ROBUST COMPRESSIVE SENSING TECHNIQUES

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND

ELECTRONICS ENGINEERING

AND THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

By

Oğuzhan Teke

July, 2014

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr. Orhan Arıkan(Advisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr. Ahmet Enis Çetin

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Ali Cafer Gürbüz

Approved for the Graduate School of Engineering and Science:

Prof. Dr. Levent Onural
Director of the Graduate School

ABSTRACT

ROBUST COMPRESSIVE SENSING TECHNIQUES

Oğuzhan Teke
M.S. in Electrical and Electronics Engineering
Supervisor: Prof. Dr. Orhan Arıkan
July, 2014

Compressive Sensing theory details how a sparsely represented signal in a known basis can be reconstructed from an underdetermined linear measurements. However, in reality there is a mismatch between the assumed and the actual dictionary due to factors such as discretization of the parameter space defining basis components, sampling jitter in A/D conversion, and model errors. Due to this mismatch, a signal may not be sparse in the assumed basis, which causes significant performance degradation in sparse reconstruction algorithms. To eliminate the mismatch problem, this thesis presents two novel robust algorithm and an adaptive discretization framework that can obtain successful sparse representations. In the proposed techniques, the selected dictionary atoms are perturbed towards directions to decrease the orthogonal residual norm. The first algorithm named as Parameter Perturbed Orthogonal Matching Pursuit (PPOMP) targets the off-grid problem and the parameters of the selected dictionary atoms are perturbed. The second algorithm named as Perturbed Orthogonal Matching Pursuit (POMP) targets the unstructured basis mismatch problem and performs controlled rotation based perturbation of selected dictionary atoms. Based on detailed mathematical analysis, conditions for successful reconstruction are derived. Simulations show that robust results with much smaller reconstruction errors in the case of both parametric and unstructured basis mismatch problem can be obtained as compared to standard sparse reconstruction techniques. Different from the proposed perturbation approaches, the proposed adaptive framework discretizes the continuous parameter space depending on the estimated sparsity level. Once a provisional solution is obtained with a sparse solver, the framework recursively splits the problem into sparser sub-problems so that each sub-problem is exposed to less severe off-grid problem. In the presented recursive framework, any sparse reconstruction technique can be used. As illustrated over commonly used applications, the error in the estimated parameters of sparse signal components almost achieve the Cramér-Rao lower bound in the proposed framework.

Keywords: Compressive Sensing, Basis Mismatch, Off-grid, Basis Perturbation, Perturbed OMP, DelayDoppler Radar, Sampling Jitter, Sparse Reconstruction, Recursive Solver, Adaptive Discretization.

ÖZET

GÜRBÜZ SIKIŞTIRILMIŞ ALGILAMA TEKNİKLERİ

Oğuzhan Teke
Elektrik ve Elektronik Mühendisliği, Yüksek Lisans
Tez Yöneticisi: Prof. Dr. Orhan Arıkan
Temmuz, 2014

Sıkıştırılmış Algılama teorisi, eksik belirtilmiş doğrusal gözlemlerden, bilinen bir tabanda seyrek olan bir sinyalin nasıl geri çatılacağını inceler. Fakat gerçekte, parametre uzayının seyrekleştirilmesi, analog-sayısal çeviricilerdeki örnekleme sapması veya modelleme hatası gibi sebepler yüzünden varsayılan ile asıl taban arasında uyumsuzluk vardır. Bu uyumsuzluk sebebiyle verilen bir sinyal varsayılan tabanda seyrek olmayabilir. Bu da geri çatım yöntemlerinin başarımını ciddi bir şekilde düşürür. Bu tezde, taban uyumsuzluğu problemini ortadan kaldırmak için, başarılı seyrek ifadeler elde edebilen özgün iki gürbüz algoritma ve bir değişken seyrekleştirme yapısı sunulmuştur. Önerilen tekniklerde seçilmiş taban öğeleri, dik kalan vektörünü azaltacak şekilde uyarlanmışlardır. Parametre Uyarlamalı Dikey Eşleyen Takip (PPOMP) isimli ilk algoritma ızgara-dışılık probleminin çözümünü hedefler ve seçilmiş taban öğelerinin parametrelerini uyarlar. Uyarlamalı Dikey Eşleyen Takip (POMP) isimli önerilen ikinci algoritma ise yapısal olmayan taban uyumsuzluğu problemini hedefler ve seçilmiş taban öğelerine döndürme tabanlı kontrollü bir uyarlama uygular. Detaylı matematiksel analizlere dayanılarak başarılı geri çatım için şartlar türetilmiştir. Benzetim çalışmaları, standart yöntemlere kıyasla hem yapısal hem de yapısal olmayan taban uyumsuzluğu problemlerinde gürbüz bir şekilde çok küçük geri çatım hataları elde edilebildiğini göstermiştir. Önerilen uyarlama tekniklerinden farklı olarak, önerilen değişken yapı, kestirilen seyreklik seviyesine bağlı bir şekilde sürekli parametre uzayını ayrıklaştırır. Herhangi bir seyrek çözücü ile öncül bir çözüm elde edildikten sonra yapı, özyineli bir şekilde ana problemi daha seyrek alt problemlere ayırır. Bu sayade her alt problem daha az etkili bir ızgara dışılık problemine maruz kalır. Önerilen özyineli yapıda herhangi bir seyrek geriçatım tekniği kullanılabilir. Yaygın bir şekilde kullanılan sıkıştırılmış algılama uygulamalarında gösterildiği üzere, önerilen yapıda kestirilen parametre hatası hemen hemen Cramér-Rao alt sınırına ulaşmıştır.

