based on smoothed l_0 norm. *IEEE Transactions on signal processing*, 57(1):289–301, November 2009.

[143] S. Bazaraa Mokhtar, D. Sherali Hanif, and Shetty C. M. *Nonlinear Programming, Theory and Algorithms*. Willey Interscience, 3 edition, 2006.

[144] Joseph F. Murray. Focuss/focuss-cndl website. http://dsp.ucsd.edu/ jfmurray/software.htm.

[145] S. Muthukrishnan. Some algorithmic problems and results in compressed sensing. *Forty-Fourth Annual Allerton Conference, Allerton House, UIUC, Illinois, USA*, September 2006.

[146] B. Natarajan. Sparse approximate solutions to linear systems. *SIAM Journal on Computing*, 24(2), April 1995.

[147] Richard Neapolitan and Kumarss Naimipour. *Foundations of Algorithms*. Jones and Bartlett Publishers, fourth edition, 2010.

[148] Needell and R. Vershynin. Uniform uncertainty principle and signal recovery via regularized orthogonal matching pursuit. Technical report, University of California, Davis, 2007.

[149] D. Needell and J. A. Tropp. Cosamp: Iterative signal recovery from incomplete and inaccurate samples. Technical report, California Institute of Technology, Pasadena, 2008.

[150] National Energy Research Scientific Computing Center (NERSC). New mathematical method reveals where genes switch on or off. https://www.nersc.gov/news-publications/news/science-news/2012/new-mathematical-method-reveals-where-genes-switch-on-or-off/, 2012.

[151] Mila Nikolova. Description of the minimizers of least squares regularized with l_0 norm. uniqueness of the global minimizer. *SIAM Journal on Imaging Sciences*, 6(2):904–937, 2013.

[152] Jorge Nocedal and J. Stephen Wright. *Numerical Optimization*. Springer Series in Operation Research and Financial Engineering. Springer, Springer Science+BusinessMedia, LLC, 233 Spring Street, New York, NY 10013, USA, 2 edition, 2006.

[153] S. Novikov and I. Ryabtsov. Optimization of frame representations for compressed sensing and mercedes-benz frame. *Proceedings of the Steklov Institute of Mathematics*, 265:199–207, 2009.

[154] University of British Columbia. Seismic laboratory for imaging and modeling. http://https://www.slim.eos.ubc.ca.

[155] MIT OpenCourseWare: Massachusetts Institute of Technology. Introduction to electrical engineering and computer science. http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-01sc-introduction-to-electrical-engineering-and-computer-science-i-spring-2011/index.htm.

[156] Center for Computer Research in Music On line material and Stanford University Acoustics (CCRMA). Aspects of digital waveguide networks for acoustic modeling applications. https://ccrma.stanford.edu/ jos/wgj/.

[157] Alan V. Oppenheim and Ronald W. Schafer. *Discrete-time Signal Processing*. Pearson Education, United States of America, 2007.

[158] Netpbm package. Pgm format specification. http://netpbm.sourceforge.net/doc/pgm.html.

[159] Daniel Palomar, P. and Yonina Eldar, C. *Convex Optimization in Signal Processing and Communications*. Cambridge University Press, 1 edition, 2009.

[160] Liam Paninski. Estimation of entropy and mutual information. *Neural Computations*, 15(6):1191–1253, June 2003.

[161] J.K. Pant, Lu Wu-Sheng, and A. Antoniou. Reconstruction of sparse signals by minimizing a re-weighted approximate l_0-norm in the null space of the measurement matrix. *Circuits and Systems (MWSCAS), 53rd IEEE International Midwest Symposium*, pages 430–433, August 2010.

[162] Imre Pólik. Sedumi 1.1 user's guide. http://sedumi.ie.lehigh.edu/.

[163] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes in C, The Art of Scientific Computing*. Cambridge University Press, 2 edition, February 1996.

[164] Andrea Prosperetti. *Advanced Mathematics for applications*. Cambridge University Press, 2011.

[165] X. QU, X. Cao, D. Guo, C. Hu, and Z. Chen. Combined sparsifying transforms for compressed sensing mri. *Electronics Letters*, 46(2):121–123, January 2010.

[166] S. Singiresu Rao. *Engineering Optimization: Theory and Practice*. Wiley-Interscience, 4 edition, 2009.

[167] Rice University Richard Baraniuk. Compressive sensing: A new framework for imaging (slides). http://www.cs.jhu.edu/ misha/ReadingSeminar/Papers/Baraniuk06.pdf., 2009.

[168] T. Roberts, N. Kingsbury, and D.J. Holland. Sparse recovery of complex phase-encoded velocity images using iterative thresholding. In *20th IEEE International Conference on Image Processing (ICIP), 2013*, pages 350–354, Sept 2013.

[169] Justin Romberg. Sensing by random convolution. *IEEE Int. Work. on Comp. Adv. Multi-Sensor Adaptive Proc., CAMPSAP*, pages 137–140, 2007.

[170] Justin Romberg. Imaging via compressive sampling. *Signal Processing Magazine, IEEE*, 25(2):14–20, March 2008.

[171] M. Rudelson and R. Vershynin. Sparse reconstruction by convex relaxation: Fourier and gaussian measurements. In *Information Sciences and Systems, 2006 40th Annual Conference on*, pages 207–212, March 2006.

[172] Guillermo Sapiro. Image inpainting. *SIAM News*, 35(4):1–2, May 2002.

[173] F. Schmidt, S. Bourguignon, S. Le Mouelic, N. Dobigeon, C. Theys, and E. Treguier. Accuracy and performance of linear unmixing techniques for detecting minerals on omega/mars express. In *Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS), 2011 3rd Workshop on*, pages 1–4, June 2011.

[174] K.M. Shadimetov, A.R. Hayotov, and N.H. Mamatova. Optimal interpolation formulae in the periodic function space of s.l. sobolev. *ArXiv e-prints*, December 2010.

[175] Sukrit Shankar. Talking time to frequency transformation. *IEEE Potentials*, 31(6):16–21, November-December 2012.

[176] Claude Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, 623–656, July, October 1948.

[177] Scott Chen Shaobing, David L. Donoho, and Michael A. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20:33–61, 1998.

[178] Jerome M. Shapiro. Embedded image coding using zerotrees of wavelet coefficients. *IEEE Transaction on Signal Processing*, 41(12):3445–3462, December 1993.

[179] A. Sharif-Nassab, M. Kharratzadeh, M. Babaie-Zadeh, and C. Jutten. How to use real-valued sparse recovery algorithms for complex-valued sparse recovery? In *Proceedings of the 20th European Signal Processing Conference (EUSIPCO), 2012*, pages 849–853, Aug 2012.

[180] Yoav Sharon, John Wright, and Yi Ma. Computation and relaxation of conditions for equivalence between l_1 and l_0 minimization. Technical report, Coordinated Science Laboratory at University of Illinois, Urbana-Champaign, 2007.

[181] Jianhong She. Inpainting and the fundamental problem of image processing. Technical report, School of Mathematics, University of Minnesota, Minneapolis, MN 55455, USA, December 2002.

[182] Milan Sonka, Vaclav Hlavac, and Roger Boyle. *Image Processing, Analysis and Machine Vision*. International Student Edition. Thomson Learning Corporation, 3 edition, 2008.

[183] Emmanuel Soubies, Laure Blanc-Feraud, and Gilles Aubert. A continuous exact l_0 penalty (cel0) for least squares regularized problem. *SIAM Journal on Imaging Sciences*, 8(3):1607–1639, 2015.

[184] Springer-Verlag. Encyclopaedia of mathematics. http://eom.springer.de/default.htm.

[185] J-L Starck, M. Elad, and D.L. Donoho. Image decomposition via the combination of sparse representations and a variational approach. *Image Processing, IEEE Transactions on*, 14(10):1570–1582, Oct 2005.

[186] Jean-Luc Starck, Fionn Murtagh, and Jalal M. Fadili. *Sparse Image and Signal Processing; Wavelets, Curvelets, Morphological Diversity*. Cambridge University Press, United Kingdom, 2010.

[187] F. Jos Strum. Using sedumi 1.05, a matlab toolbox for optimisation over symmetric cones. Technical report, Department of Econometrics, Tilburg University, Tilburg, The Netherlands, October 2001.

[188] Kunio Takezawa. *Introduction to Nonparametric Regression*. Wiley Series in Probability and Statistics, 2005.

[189] Terence Tao. Open question: Deterministic uup matrices. http://terrytao.wordpress.com/2007/07/02/open-question-deterministic-uup-matrices/, 2007.

[190] A. Tarantola. *Inverse Problem Theory and Methods for Model Parameter Estimation*. Society for Industrial and Applied Mathematics, 1 edition, 2005.

[191] S.J. Tarsa and H.T. Kung. Output compression for ic fault detection using compressive sensing. In *MILITARY COMMUNICATIONS CONFERENCE, 2012 - MILCOM 2012*, pages 1–6, Oct 2012.

[192] David B. Thomas, Wayne Luk, Philip H.W. Leong, and John D. Villasenor. Gaussian random number generators. *ACM Computing Surveys*, 39(4), November 2007.

[193] Robert Tibshirani. Regression shrinkage and selection via the lasso: a retrospective. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(3):273–282, 2011.

[194] A. Tikhonov and V. Arsenin. *Solutions of Ill Posed Problems (Translated)*. Scripta Series in Mathematics. V.H. Winston and Sons, 1511 K Street, N.W. Washington D.C., first edition, January 1977.

[195] Matlab Image Processing Toolbox. Remove noise from images tutorial. http://www.mathworks.co.uk/help/images/remove-noise-from-images.html, 2014.

[196] J.A. Tropp and S.J. Wright. Computational methods for sparse solution of linear inverse problems. *Proceedings of the IEEE*, 98(6):948–958, June 2010.

[197] Joel A. Tropp. Greed is good: Algorithmic results for sparse approximation. *IEEE Transactions on Information Theory*, 50:2231–2242, 2004.

[198] Joel A. Tropp. On the conditioning of random subdictionaries. *Applied and Computational Harmonic Analysis*, 25(1):1 – 24, 2008.

[199] Joshua Trzasko and Armando Manduca. Highly undersampled magnetic resonance image reconstruction via homotopic l_0-minimisation. *IEEE Transactions on Medical Imaging*, 28(1):106–121, January 2009.

[200] Yi-Min Tsai, Keng-Yen Huang, H.T. Kung, D. Vlah, Y.L. Gwon, and Liang-Gee Chen. A chip architecture for compressive sensing based detection of ic trojans. In *Signal Processing Systems (SiPS), 2012 IEEE Workshop on*, pages 61–66, Oct 2012.

[201] Department of Computer Science Visual Information Processing Group and Spain Artificial Intelligence, Universidad de Granada. Bayes least-squares gaussian scale mixtures (bls-gsm) de-noising method. http://decsai.ugr.es/ javier/denoise/index.html.

[202] Claudio Vita-Finzi and Dominic Fortes. *Planetary Geology: An Introduction*, volume Paperback. Dunedin Academic Press, 2 edition, June 2013.

[203] Michael Wakin. Compressed sensing, September 2009.

[204] Miaohui Wang, King Ngi Ngan, and Long Xu. Spatial-temporal decorrelation for image/video coding. In *Picture Coding Symposium (PCS), 2012*, pages 201–204, May 2012.

[205] Michael Weeks. *Digital Signal Processing Using Matlab and Wavelets (Engineering)*. Infinity Science Press, 1 edition, 2006.

[206] Thomas Weise. *Global Optimization Algorithms, Theory and Application*. it-weise.de (self-published): (Germany), 2009.

[207] Eric W. Weisstein. Mathworld–a wolfram web resource. http://mathworld.wolfram.com.

[208] J. Darren Wilkinson. *Stochastic Modelling for Systems Biology*. Chapman and Hall/CRC, 2006.

[209] R.M. Willett, M.F. Duarte, M.A. Davenport, and R.G. Baraniuk. Sparsity and structure in hyperspectral imaging : Sensing, reconstruction, and target detection. *Signal Processing Magazine, IEEE*, 31(1):116–126, 2014.

[210] Douglas B. Williams. *The digital signal processing handbook*. CRC Press, Boca Raton, Fla, 1998.

[211] Prof. David P. Williamson. Lecture notes in set covering problem at cornell university (pp.7-18). http://www.orie.cornell.edu/ dpw/cornell.ps, 2010.

[212] Q. Wu, F. Merchant, and K. Castleman. *Microscope Image Processing*. Academic Press, 2010.

[213] Qu Xiaobo, Cao Xue, Guo Di, Hu Changwei, and Chen Zhong. Compressed sensing mri with combined sparsifying transforms and smoothed l_0 norm minimisation. *Acoustics Speech and Signal Processing (ICASSP), IEEE International Conference*, pages 626–629, March 2010.

[214] Yang Xin-She. *Nature-Inspired Metaheuristic Algorithms*. Luniver Press, United Kingdom, 2008.

[215] C. Eldar Yonina and Kutyniok Gitta. *Compressed Sensing: Theory and Applications*. Cambridge University Press, 2012.

[216] N. Young. *An Introduction to Hilbert Space*. Cambridge University Press, 1988.

[217] Tong Zhang. Sparse recovery with orthogonal matching pursuit under rip. *IEEE Transactions on Information Theory*, 57(9):6215–6221, Sept 2011.

[218] M. Zibulevsky and M. Elad. L1-l2 optimization in signal and image processing. *IEEE Signal Processing Magazine*, 27(3):76–88, May 2010.

Appendix A

Finite dimensional vector spaces

During the last few years there has been a tremendous progress in the area of image processing and particularly in image analysis using different spaces which represent an image as a vector. An image can be interpreted as a representation of a vector in a Banach, Sobolev or more commonly in a Hilbert space [33]. For this reason sometimes Banach, Sobolev or Hilbert spaces are also called vector spaces. Note, however, that by the term vector spaces we do not mean ordinary vectors in $\mathbb{R}^N$ or $\mathbb{C}^N$ (vectors of length N) but "spaces of vector-like objects" where more general sets of objects (e.g. functions) follow some basic algebraic properties of vectors. These vector spaces, just as the conventional vectors, can either be the real $\mathbb{R}$ or complex numbers $\mathbb{C}$. Furthermore, normed spaces are simple metric properties (e.g. Euclidean distance), while an inner product space simply augments geometric properties (e.g. angles).

Although real life signals, such as images, are formulated in the continuous domain we need to develop signal processing methods for discrete cases so as to efficiently implement them on a computer. We are also interested in nonband-limited cases [34], which are more

[33] For a more detailed analysis and formal proofs you can refer to [164] (Chapter 19, Infinite Dimensional Spaces), [3] (Appendix, Review of Hilbert Spaces), [174] (quick yet thorough overview of Hilbert space and its properties) and [216] (Chapters 1 and 2, Inner Product and Normed Spaces).

[34] A signal is said to be a band-limited signal if all of it's frequency components (in Fourier transform) are zero above a certain finite frequency (it's power spectral density is zero above the finite frequency). In other words, if the Fourier transform or power spectral density has finite support (i.e. finite frequency content) then the signal is said to be band-limited [155, 156]. The simplest case is the sinusoidal signal (e.g. $f(t) = sin(\omega t)$), whose Fourier transform consists of a delta function centred on the frequency of the signal. A

general and imply the need for continuous-to-discrete and discrete-to-continuous conversion systems in contrast to the traditional wisdom dictated by the famous Shannon-Nyquist sampling theorem. In fact, the formulation needed in Unitary transforms, such as Fourier, Wavelets, etc. together with the input and output of a system [35] need to lie in certain signal subspaces.

Unitary transforms, as a commonly used tool in signal/image processing, are based on vector spaces. These transforms are defined by a number of basis functions which are periodical so as enhance the sparsity of a vector. In general, the aim is to find a representation for vectors in a separable vector space, analogous to an orthogonal vector basis in a finite-dimensional space (discussed in Section 2.3). For this reason Banach, Sobolev and Hilbert spaces structure has been used to efficiently represent signals which are functions of continuous variables as sequences of numbers in discrete representation (using an orthonormal basis) so as to be processed numerically (as a set of functions in a dictionary). Finite dimensional spaces also have high applicability in other areas such as communications, astronomy, seismology, biomedical engineering and crystallography [99, 164]. Finally, note that throughout this thesis we consider functions whose corresponding space can be defined by the l_2 norm with respect to $\mathbb{R}$ or C and are uniquely characterised by a sequence of coefficients in a known basis. We also consider only dictionaries which are complete in a vector space and thus they can uniquely approximate any given function as a linear combination of elements in a dictionary. Therefore, only orthogonal dictionaries are considered as they achieve a good K term approximation by collecting the K largest coefficients in a basis. This dictionary can be a wavelet basis for a vector space ($L^p, 1 < p < \infty$), the canonical basis (i.e. standard basis defined by the dirac delta function) or more generally a Unitary transform basis.

Preliminaries [3, 65, 99, 164, 216] Here, we will assume that matrices are relatively simple examples of linear operators which operate on finite dimensional spaces. A general definition of a linear vector space S over a set of real $\mathbb{R}$ (or complex C) numbers is a set of entities called vectors or alternatively elements of a space, which can be combined such as a way

band-limited signal can be reconstructed exactly if it is sampled at more than twice the maximum frequency present in the signal (Shannon-Nyquist sampling theorem). Sampling as a general concept converts a signal of continuous domain to one of discrete domain.

[35] A System, in the sense of this thesis, is any physical set of components that transforms (performs an operation such as a transform) the input signal into the output signal [155, 156].

that:

$$w = au + bv \ \in S \ \ \forall u, v \in S \text{ and } \forall a, b \in \mathbb{R}$$

The real values a,b here are considered as scalars, i.e. small numbers. We say that vectors $u_1, u_2, \ldots, u_k$ are linearly independent if

$$\sum_{j=1}^{K} a_j u_j = \varnothing,$$

where $\varnothing$ represents the zero vector (i.e. vectors are perpendicular or orthogonal to each other). Note that this relation is satisfied only if all the a_j are zero. Moreover, the space is K-dimensional if no set of linearly independent vectors contains more than K elements. Let $x_1, x_2, \ldots, x_k$ be a set of linearly independent vectors in an K-dimensional space. Any other vector u included will make the set linearly dependent so that

$$u = \sum_{i=1}^{K} u_i x_i,$$

must exist, with K real numbers u_i. This is a unique representation due to the assumed linear independence of x_i. Any set of $x_1, x_2, \ldots, x_k$ which permits a representation of an arbitrary element of the space in this way is called basis in K-dimensional space S^K. Any K-dimensional space has a basis, which contains exactly K linearly independent vectors, and any set of linearly independent vectors $x_1, x_2, \ldots, x_l$, with $L < K$, which can be combined by vectors $x_{l+1}, \ldots, x_k$, so as the combination of two sets forms a basis. The u_i are also called coordinates of u in the basis $x_1, x_2, \ldots, x_k$. This representation forms a one-to-one correspondence between a K-dimensional linear vector space S^K and a space of $\mathbb{R}^K$ (or $\mathbb{C}^K$) of K-dimensional sets of real or complex numbers. In fact, all K-dimensional vector spaces can be represented by a space of $K \times 1$ vectors consisting of the coordinates of the elements of the vector space. The set of all possible linear combinations of L given vectors $u_1, u_2, \ldots, u_l \in S^K$ (with $|1, 2, \ldots, l| = L < K$) is also a subspace of S^K, spanned by these L vectors. Obviously, by definition S^K and $\varnothing$ are also trivial subspaces of dimensions K and 0 respectively.