Anahtar sözcükler: Sıkıştırılmış Algılama, Taban Uyumsuzluğu, Izgara Dışılık, Taban Uyarlama, Darbe-Doppler Radarı, Örnekleme Sapması, Seyrek Geriçatım, Özyineli Çözüm, Değişken Ayrıklaştırma.

Acknowledgement

I am grateful to my advisor Prof. Orhan Arıkan for his guidance and support during the whole period of this work. His intuitions and perspective in problem solving have enabled me to understand and cope with many academic problems with ease. Due to his constant urge to teach, whether it is technical or non-technical, I have always learned something new from him. It was a great pleasure and an honor to study with him.

I am heartily thankful to Prof. Ali Cafer Gürbüz for his sincere collaboration during this study. His undeniable helps have shown me new dimensions in the problems and perfected this study.

I would also like to thank Prof. Enis Çetin for his careful reading of this thesis and helpful suggestions.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Sparse signal representations and the compressive sensing (CS) theory [1, 2] has received considerable attention in recent years in many research communities. In particular, CS changed the way data is acquired by significantly reducing the number of data samples. As a new technique with a significant promise, CS has been applied to a wide range of important applications, such as computational photography [3], medical imaging [4], radar [5, 6], and sensor networks [7].

Compressive sensing states that a sparse signal in some known basis can be efficiently acquired using a small set of nonadaptive and linear measurements. Consider an N dimensional signal $\boldsymbol{s}$ that has a K-sparse representation in a transform domain $\boldsymbol{\Psi}$, as $\boldsymbol{s} = \boldsymbol{\Psi}\boldsymbol{x}$ and $\|\boldsymbol{x}\|_0 = K$. Given linear measurements in the form $\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{s}$, by using compressive sensing techniques, the sparse signal $\boldsymbol{x}$, hence $\boldsymbol{s}$, can be recovered exactly with very high probability from $O(K \log N)$ measurements by solving a convex ℓ_1 optimization problem of the following form:

$$\min \|\boldsymbol{x}\|_1, \qquad \text{subject to} \quad \boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{\Psi}\boldsymbol{x}, \tag{1.1}$$

which can be solved efficiently using linear programming. Stable reconstruction methods for noisy measurements or compressible signals based on ℓ_1 minimization have been developed [8–10] for the known basis case. Suboptimal greedy algorithms have also been used in many applications. Matching pursuit (MP) [11], orthogonal matching pursuit (OMP) [12], compressive sampling matching pursuit (CoSaMP) [13], iterative hard/soft thresholding (IHT) [14] are among the most commonly used greedy algorithms. Apart from greedy algorithms, approximate message passing (AMP) uses the idea of belief propagation to achieve high reconstruction performance with low complexity [15]. If the sparse signal $\boldsymbol{s}$ has a structure, such as a wavelet tree, techniques proposed in [16] can exploit those models for better reconstruction. The study in [17] assumes a Markov-tree structure in the sparse coefficients and adapts the AMP algorithm in a Bayesian framework.

Commonly used sparse reconstruction techniques assume that the basis $\boldsymbol{\Psi}$ is exactly known and the signal is sparse in that basis. However, in some applications there is a mismatch between the assumed basis and the actual but unknown one. For example in applications like target localization [18], radar [19, 20], time delay and doppler estimation, beamforming [21, 22] or shape detection [23], the sparsity of the signal is in a continuous parameter space and the sparsity basis $\boldsymbol{\Psi}$ is constructed through discritization or griding of these parameter spaces. In general, a signal will not be sparse in such a dictionary created through discritization, since no matter how fine the grid dimensions are, the signal parameters may not, and generally do not, lie in the center of the used grid cells. As a simple example; consider a general signal which is sparse in the continuous frequency domain. This signal may not be sparse in the DFT basis defined by the frequency grid. A continuous frequency parameter lying between two successive DFT grid cells will affect not the only the closest two cells, but the whole grid with amplitude decaying with $1/T$, where T is the sampling time interval. This off-grid phenomena violates the sparsity assumption, resulting in a decrease in reconstruction performance. In addition to these structured perturbations, random time jitter in A/D conversion, modeling errors in construction of the dictionary $\boldsymbol{\Psi}$ create perturbations on the dictionary columns. Hence, in general, the signal $\boldsymbol{x}$ will be sparse in an unknown basis $\hat{\boldsymbol{\Psi}} = \boldsymbol{\Psi} + \boldsymbol{P}$ where $\boldsymbol{\Psi}$ is the adopted basis and $\boldsymbol{P}$ is the unknown perturbation matrix. Since the classical CS theory evolves around the solution to the overdetermined system in the form of $\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{\Psi}\boldsymbol{x}$, developed sparse solvers are not robust to this type of errors. Under such a basis mismatch problem, classical techniques suffer from a significant degradation in the recovery performance.