Consider now two subspaces S_1 and S_2 which have an empty set in common ($\varnothing$). Then every $u \in S$ can be represented as $u = u_1 + u_2$, when $u_1 \in S_1$ and $u_2 \in S_2$. Thus, S is

decomposed into the sum of two subspaces as $S = S_1 \oplus S_2$. Assume now more generally that u_j represents a vector u on a basis x_j of S^K as previously. Assume also that a $L \times K$ matrix $A = a_{ij}$ preforms a linear transformation correspondence between the vector $u \in S^K$ and a vector $v \in S^L$, which has coordinates v_i in a basis ϕ_i in a space S^L, as follows:

$$v_i = \sum_{j=1}^{K} a_{ij} u_j.$$

In general, if the space is finite dimensional then the sum is finite and by definition all the bases have the same number of elements. Otherwise, if the space is infinite dimensional and separable, all the bases have a countable infinity of elements. In either case, the definition defines actually a linear operator matrix A, which can be written in vector notation as $v = Au$. This set of vectors $Au \in S^L$ ranges over the entire space S^K and it is actually a subspace spanned by vectors $Au_1, Au_2, \ldots, Au_k$ which are not necessary linearly independent.

The set of vectors u such that $Au = \varnothing$ is the null space or kernel of A, denoted by Null(A) or kernel(A). The maximum number of linearly independent vectors in Null(A) is called nullity of A. Notice, however, that it is possible a non-zero vector u may exist such that $Au \neq \varnothing$, but $A^2u = \varnothing$. Finally, note that as we have discussed in Section 2.3 the matrix A is also called transformation matrix, as it determines a linear one-dimensional transformation. In other words, A contains the coordinates of the transformed basis vector with respect to the original basis. In a more general notation we can write $v = Au$ or $u = A^{-1}v$, which means by definition that A is invertible so as for any given $v \in S^K$ there must be only one u such that $Au = v$. Therefore since there are only K linearly independent vectors in a space of dimension K, for any transformation matrix (i.e. operator) A mapping S^K and any non-zero vector u, vectors $A^j u$ with $j \geq K$ must be linear combinations of $u, Au, A^2u, \ldots, A^{K-1}u$. Of course this does not exclude the possibility that for some vector u and some $j < K - 1$, $A^j u$ can be written as a linear combination of $u, Au, A^2u, \ldots, A^j u$.

Best Approximation [132, 164, 216] Let's assume a finite dimensional subspace defined by vectors $u_1, u_2, \ldots, u_k$. Starting from each of these vectors, we construct another vector with unit norm as follows:

$$e_i = \frac{u_i}{\|u_i\|_{l_2}}, \quad \forall i = 1, \ldots, k,$$

where $\|u_i\|_{l_2} = \sqrt{< u, u >}$ (i.e. square root of scalar product of two vectors). It can be easily verified that the vectors e_i constitute an orthonormal set. Now we want to determine K real valued numbers b_j:

$$w = \sum_{j=1}^{K} b_j e_j,$$

so as the difference between u and w is very small. Since $< e_j, e_i >= \delta_{ji}$, the norm indicating the difference between the two vectors is given by:

$$\|u - w\|_{l_2} = \|u - \sum_{j=1}^{K} b_j e_j\|_{l_2} =< u, u > - \sum_{j=1}^{K} b_j < e_j, u > - \sum_{j=1}^{K} b_j < u, e_j > + \sum_{j=1}^{K} \|b_j\|_{l_2},$$

or, if we add and subtract $\sum_{j=1}^{K} \| < e_j, u > \|_{l_2}$, we have:

$$\|u - \sum_{j=1}^{K} b_j e_j\|_{l_2} =< u, u > + \sum_{j=1}^{K} \|b_j - < e_j, u > \|_{l_2} - \sum_{j=1}^{K} \| < e_j, u > \|_{l_2},$$

which implies that the best K term approximation for u is obtained by the elements of the dictionary if we set $b_j =< e_j, u >$ and thus:

$$w = \sum_{j=1}^{K} < e_j, u > e_j.$$

This implies that $\sum_{j=1}^{K} \| < e_j, u > \|_{l_2} \leq \|u\|_{l_2}$, since $\|u - \sum_{j=1}^{K} < e_j, u > e_j\|_{l_2} = \|u\|_{l_2} - \sum_{j=1}^{K} \| < e_j, u > \|_{l_2}$. Note also that in general for an infinite dimensional space (e.g. Hilbert space) the series $\sum_{j+1}^{\infty} e_j a_j$ and $\sum_{j+1}^{\infty} \|a_j\|_{l_2}$ converge or diverge together, for any orthonormal set e_j and a sequence of real or complex numbers a_j. The Fourier transform is based on these fundamental properties to achieve orthogonality and thus the minimum energy between different bases.

Banach Spaces [132, 216]Banach space $L^p(\mathbb{R}) = \mathcal{B}$ is considered as an infinite dimensions vector space with the norm but without inner product. In fact a generalised Banach space with an inner product is a Hilbert space so as to support the necessary orthogonality in bases. In general, signals and images are considered as vectors and thus a distance within a

vector space can be defined using a norm, according to the following properties:

$$\forall f \in \mathcal{B}, \quad \|f\|_{l_p} \geq 0 \text{ and } \|f\|_{l_p} = 0 \text{ iff } f = 0,$$
$$\forall \lambda \in \mathbb{R}, \quad \|\lambda f\|_{l_p} = |\lambda| \|f\|_{l_p},$$
$$\forall f, g \in \mathcal{B}, \quad \|f + g\|_{l_p} \leq \|f\|_{l_p} + \|g\|_{l_p}$$

where f, g are measurable functions in Banach space with $\|f\|_{l_p} < +\infty$ and $\|g\|_{l_p} < +\infty$ with norm:

$$\|f\|_{l_p} = \left(\int_{-\infty}^{+\infty} (f(x))^p dx \right)^{1/p} < +\infty,$$

for any $p \geq 1$. Using the previous norm the convergence of a sequence of $\{f_1, f_2, \ldots, f_n\}$ implies that:

$$\lim_{n \to +\infty} f_n = f \quad \text{or} \quad \lim_{n \to +\infty} \|f_n - f\|_{l_p} = 0$$

which is an important property so as a vector space is complete (every sequence in $\mathcal{B}$ converges to an element in $\mathcal{B}$).

Sobolev Spaces [132, 164, 174] Sometimes infinite spaces (in L^p, with $p > 1$) are too large for specific problems and solutions due to specific boundary conditions (e.g. in spectral theory). For example, we may need to solve a function which belongs to $L^2(-\infty, +\infty)$ but it does not necessarily tend to 0 at infinity as it might not be defined or it might be non-smooth at its end points. Subspaces, such as Sobolev spaces, are particularly useful on these cases, as they support both differentiation and the structure of inner product in function space. Let's consider a function $f(x_1, x_2, \ldots, x_n)$ defined over a region $S \in \Re^N$. Then the Sobolev space $W^{1,2}$ is consisted of the function f itself and all its distributional first order partial derivatives $\partial f / \partial x_i$ which belong to $L^2(S)$:

$$W^{1,2} = \{f \in L^2(S), \partial f / \partial x_i \in L^2(S)\}, i = 1, 2, \ldots, n.$$

The norm of the function f can then be defined as:

$$\|f\|^2_{W^{1,p}} = (\|f\|^2_{l_p} + \|\nabla f\|^2_{l_p})^{1/p},$$

for $p = 2$ or more generally using a general p-norm definition for any $L^p(H)$ space, for $p \geq 2$. Another extension is obtained by including partial derivatives up to order k, defining spaces $W^{k,2}$ which are also complete normed separable spaces. The norm is then defined by summing over all derivatives of order $\leq k$. Note that this subspace S is not a closed linear manifold [36] (i.e. linear closed vector subspace), but it can become a complete separable Hilbert space (which is closed) with respect to the inner product between two functions, say $(f, g) \in L^2(S)$ (i.e. every function is a class of functions which differs from each other by a constant term). In fact, when $p = 2$ the $W^{k,2}$ subspace becomes a Hilbert space, denoted by H^k, obtained by including all the inner products of all derivatives up to order k. Then any function in H^K can be approximated by other functions whose derivatives exist due to completeness of the Hilbert space. In general, as we will see in the next paragraph a Hilbert space is a real or complex vector space in which the topology is induced by a skew-symmetric positive-definite inner product, and which is complete in the induced metric, such as the l_2 norm.

Hilbert Spaces [3, 164, 174] The general aim behind the time-frequency decomposition (i.e. transform domains) is to find procedures to expand function (i.e. images and signals) over a set of waveforms, selected appropriately among a large and redundant dictionary or basis Ψ. For convenience, we want an image to be interpreted as a representation of a vector in a Hilbert space. This is important in many fundamental signal/image processing tasks such as sampling, reconstruction and minimum mean-square error interpolation and prediction. For this purpose, let's assume a dictionary as a family $\Psi = (\psi_i)_{i \in \Gamma}$ (e.g. time-frequency atoms defined in $\Gamma \subseteq C\text{or}\mathbb{R}$) of vectors in a space H such that $\|\psi_i\|_{l_2} = 1$. Let V be the closed linear span of the dictionary vectors, then finite linear expansions of vectors in Ψ are dense in the space V. We say that the dictionary is complete if and only if $V = H$.

For simplicity in the definitions we assume that the dictionary Ψ of time-frequency atoms described in Section 2.3 is defined in vector format to be in a Hilbert space, such that $H^1 = L^2(\mathbb{R})$, where the domain H can be specified by the l_2 norm with respect to functions on $\mathbb{R}$. In other words, H is a function/measure space defined using the l_2 norm for finite-dimensional vector spaces over a locally compact group, i.e. $\mathbb{R}$. Indeed, this finite linear

[36] Assume that H is a vector space (such as a Hilbert space) and that $M \subset H$ is its non-empty subset. Then M will be a linear manifold of H if for all $x, y \in M$ and any $a, b \in \mathbb{R}$ that $ax + by \in M$, which simply means that M is a linear vector subspace of H [164, 216].

expansion of time-frequency atoms is dense in $L^2(\mathbb{R})$ and hence this dictionary is complete (i.e. any vector in a Hilbert space can be expanded), which establishes a mapping from an image function to a set of numbers.