In the literature, the effect of this basis mismatch has been observed and analyzed in some applications such as radar [19, 24] and beamforming [25]. In problems due to parameter space discritization, a simplistic approach is to use multi-resolution refinement and decrease the grid size. Decreasing the grid size is not a direct solution to the basis mismatch problem, because it increases the coherence between dictionary columns, which in turn result in violation of the restricted isometry property (RIP) [26] and increase in the computational complexity of the reconstruction. In [27–29] the effect of the basis mismatch problem on the reconstruction performance of CS has been analyzed and the resultant performance degradation levels and analytical ℓ_2 norm error bounds due to the basis mismatch have been investigated. However, these works do not offer a systematic approach for sparse reconstruction under random perturbation models. In [30], the dictionary is extended to several dictionaries and solution is pursued not in a single orthogonal basis, but in a set of bases using a tree structure, assuming that the given signal is sparse in at least one of the basis. However, this strategy does not provide solutions if the signal is not-sparse in the extended dictionary. In the continuous basis pursuit approach (CBP) [31], perturbations are assumed to be continuously shifted features of the functions on which the sparse solution is searched for, and ℓ_1 based minimization is proposed. In [32], ℓ_1 minimization based algorithms are proposed for linear structured perturbations on the sensing matrix. In [33] a total least square (TLS) solution is proposed for the problem, in which an optimization over all signals $\boldsymbol{x}$, perturbation matrix $\boldsymbol{P}$ and error vector spaces should be solved. To reduce complexity, suboptimal optimization techniques have been pursued in [33].

To overcome the basis mismatch problem, this thesis introduces two novel perturbed sparse recovery algorithms. The first one primarily focuses on reconstruction of sparse parameter scenes and proposes a novel parameter perturbation based sparse reconstruction technique to provide robust reconstructions in the off-grid case. The proposed technique is an iterative algorithm that works with a selected set of dictionary vectors that can be obtained via one of sparse greedy techniques such as MP, OMP, IHT, CoSaMP. The parameters of the selected dictionary atoms are iteratively adapted within their grids towards directions that decreases the residual norm. The proposed technique presently is used within the general OMP framework hence named as Parameter Perturbed OMP (PPOMP). As demonstrated in the reconstruction of sparse delay-Doppler radar scenes, the proposed method is successful in recovering the targets with arbitrary positions. Compared to conventional CS reconstruction techniques like OMP or ℓ_1 minimization, proposed PPOMP technique achieves lower reconstruction errors for a general delay-Doppler scene in all the conducted performance tests. The general idea of proposed parameter perturbation can also be applied to other areas where discrete parameters are selected from continuous parameter spaces such as frequency or angle of arrival estimation problems.

The second proposed technique mainly focuses on unstructured basis mismatch problem which is observed under the random time jitter in A/D conversion and modeling error. To overcome this type of mismatch, a novel Perturbed OMP (POMP) algorithm is presented. In the standard OMP algorithm [12] the column vector that has the largest correlation with the current residual is selected and the new residual is calculated by projecting the measurements onto the subspace defined by the span of all selected columns. This procedure is repeated until the termination criteria is met. In the proposed POMP algorithm, controlled perturbation mechanism is applied on the selected columns. The selected column vectors are perturbed in directions that decrease the orthogonal residual at each iteration. Proven limits on perturbations are obtained. Relative to the regular OMP, under the perturbation scheme, POMP is able to produce sparser solutions. The proposed technique is simulated on a frequency estimation problem under sampling time jitter. Due to its optimal but simple perturbation scheme, the proposed method is fast, simple to implement and successful in recovering sparse signals under random basis perturbations.