Usually, in image processing we assume that a space $H = L^2(\mathbb{R})$ is a Hillbert space of complex valued functions (i.e. we will keep here the notation of functions for higher applicability instead of defining an ordered pair of elements u, v as vectors in S as we did earlier). Let a function $f \in H$ expanded over a set of vectors selected from Ψ in order to best match its inner structure. Then we have the following properties:

$$\|f\|_{l_2} = \int_{-\infty}^{+\infty} \|f(t)\|^2 dt < +\infty.$$

The inner product of $(f, g) \in L^2(\mathbb{R})^2$ is defined by (for complex numbers):

$$< f, g >= \int_{-\infty}^{+\infty} f(t)\bar{g}(t)dt,$$

where $\bar{g}(t)$ is the complex conjugate of $g(t)$. Note that we consider a Hilbert space as a vector space over $\mathbb{R}$, equipped with a scalar product (for example denoted $< f, g >$) which is complete for the norm associated with this scalar product (denoted for example $\|f\|_{l_2} = \sqrt{< f, f >}$. In general, according to Riesz theorem, any function f can be represented in a form of inner product (as a generalisation of the scalar product in a vector space) in Hilbert space:

$$< f, \psi >= \int_{-\infty}^{+\infty} f(t)\psi(t)dt, \tag{A.1}$$

where $\psi(t)$ is a function uniquely defined in $L^2(\mathbb{R})$.

The 1D Fourier transform of $f(t) \in L^2(\mathbb{R})$ is written as $\hat{f}(\omega)$ and is defined by:

$$\hat{f}(\omega) = \int_{-\infty}^{+\infty} f(t)e^{-i\omega t}dt.$$

Let $K \subset L^2(\mathbb{R})$ be a convex closed nonempty subset. Then, for all $x \in L^2(\mathbb{R})$ there exists a

unique $x_k \in K$ such that:

$$\|x - x_k\|_{l_2} = \min_{y \in K} \|x - y\|_{l_2}.$$

Equivalently, we have:

$$x_k \in K, \quad < x_k - x, x_k - y > \leq 0 \quad \forall y \in K.$$

We call x_k the orthogonal projection over K of x. If P_K is a projection operator on the convex set K, then a mapping P_K is set as $P_K x = x_k$. Moreover, P_K is continuous and weakly contracting, which means:

$$\|P_K x - P_K y\|_{l_2} \leq \|x - y\|_{l_2} \quad \forall x, y \in L^2(\mathbb{R}).$$

Now, let $L^2(\mathbb{R})$ be a Hilbert space for the scalar product $< . >$ and $(e_n)_{n \geq 1}$ a basis of $L^2(\mathbb{R})$ (i.e countable family of elements which is orthonormal and dense for this product). Then, for every element $x \in L^2(\mathbb{R})$ there exits a unique sequence $(x_n)_{n \geq 1}$ of real numbers such that the partial sum $\sum_{n=1}^{p} = x_n e_n$ converges to x when $p \to \infty$. This sequence is defined as $x_n =< x, e_n >$ and thus we have:

$$\|x\|_{l_2}^2 =< x, x >= \sum_{n \geq 1} | < x, e_n > |^2,$$

which gives:

$$x = \sum_{n \geq 1} < x, e_n > e_n.$$

Therefore, if there exists a countable dense family in a separable Hilbert space $L^2(\mathbb{R})$ then there exists a countable basis of $L^2(\mathbb{R})$. Note that this analysis applies to infinite-dimensional Hilbert spaces as if a Hilbert space was finite then there would be only a finite number of mutually orthogonal vectors and thus we could not have a orthonormal sequence with more than a finite number of terms equal to that dimension. Furthermore, orthonormal sequences are also complete in separable Hilbert spaces as if Hilbert space was non-separable (i.e. there is no countable dense [37] set in the space), then there is no way to accurately represent all

[37] In vector spaces, a subset A of a space $H = L^2(\mathbb{R})$ is called dense in H if for every element $a \in H$, either $a \in A$ or a is a limit point of A (arbitrary close to a member of A). Equivalently A is dense in H if and only if the only closed subset of H containing A is H itself. In metric spaces any operation on any element of A

its elements by an orthonormal sequence. However, note that separability is not a guarantee for completeness of a given orthonormal sequence. Also we need to clarify that an infinite, linearly independent dense set is not necessarily a basis for a Banach, Sobolev or Hilbert space. Such a set is a basis provided that it is orthonormal or more generally orthogonal. In general, there is a one-to-one correspondence (i.e. isomorphism) between separable Hilbert spaces and the space $L^2(\mathbb{R})$ of sequences $(x_1, x_2, \ldots, x_n, \ldots)$ which adopts scalar products. In fact, $L^2(\mathbb{R})$ is indeed an extension of the well-known finite dimensional Euclidean space.

will return another element on A and thus it is closed under any operation (since it converges). For further details see [164, 207, 216].

Appendix B

Definitions and Glossary

Here we summarise some key concepts used throughout this thesis and how these are defined. Most of the notations and definitions presented here are universally adopted notations and definitions. As a general note in matrix equations addition is defined as element-wise, while multiplication is defined in a special way. The product of the $n \times n$ matrix A and the $n \times m$ matrix B is the $n \times m$ matrix whose element in position (i, j) is defined as $\sum_{k=1}^{m} a_{ik} b_{kj}$. A vector is treated as a matrix with one column.

Linear Programming problems [24, 28, 143] Linear programming (LP) problems have both a linear objective function and linear constraints which may include equalities, inequalities or even both. Usually LP problems are stated and analysed in the following standard form:

$$\min_X A^T X, \quad s.t. \;\; CX = Y, \quad X \geq 0,$$

where A and X are vectors in $\mathbb{R}^N$, Y is a vector in $\mathbb{R}^M$ and C is an $M \times N$ matrix. Different variations exist though any linear program can be transformed to this form.

Semi-definite optimisation problems [28, 143, 166] Semi-definite optimisation problem (SDP) is a conic optimisation problem of the form:

$$\min_X C^T X, \quad s.t. \;\; AX = B, \quad X \in K,$$

where $K \in \mathbb{R}_+^N$ is a cone or product of semi-definite cones of a given size, so as:

$$Q^N := \{(X, t) \in \mathbb{R}_+^N : t \geq \|X\|_{l_2}\}$$

defines a second order cone problem, which is a convex optimisation problem and

$$S_+^N := \{X = X^T \succeq 0\}$$

defines a semi-definite cone. In general, there are several variations of conic optimisation problems, with many useful applications.

Interior Point methods [143, 152, 166, 214] Interior Point methods constitute one of the oldest and most successful optimisation. The aim is to derive what we call a fixed point or equilibrium point, which is simply a set of variables satisfying the constraints. As their name implies interior point methods find this point by traversing the interior of the feasible set of solutions defined by the constraints. This is accomplished by going through a number of solutions based on the given symmetric indefinite system of equations together with some heuristics to correct the solutions as the methods proceed to iterations. This type of optimisation methods were known as early as the 1960s in the form of the barrier function methods receiving a great deal of scientific attention with a number of published papers on this area. The polynomial time linear programming algorithm using an interior point method introduced by Karmarkar in 1984 is probably the most well-known method. Interior point methods are also called Gradient-Based methods due to the fact that they are heavily based on spatial and temporal partial derivatives, or related functions, as part of their optimisation technique. Sometimes they are also called Barrier methods since they construct a barrier function that corresponds to the given functions using Lagrange multipliers and derive a solution based on this function. One of the most common method in this category is the Newton's or Newton-Raphson method. This method is based on the Taylor series expansion of a function. Given a function $f(x) = 0$, we can expand it as:

$$f(x + \Delta x) = f(x) + f'(x)\Delta x + \frac{f''(x)}{2!}(\Delta x)^2 + \frac{f^{(3)}(x)}{3!}(\Delta x)^3 + \ldots + \frac{f^{(n)}(x)}{n!}(\Delta x)^n = 0,$$

where $f'(x) = \frac{\partial f(x)}{\partial x}$, $f''(x) = \frac{(\partial)^2 f(x)}{\partial x^2}$ and so on. Ignoring the higher order terms and setting $f(x + \Delta x) = 0$, we derive:

$$\Delta x = \frac{-f(x)}{f'(x)}.$$

Setting $\Delta x^{(k+1)} = x^{(k+1)} - x^{(k)}$ and $\Delta x^{(k+1)} = \frac{-f(x^{(k)})}{f'(x^{(k)})}$ and re-ordering the terms, we obtain:

$$x^{(k+1)} = x^{(k)} - \frac{f(x^{(k)})}{f'(x^{(k)})},$$

which is an iterative procedure of generating solutions; $x^{(k)}$ is the current solution found and $x^{(k+1)}$ is the newly generated solution. note that at each iteration, an approximation model of the problem is created and solved providing the necessary direction for finding the solution of the originally defined problem. The whole process continues till we derive the desired convergence level, which can be expressed by introducing the concept of tolerance or error in our method. For example, the algorithm will stop when $\|f(x)\| \leq \epsilon$ or $\|x^{(k+1)} - x^{(k)}\| \leq \epsilon_1$, where ϵ and ϵ_1 are some small constants defined based on the nature of the problem we deal. Several sparse recovery methods, including Nesta, CVX and l_1 Magic packages are based on interior point methods.

Steepest Descent methods [24, 143, 152, 166] Steepest Descent methods constitute another family of gradient based methods which are very similar to Newton's methods. These methods are also called hill-climbing especially if they are used for finding the maximum of a function. In essence, they try to minimise an objective function $f(x)$ given a current feasible point x, based on the Taylor expansion:

$$f(x^{(n)} + \Delta s) - f(x^{(n)}) = (\nabla f)^T \Delta s < 0,$$

where $x^{(n)}$ and $x^{(n+1)}$ are the current solution at iteration (n) and the new generated solution at iteration $(n+1)$ respectively. $\Delta s = x^{(n+1)} - x^{(n)}$ is the increment factor which is required to be negative. Therefore,

$$\Delta s = -\alpha \nabla f(x^{(n)}),$$

where $\alpha > 0$ is the step size that defines the negative gradient direction of Δs for the search. The choice of proper α highly depends on the nature of the problem. If we want small step

size and slow convergence towards the optimal solution we may choice small value for α while large value of α may achieve quick but yet far from the minimum convergence. Thus, it is advisable to change the value of α at each iteration of the algorithm. In that case the gradient and the step size are calculated at each iteration and therefore the new solution $f(x^{(n+1)})$ is generated as:

$$f(x^{(n+1)}) = f(x^{(n)}) - \alpha^{(n)}(\nabla f(x^{(n)}))^T \nabla f(x^{(n)}),$$

Of course a generation of a good initial solution and a starting step size constitutes another major issue for the successful convergence of the algorithm. Note also that in general this type of method is usually slow in convergence once a local minimisation point is found. This can mainly be attributed to the fact that the gradient, in such point, is very close to zero and thus the convergence to the minimum is very slow.

The Fast Fourier transform [99, 114, 132, 205] A slight variation of Discrete Fourier transform (DFT) is the Fast Fourier transform (FFT) which shares the same properties as the DFT but it is much faster in terms of performance. This computational procedure has been introduced by many researchers, mainly by Cooley, Turkey, Sande, Bendat, Piersol, Bloomfield and Priestley and it is more preferable for long computations and analysis of a set of data. It requires that the value of the terms of a function is compositive and not prime number as this will decrease the number of additions and multiplications. If N is the value of terms, it can be factorised in the form $N = K \times L$, where K, L are integers which can be easily chosen to be less than $N/2$. Furthermore, if N has several small prime numbers as factors, this will reduce even further the necessary computations of the Fourier coefficients. Since the Fourier transform precedes the sampling process, in general, a highly compositive value of N may be chosen before the measurements process starts. Note also that in all the experiments discussed in Chapter (9) the FFT transform has been used instead of DFT, as a build-in function in Matlab.

The Z-transform [114, 136, 205] The Z-transform provides a very concise and characteristic analysis of the frequency domain representations for discrete-time signals where their Fourier transform might not exist or may not satisfy the absolute summable condition $\sum_{t=-\infty}^{+\infty} |f(t)| < +\infty$. Let $f(t)$ be a discrete function defined on $(-\infty, +\infty)$, then the

Z-transform can be defined as:

$$F(z) = \sum_{t=-\infty}^{+\infty} f(t) z^{-t},$$

where $t > 0$ is a small integer, $z = e^s = e^{\sigma + i\omega}$ is a complex variable with real and imaginary parts, where s is a small constant called magnitude of z (i is the imaginary unit and ω is the complex argument, angle or phase in radians). The Z-transform has many useful properties mainly used in probability theory. By substituting $z = e^{\sigma + i\omega}$ in (B) we have:

$$F(z) = \sum_{t=-\infty}^{+\infty} f(t) e^{-(\sigma + i\omega)t} = \sum_{t=-\infty}^{+\infty} [f(t) e^{-\sigma t}] e^{-i\omega t},$$

where $f(t)e^{-\sigma t}$ is calculated using the inverse Fourier transform formulae:

$$f(t) e^{-\sigma t} = \frac{1}{2\pi} \int_0^{2\pi} F(z) e^{i\omega t} d\omega,$$

multiplying both sides with $e^{\sigma t}$ we have:

$$f(t) = \frac{1}{2\pi} \int_0^{2\pi} F(z) e^{(\sigma + i\omega)t} d\omega = \frac{1}{2\pi} \int_0^{2\pi} F(z) z^t d\omega$$

Changing the representation in terms of z instead of ω we have:

$$dz = d(e^{\sigma + i\omega}) = e^{\sigma} i e^{i\omega} d\omega = iz d\omega,$$

or

$$d\omega = \frac{dz}{iz},$$

which finally gives

$$f(t) = \frac{1}{2\pi i} \oint F(z) z^{t-1} dz,$$

where the closed curve integral (defined in a closed path) is calculated with respect to $z = e^{\sigma + i\omega}$ for a fixed value e^{σ} and a varying angle between 0 and 2π. Note that for $\sigma = 0$ ($z = e^{i\omega}$) the Z-transform becomes a general DFT transform discussed in (2.3.2). In general, note that DFT converts a $1D$ signal from time domain to $1D$ frequency domain, while Z-transform converts the same $1D$ signal from time domain to $2D$ complex plane (i.e. z-plane)

represented by $z = e^{\sigma+i\omega}$ for some angle ω. In other words, the real axis of the function's $f(t)$ plane maps one-to-one onto the unit circle in the complex function's $F(z)$ plane.

Laplace transform [32, 114, 136, 164, 205] Laplace transform is a natural extension of Fourier transform which typically is not directly applied to signals/images. However, we include it here for completeness of the discussion of the Unitary transforms and because it is used in many electronic applications (e.g. analogue filters) or even for solving certain types of differential equations. The Laplace transform of a function $f(t)$ which is defined on the set of the non-negative integers is as follows:

$$F(s) = \int_0^{+\infty} f(t)e^{-st}dt,$$

where $s = \sigma + i\omega$ is a complex frequency variable (transform variable) with real (σ) and imaginary parts (ω), $t > 0$ is the time variable and $F(s)$ is a complex-valued function of complex numbers (since we are integrating over t the result is a function of s). Note that we assume $f(t)$ (signal) is defined in positive time and it is absolutely integrable for $0 < f(t) < +\infty$, while $f(t) = 0$ for $-\infty < f(t) < 0$. Note that the right hand side in (B) is finite for all $s \geq 0$, which means for all complex values with non-negative real part. Also, since the upper limit of integration in (B) is infinite, for a given $f(t)$ and s, the Laplace integral may or may not exist (i.e. integral may not converge). However, for most real signals of interest the Laplace transform exists since the integral converges for some complex value of s. In fact it has been proved that the integral in (B) converges when the real part of s exceeds some number called abscissa (a) of convergence. The Laplace inversion operation is given by:

$$f(t) = \frac{1}{2\pi i}\int_{a-i\infty}^{a+i\infty} F(s)e^{st}ds,$$

where a is a real constant so as $\mathbb{R}(s) = a$ (abscissa of convergence) chosen so as $F(s)$ is analytic in region $\Re(s) \geq a$ and $a - i\infty < \sigma + i\omega < a + i\infty$. It is important to mention that the inverse is usually discontinuous at $t = 0$ so that $\lim_{t\to 0+} f(t) \neq (2\pi i)^{-1}\int_{a-i\infty}^{a+i\infty} F(s)$. Standard integral tables for standard functions $f(t)$ can used for the direct transform while the inversion requires specific tables as well. Note also that the Laplace transform is very similar with the Z-transform on replacing the constant e^{-st} by the constant z^{-t}. Among the properties of these transforms is the fact that a convolution in time domain corresponds to

a multiplication in frequency domain (see Appendix for convolution definition). In general, Laplace transform is usually applicable to general case functions $f(t)$ that vanish for negative values of t (e.g. time-like variable) and do not grow faster than e^{ct}, as $t \to \infty$, for some small real constant $c > 0$. Laplace and Fourier transforms are equivalent for discrete time signals defined in positive domains (only for $t > 0$), provided that the real part of the variable s is zero. Its inversion, however, is usually more difficult than that of the Fourier transform, which is thus more preferable when it is applicable. Also, Z-transform is simply a variant of discrete time Laplace transform.

Discrete Cosine transform [182, 205, 210] The Discrete Cosine transform (DCT), very similar to DFT, transforms a signal/image very accurately using a few coefficients, by separating it into smaller parts (i.e. spectral sub-bands) based on it's visual components. The basis functions of the transform are cosines with growing frequencies which do not require complex numbers (real values are suffice for the expansion). The Discrete transform of a discrete function $f(t)$ defined a set of positive values is defined as:

$$f(k) = w(k) \sum_{t=0}^{N-1} f(t) \cos(\frac{\pi k(2t+1)}{2N}),$$

where $k = 0, 1, 2, \ldots, N-1$, N is the signal length and $w(k)$ is a normalisation constant defined:

$$w(k) = \begin{cases} \sqrt{\frac{1}{N}} & k = 0, \\ \sqrt{\frac{2}{N}} & 1 \leq k \leq N-1. \end{cases}$$

The inverse Discrete Cosine transform is:

$$f(t) = \sum_{k=0}^{N-1} w(k) f(k) \cos(\frac{\pi k(2t+1)}{2N}),$$

Note that there are four types of DCT based on the periodicity of the basis frequencies. Here we used the most widely used and simpler one (DCT-2). Also note that a computational efficient approach of DCT exists in an analogous way of FFT, with complexity $O(N \log N)$, where N is the length of signal.