The rest of the thesis also focuses on the off-grid problem. The proposed recursive CS framework addressed the off-grid problem by developing an adaptive discretization scheme so that a regular solver such as OMP and CoSaMP can recover a sparse signal without observing an off-grid problem. The relation between the discretization limit and the sparsity level plays a key role in the proposed recursive reconstruction framework. In the proposed approach, the observation is represented as a superposition of c components: $\boldsymbol{y} = \boldsymbol{y}_1 + \boldsymbol{y}_2 + \ldots + \boldsymbol{y}_c$ with sparsities K_1, K_2, $\ldots, K_c$, respectively. If $\boldsymbol{y}_i$ and K_i, where K_i is significantly less than K, are given, continuous signal space can be discretized denser. Thus estimates for K_i-sparse signals would have less off-grid resulted degradation. The use of adaptive reconstruction grids whose density depends on the sparsity level is the main difference between the proposed reconstruction technique and the E-M based approaches. A powerful aspect of this approach is that the main problem and the sub-problems are equivalent to each other in the structural sense: they all take an observation vector and a sparsity level as inputs and produce estimates of the parameters as outputs. Therefore, each sub-problem can be partitioned further into sparser problems and solved with denser discretization. Due to self-similar structure of the partitioning, a recursive algorithm that discretizes the space adaptively is presented. The proposed approach is fast, suitable for parallel computing and provides powerful estimation results by achieving Cramér-Rao bound.

1.1 Contribution and Organization

The thesis starts with an introduction to compressive sensing. Well known reconstruction algorithms are summarized and their performances are illustrated over well known applications. In the exposition of ideas, many of the technical details are avoided and directed to key references on the subject.

Following the introduction to CS, Chapter 3 details the shortcomings of the CS reconstruction algorithms in the continuous signal spaces. Afterwards, as solutions to these shortcomings, novel Parameter Perturbed Orthogonal Matching Pursuit (PPOMP) and Perturbed Orthogonal Matching Pursuit (POMP) algorithms are introduced. Theoretical investigations and simulated reconstruction performances of these novel algorithms are also provided in this chapter.

Chapter 4 discusses the relation between the sparsity and the discretization of the continuous space and treats the CS problem as an estimation problem rather than a detection problem. For highly improved reconstructions, a novel recursive framework, in which the continuous parameter space is adaptively discretized, is introduced. Simulations of CRLB achieving performance of the proposed techniques is also presented there.

Lipschitz continuity of the cost function in delay-Doppler formulation, detailed proof of a theorem related to the POMP algorithm and the CRLB analysis for single frequency estimation under random sampling are provided in the appendix.

1.2 Notation

Thorough out the thesis, several norms are utilized. For vectors, mostly ℓ_p norm, which is indicated as $\|\cdot\|_p$, is used. In order to prevent any ambiguity, Euclidean norms are also stated explicitly as $\|\cdot\|_2$. Even though ℓ_p norm is not defined for $p = 0$, we use $\|\cdot\|_0$ "norm" to denote the number of non-zero elements in the given vector. For matrices, $\|\cdot\|_2$ is defined as the spectral norm, $\|\cdot\|_\star$ is defined as the nuclear norm and $tr(\cdot)$ denotes the trace of the argument.

$\boldsymbol{A}^T$ denotes the transpose of the matrix $\boldsymbol{A}$. $\boldsymbol{A}^H$ denotes the hermitian of the matrix $\boldsymbol{A}$. Superscript $\boldsymbol{A}^\dagger$ is the Moore-Penrose pseudo-inverse of $\boldsymbol{A}$.

For $\boldsymbol{A}$, $\boldsymbol{P}_{\boldsymbol{A}}$ denotes the projection matrix onto the column space of $\boldsymbol{A}$ and $\boldsymbol{P}^{\perp}_{\boldsymbol{A}}$ denotes the projection to the perpendicular space of the column space of $\boldsymbol{A}$.

The operator "$\circ$" denotes the Hadamard product, or entrywise product. For two matrices $\boldsymbol{A}, \boldsymbol{B} \in \mathcal{C}^{M\times N}$, their Hadamard product is denoted as $\boldsymbol{M} = \boldsymbol{A} \circ \boldsymbol{B} \in \mathcal{C}^{M\times N}$ with elements given by $\boldsymbol{M}(l, k) = \boldsymbol{A}(l, k) \cdot \boldsymbol{B}(l, k)$.

Chapter 2

Compressive Sensing

In this chapter, a summary of the well-established CS theory is presented. Basic insights and heuristics behind the idea of the transform domain representation and compressed measurements is provided. For more comprehensive and mathematical introduction, the reader may refer to [34, 35]. Also, highly extensive and well categorized publication lists on the theory of CS and its applications can be found in [36].

2.1 Transform Domain Representation

Suppose N dimensional signal $\boldsymbol{s}$ has a representation in a transform domain as follows:

$$\boldsymbol{s} = \boldsymbol{\Psi}\boldsymbol{x}, \tag{2.1}$$

where $\boldsymbol{\Psi} \in \mathcal{C}^{N\times N}$ is a non-singular, and generally orthonormal, transformation matrix and $\boldsymbol{x}$ is the representation of the non-sparse signal $\boldsymbol{s}$ in the transform domain. Since the basis $\boldsymbol{\Psi}$ is non-singular, representation of the sampled signal $\boldsymbol{s}$ in this basis can be obtained as:

$$\boldsymbol{x} = \boldsymbol{\Psi}^{-1}\,\boldsymbol{s}. \tag{2.2}$$

As a simple example, consider a time-sampled signal $\boldsymbol{s} \in \Re^{256}$. Visual representation of such a signal is shown in Fig. 2.1. In the time domain, the signal is dense, i.e. $\|\boldsymbol{s}\|_0 = N$ meaning that all the elements are non-zero. However, if the representation basis $\boldsymbol{\Psi}$ is selected as Inverse Discrete Cosine Transform (IDCT) matrix, it is possible to achieve a sparse decomposition. IDCT is defined as:

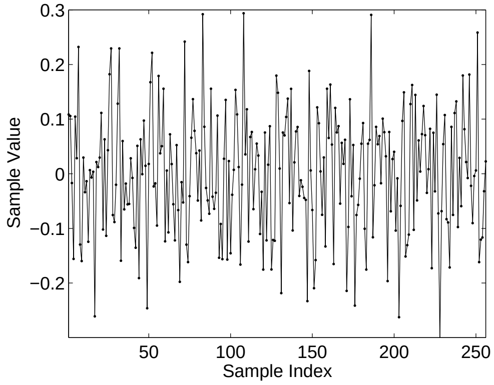
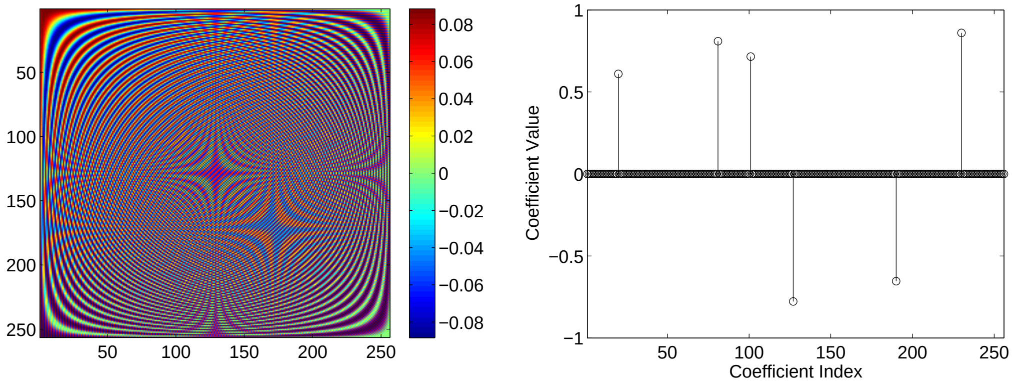


Figure 2.1: Time-sampled signal $\boldsymbol{s}$.

$$\Psi_{l,k} = \begin{cases} \dfrac{1}{\sqrt{N}}, & \text{if } k = 1, \quad 1 \leq l \leq N, \\ \dfrac{\sqrt{2}}{\sqrt{N}} \cos\left(\dfrac{\pi\,(2l-1)\,(k-1)}{2N}\right), & \text{if } 2 \leq k \leq N, \quad 1 \leq l \leq N, \end{cases} \tag{2.3}$$

where $\Psi_{l,k}$ is the $(l,k)^{th}$ element of the matrix $\boldsymbol{\Psi}$. Visual representation of this orthonormal basis is shown in Fig. 2.2(a) for $N = 256$. Resulting transform domain representation of $\boldsymbol{s}$, computed with (2.2), is shown in Fig. 2.2(b).

Most important property of the transform domain signal $\boldsymbol{x}$ is its sparse behavior. Since $\|\boldsymbol{x}\|_0 = 6$, dense signal $\boldsymbol{s}$ of length 256 can be represented only by 6 coefficients. Therefore, it is significantly more efficient to process $\boldsymbol{s}$ in the representation domain. Furthermore, such a sparsifying transform also provides a way of compressing the sampled data sequence. In the illustrated case, sampled data of length 256 is compressed down to 6 coefficients. Taking the corresponding indexes of the coefficients into account, this reduction in the required number of coefficient is equivalent to lossless compression of rate 95.3%.



(a) (b)

Figure 2.2: (a) IDCT matrix $\boldsymbol{\Psi}$ with size $N = 256$, (b) sparse representation of the signal $\boldsymbol{s}$ in the domain defined in (2.3).

2.2 Basic Idea

Even though transform domain representation enables us to compress and process a signal very efficiently, there is no improvement for the sampling process since it is a necessity to acquire full set of N samples in the first place. Then the regular process is to compress the samples of length N to few coefficients of length K. The drawback of this classical sample & compress approach is to acquire too many samples in order to compress them using a sparse representation. To alleviate this drawback, Compressive Sensing (CS) offers a new sampling paradigm in which the transform domain representation (compressed version) of the signal is observed rather than the signal itself, as the name *"Compressive Sensing"* suggests.

Basically, the CS states that a signal with a sparse representation in some known basis can be efficiently acquired using a small set of nonadaptive and linear measurements. For this purpose, let $\boldsymbol{\Phi} \in \mathcal{C}^{M \times N}$ be a sensing matrix, where the effective number of samples M is smaller than the signal length N. Instead of observing the signal $\boldsymbol{s}$ directly, it is sampled through the sensing matrix as follows:

$$\boldsymbol{y} = \boldsymbol{\Phi}\, \boldsymbol{s}, \tag{2.4}$$

or, with the transform domain representation of $\boldsymbol{s}$, it can also be stated as:

$$\boldsymbol{y} = \boldsymbol{\Phi}\,\boldsymbol{\Psi}\,\boldsymbol{x}. \tag{2.5}$$

In the following, the sensing matrix $\boldsymbol{\Phi}$ and the basis $\boldsymbol{\Psi}$ will be combined into a dictionary $\boldsymbol{A} \in \mathcal{C}^{M \times N}$ as $\boldsymbol{A} = \boldsymbol{\Phi}\boldsymbol{\Psi}$ for the simplicity. Hence in the combined model, the measurements will be related to the sparse vector as:

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}. \tag{2.6}$$

Having $M < N$, the main problem with the model in (2.6) is that $\boldsymbol{y}$ does not have a unique representation in the domain of $\boldsymbol{A}$. In fact, assuming $rank(\boldsymbol{A}) = M$, set of all solutions for a given $\boldsymbol{y}$ forms a $N - M$ dimensional manifold in the N dimensional solution space. Most importantly, hardly any of these possible solutions are sparse! However, it is known that $\boldsymbol{s}$ has a sparse decomposition in the basis $\boldsymbol{\Psi}$ and the same representation vector $\boldsymbol{x}$ is also a valid solution for compressed measurements $\boldsymbol{y}$. Therefore, the main purpose in CS theory is to find the most sparse vector that explains the measurements, which can be cast as the following optimization problem:

$$\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_0 \quad \text{s.t.} \quad \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2 = 0. \tag{2.7}$$

In most of the systems, an additive and independent, generally Gaussian, noise is present in the sampling process. Hence, in a more realistic scenario the compressed measurement vector can be represented as:

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{n}, \tag{2.8}$$

where $\boldsymbol{n}$ stands for the additive noise term with known statistical behavior. Since the compressed measurements cannot be stated as a sparse linear combination of the columns of $\boldsymbol{A}$ under the noisy setting, more general optimization problem of finding the sparse solution can be stated as:

$$\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_0 \quad \text{s.t.} \quad \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2 \leq \epsilon. \tag{2.9}$$

where ϵ is the expected residual for the given noise statistics.

Given the compressed measurement vector $\boldsymbol{y}$, solution to problem in (2.9) reveals the desired sparse representation. However, due to discontinuous characteristics of the ℓ_0 norm, solution is not computationally efficient and requires a combinatoric search over all possible sparse representations. Assume that the signal in Fig. 2.1 is acquired through a sensing matrix $\boldsymbol{\Phi}$ under an additive noise with known statistics. If the best 6-sparse representation of the measurements is searched for, one should check every sub-matrix consisting of 6 different columns of $\boldsymbol{A}$ to ensure that resulting residual from the projection of $\boldsymbol{y}$ onto the column space of the sub-matrix is below the set noise level ϵ. Although total number of such sub-matrices is finite and solution can be obtained in a finite amount of time, even in this moderate size example there are total of $\binom{256}{6}$ such sub-matrices. As a result, the best 6-sparse representation of the signal in Fig. 2.1 can be found via solving total of approximately 3.7×10^{11} projections, which can be stated as least-squares problems. Considering that MATLAB running on a regular desktop computer can solve a such least-squares problem in approximately 10^{-5} seconds, total required computation time is around 40 days! If the signal had a 7-sparse representation this computation time would jump to 4 years!

2.3 Sparse Reconstruction Algorithms

Even though sparse approximation is a well-defined problem in (2.9), the optimal solution requires an exponentially complex exhaustive search due to discontinuous characteristics of the sparsity measure, ℓ_0 norm. In the following sub-sections, several algorithms are introduced, which can approximate the optimal solution.

2.3.1 Convex Relaxation

Basic idea in the convex relaxation approach is to replace the problematic ℓ_0 norm with a more tractable sparsity measure. One straightforward approach is to use ℓ_2 norm. However, with the use of ℓ_2 norm, (2.7) reduces to the standard least-squares problem. Even though the solution is straightforward, it is not sparse.

The most well-recognized convex relaxation is to replace the ℓ_0 with the ℓ_1 norm. Thus, the problem is stated as:

$$\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_1 \quad \text{s.t.} \quad \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2 = 0. \tag{2.10}$$

Main advantage of this form is that this problem can be converted to the following standard linear programming [37]:

$$\min \mathbf{1}^T\boldsymbol{u} \quad \text{s.t.} \quad \boldsymbol{W}\boldsymbol{u} = \mathbf{0}, \quad \boldsymbol{u} \geq 0, \tag{2.11}$$

with $\boldsymbol{W} = [\boldsymbol{A},\ -\boldsymbol{A}]$ and $\boldsymbol{x} \in \Re^{2M}$. The important thing about linear programing is that it can be solved numerically in a polynomial time. Results reported in the key papers [8–10] have shown that sparse representation of the signal can be revealed with the solution of the problem in (2.10), which motivated the compressive sensing society to concentrate on the ℓ_1 solution, hence convex optimization techniques in general. A willing reader may refer to [35] for detailed analysis and solution techniques of the ℓ_1 problem in (2.10). Comprehensive introduction to convex optimization techniques can also be found in [38, 39].