Short Time Fourier transform [27, 114, 132, 182, 186] Window or Short Time Fourier transform (STFT) constitutes a family of transforms related to Fourier transform, which are used to determine the frequency and phase content of local (instead of global) sections of a signal. in fact, a signal is divided into smaller pieces, called windows, assuming that anything outside these windows is periodic. Unlike with the traditional Fourier, in the window Fourier transform all atoms in the basis have a constant scale and are thus mainly localised over a signal interval whose size is proportional to that scale. If a signal structure is localised over a time-scale of a specific order, the expansion (Fourier) coefficients are able to provide us important insights on local and frequency content. This aspect has high applicability in time-scale modifications, speech morphing and signal enhancement [27, 114]. Given a N length window function $w(t-l)$ at time point l and a signal $f(t)$, the STFT is defined as:

$$F(\omega_k, l) = \int_{-\infty}^{+\infty} w(t-l) f(t) e^{-i\omega_k n} dt,$$

where $\omega_k = \frac{2\pi k}{NT}$ is the frequency from DFT, for $k = 1, 2, \ldots, N-1$. Several methods have been proposed to recover a signal from a partial STFT magnitude (For further details see. Note also that a STFT is not well adapted to describe structures that are much smaller or larger than the predefined scale (i.e. window). The smaller the window size the better the time resolution, but the smaller the frequencies represented. In fact, in order to analyse components of various sizes we need to use time-frequency atoms of bases of different scales. This is commonly achieved by using the Wavelet transform, which decomposes signals over time-frequency atoms of varying scales, called wavelets. The wavelet family is built by relating the signal's frequency parameter to the scale, so as the composed decomposition function (i.e. basis Ψ) can efficiently represent the signal's structure. However, note that wavelets fail to provide precise estimates of the frequency content of waveforms whose Fourier transform is well-localised, particularly at high frequencies. This means that Wavelet transform succeeds only in cases where Fourier transform fails. STFT and Wavelet transforms correspond to different families of time-frequency transforms which are frames or orthonormal bases of $L^2(\mathbb{R})$ (i.e. Hilbert spaces, see Appendix). Both STFT and Wavelets have high applicability in cases where we want to modify the spectral (frequency magnitude) of a signal so as to choose the best possible time-domain sequence or to predict the structure of a signal (in remote sensing, X-RAY, crystallography, etc).

Wavelet transform [114, 132, 186, 204, 205] Wavelet transform is actually a more efficient extension of the short time Fourier transform mainly used for saving computing time. It expands the signal by multiplying it with a window function and also performs an orthogonal expansion. Wavelet transform can be defined in a general form for a one dimensional continuous function $f(t) \in L^2(\mathbb{R})$ (space of all square-integrable functions) as follows:

$$c(a,b) = \int_{\mathbb{R}} f(t)\Psi^*_{a,b}(t)dt,$$

with $a \in \mathbb{R}^+$ and $b \in \mathbb{R}$ represent the scale and translation respectively (shifted and scaled version) of the wavelet $c(a,b)$ (sometimes also called coefficient of function $f(t)$). Ψ^* denotes the complex conjugate of the basis functions Ψ, called mother-analysing wavelet (as there are many variations) defined as:

$$\Psi_{a,b}(t) = \frac{1}{\sqrt{a}}\Psi(\frac{t-b}{a}),$$

where $\frac{1}{\sqrt{a}}$ as a normalisation parameter in terms of scale. Note that the basis provides both localisation in space and multiple scales so as to create a more compact representation, which has many useful applications particularly in data/image compression, noise filtering, data smoothing, periodicity detection and feature detection in images. Also note from the previous Equation (B) that the Wavelet transform of $1D$ vector (signal) is actually $2D$ and thus the transform of an image is a four dimensional (combinations of multiple horizontal and vertical operations). Now given the wavelet transform $c(a,b)$ of a given function $f(t)$, the generalised inverse Wavelet transform is simply a synthesis from its coefficients:

$$f(t) = \frac{1}{w}\int_0^{+\infty}\int_{-\infty}^{+\infty}\frac{1}{\sqrt{a}}c(a,b)x(\frac{t-b}{a})\frac{dadb}{a^2},$$

where w is a finite constant implying that $\Psi_{a,b}(0) = 0$, calculated as:

$$w = \int_0^{+\infty}\frac{\Psi^*_{a,b}(t)x(t)}{t}dt = \int_{-\infty}^{0}\frac{\Psi^*_{a,b}(t)x(t)}{t}dt,$$

where usually $x(t) = \Psi^*_{a,b}(t)$, but some other non-negative functions can also be used instead of a wavelet function. One of the simplest and oldest mother wavelet is the Haar Wavelet

given by the following translated and scaled functions:

$$\Psi_{a,b}(t) = 2^{a/2}\Psi(2^a t - b), \quad b = 0, \ldots, 2^a - 1 \tag{B.1}$$

$$\Psi(t) = \begin{cases} 1 & \text{for } 0 \leq t < \frac{1}{2}, \\ -1 & \text{for } \frac{1}{2} \leq t < 1, \\ 0 & \text{otherwise.} \end{cases}$$

Other examples of wavelet functions are the Morlet's wavelet $\Psi(t) = e^{-2\pi^2(t-t_0)^2}$ for $t_0 \approx 0.8$ a small constant and the Mexican hat, $\Psi(t) = (1 - t^2)e^{-t^2/2}$. In general, wavelets form a collection of functions that form an orthonormal basis for $L^2(\mathbb{R})$, which never overlaps, since $< \Psi_{a,b}, \Psi_{a,b'} >= \delta_{b,b'}$, unless two wavelets have different scales, for example $\Psi_{1,2}(t)$ and $\Psi_{3,1}(t)$.Another important property of wavelets is the decay growth of their coefficients, $|\Psi_{a,b}(t)| \geq 2^{-k}$, for $t = 0, 1, \ldots$ and some $k > 0$, aspect which implies sparsity. As with the previous transforms a more time efficient variant of the discrete wavelet transform has been introduced by Mallat in 1989 [132]. Note also that the famous JPEG-2000 compression method is also based on Wavelets. In particular, JPEG-2000 standard incorporates the use of multiple wavelets and user-defined wavelet transforms, including biorthogonal (e.g. Daubechies and Le Gall; see Appendix) basis functions so as to lower complexity and achieve highest compression. These wavelet transform options also provide lower resolution and spatial decorrelation (i.e. remove spatial redundancy) of the images, so as to enhance the storage and transmission of digital images. It is not surprising that Wavelets have been used in a variety of application, including video compression, internet communications compression, object recognition, filter design or even in numerical analysis. A more detailed analysis about JPEG-2000 standard can be found in [182], while a useful introduction about modern image compression and coding techniques for image decomposition and decorrelation can be found in [204, 205].

Matrix Norms [3, 65, 164] In general, a norm on a vector space $\mathbb{R}^N$ is a function $\|.\| : \mathbb{R}^N \to \mathbb{R}^+$, such that for any $x, y \in \mathbb{R}^N$ and $\lambda \in \mathbb{R}$:
1) $\|x\| = 0$ if and only if $x = 0$,
2) $\|\lambda x\| = |\lambda| * \|x\|$,
3) $\|x + y\| \leq \|x\| + \|y\|$.

The squared Frobenius norm of a matrix $A_{m\times n}$ is defined as:

$$\|A\|_{F^2} = \sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij}^2 = \text{trace}(A^T A).$$

The l_p-norm ($1 \leq p < \infty$) of a matrix $A_{m\times n}$ is defined as:

$$\|A\|_{l_p} = \sum_{i=1}^{m}\sum_{j=1}^{n} (A_{ij}^p)^{1/p},$$

while for $p = \infty$, we have:

$$\|A\|_\infty = \max_{i=1,\ldots,n;j=1,\ldots,m} \|A_{ij}\|.$$

Note that for $1 \leq p \leq \infty$ we have a real norm while in the case of $0 < p < 1$ we have a quasi-norm which represents the number of the non-zero entries of a matrix. Sometimes, in the literature, we also write the l_p as l_p^N referring to real or complex vectors. For example, the unit ball (i.e. the interior of a sphere of radius 1) in l_p^N is defined as:

$$B_p^N = B(r = 1) = x \in \mathbb{C}^N, \|x\|_{l_p} \leq 1.$$

The associated l_p metric, usually denoted as d_p or d_{l_p}, is defined as:

$$d_{l_p} = \|x - y\|_{l_p},$$

for all $x, y \in \mathbb{R}^N$. The subordinate norm, denoted as $\|.\|$ on $\mathbb{R}$ (or on $\mathbb{C}$), of a matrix $A_{n\times n}$ is defined as:

$$\|A\| = \sup_{x\in\mathbb{R}^n, x\neq 0} \frac{\|Ax\|}{\|x\|}.$$

We say that the condition number of a matrix $A_{n\times n}$, relative to the subordinate matrix norm, is defined as:

$$\text{cond}(A) = \|A\| \times \|A^{-1}\|.$$

This definition helps us define the errors in the data (right-hand side or matrix) which might occur. We also say that a matrix is well-conditioned if it's condition number is close to 1 (its minimal value) and it is ill-conditioned if its condition number is large. This is important particularly in solving a linear system for an ill-conditioned matrix.

Vector Spaces [15, 216] Let S be a set where the operations of addition $+$ (mapping $S \times S \to S$) and scalar multiplication $< . >$ (R (or C) $\times S \to S$) are defined. This set is called vector space if the following rules are satisfied for all $x, y, z \in S$ and $k_1, k_2 \in C$(or $\mathbb{R}$):
1) $x + y = y + x$,
2) $x + (y + z) = (x + y) + z$,
3) $x + 0 = x$, where 0 is a unique zero vector,
4) $x + (-x) = 0$, where $-x$ is a unique vector for x,
5) $1x = x$, where 1 is a unique vector,
6) $(k_1 k_2)x = k_1(k_2 x)$,
7) $k_1(x + y) = k_1 x + k_1 y$,
8) $(k_1 + k_2)y = k_1 y + k_2 y$.
Note that the Euclidean space $\mathbb{R}^N$ is a real vector space.

Subspaces [15, 216] A subspace H of a vector space S is simply a subset of a set S with the following three properties:
1) The zero vector of S is also in S.
2) For each x and y in H, $x + y$ is also in H (H is closed in vector addition).
3) For each x in H and a scalar c in H, then cx is also in H (H is closed under scalar multiplication).

Inner product [15, 216] Inner product or scalar product on a complex vector space S is the mapping $< . >: S \times S \to C$ such that for all $x, y, z \in S$ and $k \in C$ we have:
1) Conjugate symmetry: $< x, y >=< y, \bar{x} >$,
2) Linearity: $< kx, y >= k < x, y >$ and $< x + z, y >=< x, y > + < z, y >$,
3) Positive-definiteness: $< x, x >> 0$ if $x \neq 0$.