2.3.2 Greedy Pursuits

Unlike the convex optimization approaches, greedy techniques iteratively search for the optimal support of the sparse signal. Then the corresponding coefficients are found from a projection of the compressed measurement to the selected support. In the following, well-known Orthogonal Matching Pursuit(OMP) and

Compressive Sampling Matching Pursuit(CoSaMP) are detailed. Even though they are both greedy techniques, OMP tries to find a sparsest representation with a residual smaller than a preset level ϵ, whereas CoSaMP tries to find the best K-sparse representation of the compressed measurements for a given K.

2.3.2.1 Orthogonal Matching Pursuit (OMP)

Solution to the sparse representation problem of (2.9) requires a search over all possible sub-matrices of the dictionary $\boldsymbol{A}$ for the global optimality. Given a compressed observation vector $\boldsymbol{y}$, the optimal brute-force solution first considers all possible 1-sparse representations and checks that residual error is below the given threshold ϵ. If all of 1-sparse representations fails to produce a sufficiently small error, then the optimal solution searches over all possible 2-sparse representations. Proceeding with the same rationale, the optimal solution searches for all possible K-sparse representations until the resulting error is sufficiently small.

Orthogonal Matching Pursuit (OMP), on the other hand, gives up the desire for global optimality and makes series of locally optimal decisions. OMP starts with a search for the best 1-sparse representation. However in the next turn, instead of looking for all 2-sparse representations, OMP keeps the previously selected representation vector and searches for the optimal second vector. In the next turn, it keeps the previously selected 2 representation vectors and searches for the optimal third vector. Proceeding with the same idea, OMP adds a locally optimum support vector in each iteration. Even though this strategy does not guarantee the global optimality, OMP algorithm can produce a solution in a couple of iterations lasting a fraction of a second. More importantly, total computational complexity scales linearly with K contrary to the exponential growth in the optimal search.

More formally, k^{th} iteration of this type of greedy pursuit starts with an current solution set $\boldsymbol{S}_{k-1}$ and a residual to fit $\boldsymbol{y}_{\perp,k-1}$. The main step of OMP is to find the most suitable basis vector as a solution to the following problem:

$$j^{*} = \arg \max_{1 \leq j \leq N} \frac{\left| \boldsymbol{a}_j^H \; \boldsymbol{y}_{\perp,k-1} \right|}{\|\boldsymbol{a}_j\|_2^2}. \tag{2.12}$$

The problem in (2.12) is a simple search over the dictionary to find the vector that has the highest normalized inner product with the current residual $\boldsymbol{y}_{\perp,k-1}$. After the search step, let j^* denote the column of $\boldsymbol{A}$ so that it has the highest correlation among all. After updating the solution set as $\boldsymbol{S}_k = \left[\boldsymbol{S}_{k-1} \; \boldsymbol{a}_{j^*}\right]$, the following phase is to compute the new residual as a solution of the following optimization:

$$\min_{\boldsymbol{x}} \left\| \boldsymbol{y} - \boldsymbol{S}_k \, \boldsymbol{x} \right\|_2, \tag{2.13}$$

which is a simple least-squares problem with the following residual:

$$\boldsymbol{y}_{\perp,k} = \boldsymbol{y} - \boldsymbol{S}_k \, \boldsymbol{S}_k^{\dagger} \, \boldsymbol{y}, \tag{2.14}$$

where $\boldsymbol{S}_k^{\dagger}$ is the pseudo-inverse of $\boldsymbol{S}_k$. If the norm of the residual $\boldsymbol{y}_{\perp,k}$ is not below the pre-specified threshold level ϵ, OMP starts the $(k+1)^{th}$ iteration with $\boldsymbol{S}_k$ and $\boldsymbol{y}_{\perp,k}$ and goes through the same steps. The steps of the OMP algorithm are detailed as a pseudo-code in Table 2.1.