Metric space [15, 216] A metric space is a vector space with a metric. For example, let (S, d) or $(d(S))$ be a metric space. Then the metric (mapping) $d(.) : S \times S \to [0, \infty)$ satisfies the following properties for $x, y \in S$:
1) $d(x, y) = d(y, x)$,
2) $d(x, y) > 0$, if $x \neq y$,
3) $d(x, x) = 0$,
4) $d(x, y) \neq d(x, z) + d(z, y)$, (i.e. triangular inequality)

Note that any space with a metric is a metric space. However, a metric space (e.g. $S^z, z < \infty$) is complete only if every Cauchy sequence in this vector space converges to a limit. This means that if (x_n) is a sequence of elements of a metric space S with metric $d(.)$ then for every $\epsilon > 0$ there exists an integer r such that $k, l \geq r$ and $d(x_k, x_l) < \epsilon$ (i.e. Cauchy sequence). Hilbert space ($L^2(\mathbb{R})$) is a metric space, where the vector space is $\mathbb{R}$ and the metric is the l_2 norm. Also $\mathbb{R}$ is a complete metric space with respect to its natural metric.

Norms on a vector space [15, 215, 216] The l_p norm for $p > 0$ and $p = \infty$ on a vector space $L_p^M(a, b)$ can be defined as:

$$\|f(x)\|_{l_2} = \left(\int_a^b (f^{(M)}(x))^p dx \right)^{1/p} < \infty,$$

$$\|f(x)\|_{l_\infty} = \sup \int_a^b |f^{(M)}(x) dx| < \infty,$$

where (a, b) is the measure space, p is the power of integral, M represents the countable finite dimensions of vector space, while $f(x)$ denotes the measurable function from (a, b) to C or $\mathbb{R}$ whose absolute value to the p-th power has a finite integral. Note that any space $L^2(\mathbb{R})$ of measurable real functions $f(x), g(x)$ satisfying $\int f(x)^2 dx < \infty$, with inner product $< f, g >= \int f(x)g(x)dx$, associated norm $\|f\|_{l_2} = \sqrt{< f, f >}$ and metric norm $\|f - g\|_{l_2}$ is a Hilbert space.

Projection [3, 65, 164] For a subspace $V \subseteq \mathbb{R}^n$ and a vector $u \in \mathbb{R}^n$, let $\pi_v(u)$ be a projection of u onto subspace V . If $\{v_1, \ldots, v_k\}$ is a basis for V, then

$$\pi_v(u) = \sum_{i=1}^{k} < v_i, u > .$$

Orthonormal and Biorthogonal Basis [33, 215, 216] In a vector space, i.e. $L^2(\mathbb{R})$, assume an orthonormal (Unitary) basis $C = \{C_1, C_2, \ldots, C_i\}$ with $< C_i, C_j >= \delta_{ij}$, for $i \neq j$, then every function $f \in L^2(\mathbb{R})$ ($f : \mathbb{R} \to \mathbb{R}$) can be expanded as:

$$f = \sum_i < f, C_i >,$$

and

$$\|f\|^2_{L^2(\mathbb{R})} = \sum_{i=1}^{\infty} \| < f, C_i > \|_{l_2} < +\infty$$

In the same vector space, i.e. $L^2(\mathbb{R})$, assume now two bases $C = \{C_1, C_2, \ldots, C_i\}$ and $G = \{G_1, G_2, \ldots, G_j\}$ with $< C_i, G_j >= \delta_{ij}$. Then any function $f \in L^2(\mathbb{R})$ ($f : \mathbb{R} \to \mathbb{R}$) can be expanded using both bases. It can be decomposed by one and reconstructed by another (the order does not matter) as:

$$f = \sum_i \sum_j < f, C_i > G_j = \sum_i \sum_j < f, G_j > C_i$$

This is a common case in Wavelets where the mother wavelet is composed of two bases; one for the analysis and the other one for the synthesis.

Mutual Coherence [39, 41, 79] The mutual coherence of N by N orthonormal (sparsity) basis Ψ and M by N Sensing (measurement) matrix Φ with unit norm columns is the maximum absolute value of the inner product between the elements of columns of two matrices:

$$\mu(C) = \mu(\Phi, \Psi) = \max_{1 \le i,j \le N} | < \Phi_i, \Psi_j > | = \max_{1 \le i \ne j \le N} | < C_i, C_j > |,$$

where the matrix $C = \Phi\Psi$ maps an object from Ψ to Φ domain. The values of $\mu(C)$ are in the range of $[\frac{1}{\sqrt{N}}, 1]$ where the lower bound is for the DFT basis while the upper bound is given for the canonical or identity basis. Note also that the greater the incoherence, the smaller the number of measurements needed.

Sparsity [30, 132, 186, 215] In many real practical problems signals or images of interest are not exactly sparse. Using a dictionary Ψ or in general by changing the domain of representation (usually to frequency domain), signals or images can be represented exactly as a superposition of a small fraction of atoms and thus become sparse or compressible. Note that there are some functions (signals) where the Fourier transform cannot be found directly by evaluating the Fourier integral. These types of functions as not sparse as they are not absolutely convergent to zero and thus their Fourier integral cannot be evaluated. In particular, sinc filter as a function has large support and is slow to compute. The Fourier integral of this function is periodic and does not decrease to zero as frequency approaches to infinity. Sinc as a signal can be defined as $\sin(x)/x$ or $\sin(\pi x)/\pi x$. It's indefinite integral does

not converge $\int_{-\infty}^{+\infty} |\sin(\pi x)/\pi x| = +\infty$ and thus it is not sparse. Usually, these functions can be used as filters to remove random noise or as interpolating functions, to recover a function/signal by interpolating its samples. For further analysis on these types of functions and mathematical ways to overcome this problem see [30].

Least squares K-terms approximation [28, 33, 132] Consider a discrete time signal as a function of time (t), defined as $F(t)$ on a specific time interval, say $[a, b]$ and a set of sampled values of N length, $F_1, F_2, \ldots, F_N$ with $F_N = F(t_N)$ and $N \in [a, b]$. We seek to find an approximate representation of $F(t)$, defined $\hat{F}(t)$, as a linear combination of its sparsifying basis functions $\Psi_i(t)$, $i = \{1, 2, \ldots, K\} \subset N$ (e.g. Fourier dictionary). Then we have:

$$\hat{F}(W, t) = \sum_{i=1}^{K} W_i \Psi_i(t), \quad \text{for} \quad a \leq t \leq b,$$

where W_i is the coefficients vector found so as to minimise the objective:

$$\min E^2(W) = \sum_{i=1}^{N} (F_i - \hat{F}(W, t_N))^2,$$

with analytical solution (in vector notation),

$$E^2(W) = E^T E = [F - W\Psi]^T [F - W\Psi] = F^T F - F^T W\Psi - W^T \Psi^T F + W^T \Psi^T W \Psi,$$

which gives:

$$E^2(W) = F^T [I - \Psi[\Psi^T \Psi]^{-1} \Psi^T] F + [\Psi^T W \Psi - \Psi^T F]^T [\Psi^T \Psi]^{-1} [\Psi^T W \Psi - \Psi^T F],$$

keeping only the second terms which depends on W, the objective now becomes:

$$\Psi^T W \Psi - \Psi^T F = 0; \quad \text{or} \quad W = [\Psi^T \Psi]^{-1} \Psi^T F.$$

Interpolation [90, 163, 174] Interpolation refers to the process of recovering a signal (i.e. derive an analytic expression for the signal) by forming its corresponding function $f(x)$ based on a particular finite set of points or sample data that have been collected using various digital filtering techniques. In analogue reconstruction filters interpolation is used

to increase a low sampling frequency to a higher one. In order to increase the number of samples, new samples are generated by interpolating between pairs of given samples, using a linear interpolator or oversampling filter. In a more broader mathematical framework, the aim is to create the curve which best represents the measurement points (sample values) so as the content of the original signal will not be distorted. In finite dimensional vector spaces, e.g. $L^2(a,b)$, optimal interpolation formulae are also important as ways to decrease the norm of the error functional between the data set and the interpolant. In general, a function $f(x)$ whose values are known only on a small set of points (interpolation nodes) x_k, for $k = 1, 2, \ldots, N$ in an interval (a,b) can be approximately defined (i.e. interpolated) as:

$$\min_x \int_a^b (f(x)^2) dx,$$

in the measure space (a,b) using the l_2 norm and with the aim to minimise the set of interpolating functions in the space $L^2(a,b)$. Note that in cases where the system of equations formulating the samples is ill-conditioned, no numerical interpolation method can provide an accurate solution and thus the coefficients of the interpolating polynomial cannot be found as a system of linear equations using standard techniques of linear programming. Note also that several interpolating techniques have been proposed in the literature for this purpose such as the construction of the function using polynomials, quotients of polynomials (rational functions), sinusoids and cosines (trigonometric functions) or even Fourier based functions. Some of these methods are very efficient and they are based on the choice of an appropriate algorithm. It is notable that they all have high applicability in the image processing and signal processing areas, where the purpose is to analyse the colour of the pixels or value of signal coefficients and use that information to determine the colour/coefficient values of the new pixels/coefficients in an image/signal we want to process (enlarge an image or enhance a signal).

Convolution [99, 111, 136, 205] Convolution is a mathematical way of combining two functions together (i.e. signals or images). It is very important in Digital Signal Processing, as two signals are combined to form a third one, but it also has high applicability in low-pass and high-pass filtering for properly selecting components of a signal. In general, it can be

defined in the one dimension continuous space as follows:

$$f(x) = \int_{-\infty}^{+\infty} g(u)h(u-x)du,$$

which is simply a multiplication of elements (e.g. pixels) together shifted to position x; or in two dimensions,

$$f(x,y) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} g(u,v)h(u-x,v-y)dudv,$$

where function $h(.)$ is sometimes called filter, convolution mask or convolution kernel, especially if it is defined in only a small subset of a domain (e.g. neighbohoud of pixels). This has a particular application in Unitary transforms and adaptive structure filtering, where we have to define the window of application. In discrete space the two dimension convolution summation is defined as:

$$f(x,y) = \sum_{u,v\in\Omega}\sum g(u,v)h(u-x,v-y),$$

where Ω is the domain where $g(.)$ and $h(.)$ are defined. In Image Processing, the linear cyclic convolution can also be defined as a general mathematical operation between two functions as follows:

$$e = f \star h,$$

where f represents an image (2-D function) and $h = \{h_{k,l}\}$ represents a 2-D window, assuming that images are in general represented as 2D discrete signals (vectors). More precisely the linear convolution of every point (n,m) (i.e. pixel) is defined as:

$$e_{n,m} = (f \star h)_{n,m} = \sum_k \sum_l f_{n-k,m-l}h_{k,l}, \quad n,m = 0:255,$$

with indices $n-k$ and $m-l$ are taken modulo $N = 256$ (i.e. for a 256×256 pixels image). In the simplest case where h is a 3×3 window defined, for example, as a matrix

$$\|h_{k,l}\| = \frac{1}{3}\begin{bmatrix} 0 & 1 & 1 \\ 1 & 3 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

with indices k and l take values $3, 0, 1$. Then the convolved image f with window h is calculated as:

$$e_{n,m} = \frac{1}{3}[3f_{n,m} + f_{n,m-1} + f_{n-1,m} + f_{n,m+1} + f_{n+1,m}] \quad n, m = 2 : 255.$$

In particular, the window is positioned with the centre at point (n, m) together with the values of the image while the four nearest points are weighted based on the previous rule. Convolution is used in the calculation of the Discrete Fourier transform (DFT) of an image/signal though for ease in calculations we usually vectorise (i.e. transform from $2D$ to $1D$) the image matrices (e.g. in Fast Fourier transform). Note that the Fourier transform of the convolution of two functions is equal to the product of their individual Fourier transforms, while the order of the convolution of two functions does not affect the final outcome.

Another common similar operation on images apart from linear convolution and application of a $1D/2D$ Unitary transform is the process of the linear combination of two images. Assume two 256 grey level images f and g:

$$f = f_{m,n}; \quad m, n = 0 : 255 \quad \text{and} \quad g = g_{m,n}; \quad m, n = 0 : 255$$

then we have:

$$c = kf + lg,$$

where $k, l \in \mathbb{R}$ are real numbers used for the combination of two images together.

Image sampling [111, 182, 212] In image processing images can be represented as superpositions of point spread functions (i.e. Dirac delta functions δ). Consider a continuous image function represented as $f(x, y)$ sampled at a grid of discrete points $x = i\Delta x$ and $y = j\Delta y$, where $i, j = 1, 2, \ldots, N$ are the indexes of the sampling points of an $N \times N$ im-

age and distances Δx, Δy are the sampling intervals (i.e. distances between two sampling points). The sampled image $f_s(x, y)$ as the outcome of the sampling process can be derived as:

$$f_s(x, y) = f(x, y) \sum_{i=1}^{N} \sum_{j=1}^{N} \delta(x - i\Delta x, y - j\Delta y).$$

In ordinary sampling schemes, the sampling interval has to be chosen in size so as it is less than half of the smallest area of interest, in terms of detail, for efficient recovery. Note also that the same sampling process can also be applied in the frequency domain.

Image Inpainting [14, 103, 181] The process of reconstructing lost or deteriorated parts of images, usually includes restoring a selected (missing or damaged) region in an image using the pixel region neighborhood, through image interpolation, which is the notion of inpainting. Note that it was only till recently with the papers of Sapiro et al. where digital inpainting was first introduced together with a detailed description of its motivations and its numerous applications. In these papers Sapiro et al. introduced a novel algorithm for modifying images, which attempts to automatically replicate destroyed areas of an image based on the structure of their surrounding regions and boundaries without any user intervention. No assumptions on the topology of the region to be inpainted or on the structure of the image are made as the main idea of the algorithm is to smoothly propagate information from the surrounding areas (extend the gray-levels at the boundary of the damaged area) so as to gradually rebuilt the area that needs inpainting. A more mathematical description of the Digital Inpainting problem can be given as:

Definition [14, 172, 181]: *Let Ω denote the complete image domain, or more generally, a finite domain in $\mathbb{R}^2$. Due to different factors, such as object occlusion, packet losses in communication, etc. there is $G \subset \Omega$ where image data are missing. Let also θG be the boundary of region G which actually needs to be inpainted. Intuitively, the aim is to recover the original image X on the entire domain Ω based on the partial measurements $X'|_{\Omega/G}$ and using θG.*

There are several sophisticated digital models, techniques, and methods for replacing lost or corrupted parts of the image data. Usually, they are based on geometric approaches for filling in the missing information in the region which should be inpainted. Alternatively, textural

inpainting is also used for restoration. Indeed, this field of research has been very active over the recent years boosted by numerous objectives and applications, including image interpolation, photo restoration (e.g. repair old photos), zooming and super-resolution (e.g. remove watermarks and erase power lines), primal-sketch based perceptual image compression and coding, error concealment of (wireless) image transmission (loss concealment), etc.

Set Covering problem [62, 211] The aim is to select the minimum number of the given sets so that all the elements contained are any of these sets. In particular, given an instance (X, F) where X is a finite set and F represents a set of subsets of X, we need to find the smallest one of the subsets in F. We want to determine the minimum sized subset of F which covers all the elements of X; that is to find $C \subseteq F$ so as:

$$X = \underset{S \in C}{\cup} S \quad \text{and} \quad |C| \text{ minimal}$$

Equivalently, as a linear programming problem it can be defined as follows:

$$\min_{x_i} \sum_{i=1} \{N\} x_i \quad \text{s.t.} \quad \sum_{i|j \in S_i} x_i \geq 1,$$

for all $j \in \{1, 2, \ldots, M\}$ and $0 \leq x_i \leq 1$ for all $i \in \{1, 2, \ldots, N\}$. We assume a family of N subsets $(S_1, S_2, \ldots, S_N)$ of $\{1, 2, \ldots, M\}$. Also note that Set Covering problem has many practical and useful applications though it belongs to the category of one of Kurp's NP-Complete problems being proven in 1972 (reduction to the minimal vertex cover problem). It's dual is the Set Packing problem.

Useful Glossary-Abbreviations[32, 99, 136, 157, 182, 186, 205]:

Analog-to-Digital Converter (ADC): Converts analog voltage into digital numbers.
Aliasing: Unwanted replications caused by sub-sampling, which distort the recovery.
Anti-Aliasing Filter: A filter used to limit the bandwidth of any signal.
Anti-aliasing and smoothing filter: An anti-aliasing filter limits the bandwidth of a signal, while a smoothing filter is used on the output of a digital processing system to smooth the sampling pattern of the system.
Banded: If the non-zero elements of a sparse matrix are near to the diagonal, this matrix has a banded structure.
Bandlimited signal: A signal whose Fourier transform has only finite support.
Cardinality: A metric for measuring the number of non-zero entries of a vector. In a vector space cardinality measures its dimension or simply the number of vectors.
Cycle: A complete sequence of signal indications.
Cycle length: The total time for a signal to complete one cycle length.
Deconvolution: The inversion of convolution; solving h from $e = f \star h$, where e, f are given and $\star$ is the convolution operation. This is an ill-posed inverse problem with many applications and filters.
Digitization: The conversion of a continuous time or spatially continuous distribution (brightness function) to a 1D or 2D array of integers.
Discrete Cosine transform (DCT): Similar to DFT, a Mathematical transform which uses only the sum of cosine functions (basis) to express a sequence of data.
Discrete Fourier transform (DFT): A mathematical technique for computing the transform (spectrum) of a signal/image from the time/spatial domain, using sinusoids as basis.
Fourier series: Any continuous function produced as an infinite sum of sine or cosine waves.
Fast Fourier transform (FFT): A computational efficient version of DFT based on eliminating redundant computations found in DFT.
Fundamental frequency: Lowest frequency of periodic signal (as a superposition of sinusoids, Fourier series, etc.). Sometimes also called first harmonic of a signal (i.e. first repetition of a waveform).
Harmonics: Multiples of fundamental frequency in Fourier series representation, where the frequencies $\omega_p = 2\pi p/N$ for $p = 1, 2, \ldots, N/2$ (N is signal length) are integer multiples of

fundamental frequency (i.e. n-th harmonic $= n\times$ fundamental frequency).

Hilbert space: A complete inner product space with norms (i.e. infinite dimensional space).

Interval: A discrete portion of the signal cycle during which the signal indications remain unchanged.

Laplace transform: It is a natural extension of Fourier transform, mainly used in analogue filters.

Low pass filter: A filter which can suppress values which are larger than a certain threshold (e.g. remove additive noise).

Nyquist frequency: The highest frequency necessary for efficient sampling and recovery.

Offset: The time relationship in seconds or percent of cycle length.

Orthogonal: A transform matrix is orthogonal if its rows or columns are orthogonal unit vectors (perpendicular to each other).

Period T: Reciprocal to the frequency f, it measures the number of time units between successive peaks. For multiples of the same period T the value of the signal always remains the same; a fundamental component in sampling theory.

Periodic: A function $f(t)$ with property $f(t+nT) = f(t),\ \forall n \in Z$ and $T = 2\pi/\omega$ is the period, or $\omega = 2\pi/T$ is the fundamental frequency.

Quantization: Application of a thresholding process at each sample so as to decrease the number of integers representing the different levels of brightness in an image.

Sampling: The evaluation of a function of discrete values on an indexed matrix of samples.

Shannon-Nyquist Theorem: States that a sampling rate greater than twice the frequency of the highest component is needed to efficiently reproduce a signal.

Sparse: A matrix or vector which has many zero elements (entries).

Split: The percentage of a cycle length allocated to each of various phases in a signal cycle.

f or ω: Frequency/angular frequency which refers to the number of cycles per unit time.