Table 2.1: Orthogonal Matching Pursuit(OMP) Algorithm

```
Inputs: $(\boldsymbol{y},\ \boldsymbol{A},\ \epsilon)$

Initialization: $\boldsymbol{y}_\perp = \boldsymbol{y}$, $\boldsymbol{S}_0 = \{\}$, $e = \|\boldsymbol{y}_\perp\|_2$, $k = 1$

Keep iterating until $e < \epsilon$
    $j^* = \arg \max_{1 \leq j \leq N} |\boldsymbol{a}_j^H \; \boldsymbol{y}_\perp| \, / \, \|\boldsymbol{a}_j\|_2^2$
    $\boldsymbol{S}_k = \left[\boldsymbol{S}_{k-1} \; \boldsymbol{a}_{j^*}\right]$
    $\boldsymbol{\alpha} = \boldsymbol{S}_k^{\dagger} \; \boldsymbol{y}$
    $\boldsymbol{y}_\perp = \boldsymbol{y} - \boldsymbol{S}_k \; \boldsymbol{\alpha}$
    $e = \|\boldsymbol{y}_\perp\|_2$
    $k = k + 1$

Output: $(\boldsymbol{\alpha},\ \boldsymbol{S}_k)$
```

The most important property of OMP is that the residual has always zero inner product with the selected basis vectors. In the $(k+1)^{th}$ iteration, the inner-product between the residual and the previously selected k vectors can be written as:

$$\begin{aligned} \boldsymbol{S}_k^H \boldsymbol{y}_{\perp,k} &= \boldsymbol{S}_k^H \left(\boldsymbol{y} - \boldsymbol{S}_k \boldsymbol{S}_k^{\dagger} \boldsymbol{y} \right) \\ &= \boldsymbol{S}_k^H \boldsymbol{y} - \boldsymbol{S}_k^H \boldsymbol{S}_k \boldsymbol{S}_k^{\dagger} \boldsymbol{y} \\ &= \boldsymbol{S}_k^H \boldsymbol{y} - \boldsymbol{S}_k^H \boldsymbol{S}_k \left(\boldsymbol{S}_k^H \boldsymbol{S}_k \right)^{-1} \boldsymbol{S}_k^H \boldsymbol{y} \\ &= \boldsymbol{0}, \end{aligned} \tag{2.15}$$

hence $\boldsymbol{a}_j^H \boldsymbol{y}_{\perp,k} = 0$ if $\boldsymbol{a}_j$ is contained in $\boldsymbol{S}_k$. Therefore as the name Orthogonal Matching Pursuit suggests, the residual which is orthogonal to the selected columns is tried to match maximally to a column of the dictionary $\boldsymbol{A}$ at each iteration of the algorithm.

2.3.2.2 Compressive Sampling Matching Pursuit(CoSaMP)

CoSaMP [13] tries to fit the best K-sparse representation independent of the resulting residual error. From this perspective, CoSaMP initialized with K is similar to first K step of OMP iterations.

In the l^{th} iteration, CoSaMP starts with a current solution set $\boldsymbol{S}_K$ and a residual $\boldsymbol{y}_{\perp,l-1}$. Similar to the search step of OMP, CoSaMP also computes the normalized inner products between the residual and the each dictionary columns as $|\boldsymbol{a}_j^H \boldsymbol{y}_{\perp,l-1}|/\|\boldsymbol{a}\|_2^2$ for $1 \leq j \leq N$. Unlike the OMP searching for the maximum correlation, CoSaMP selects $2K$ dictionary vectors with the largest normalized inner products. Let $\boldsymbol{U}_{2K}$ denote the set of such $2K$ column vectors of $\boldsymbol{A}$. The next step is to merge $\boldsymbol{S}_K$ and $\boldsymbol{U}_{2K}$ into a new set denoted by $\boldsymbol{T}$. Notice that size of the merged set $\boldsymbol{T}$ is between $2K$ and $3K$ depending on the length of the intersection of $\boldsymbol{S}_K$ and $\boldsymbol{U}_{2K}$. In this phase, CoSaMP finds a solution on the merged support $\boldsymbol{T}$ with the following optimization problem:

$$\arg\min_{\boldsymbol{x}} \left\| \boldsymbol{y} - \boldsymbol{T}\boldsymbol{x} \right\|_2, \tag{2.16}$$

which has the following solution:

$$\boldsymbol{\alpha} = \boldsymbol{T}^{\dagger}\, \boldsymbol{x}, \tag{2.17}$$

where $\boldsymbol{\alpha}$ is the vector containing the coefficients, weights, of the corresponding columns. Since $\boldsymbol{\alpha}$ contains more than $2K$ coefficients, next step of the algorithm is to prune the coefficient set by eliminating all but the largest K. Let $\boldsymbol{\alpha}'$ be the set of largest K coefficients in $\boldsymbol{\alpha}$ and $\boldsymbol{T}'$ be their corresponding column vectors. Then, the residual from the current K-sparse estimation is computed as:

$$\boldsymbol{y}_{\perp,l} = \boldsymbol{y} - \boldsymbol{T}'\, \boldsymbol{\alpha}'. \tag{2.18}$$

The algorithm repeats this procedure until some termination criteria is met.

Table 2.2: Compressive Sampling Matching Pursuit(CoSaMP) Algorithm

```
Inputs: $(\boldsymbol{y},\ \boldsymbol{A},\ K)$

Initialization: $\boldsymbol{y}_{\perp} = \boldsymbol{y}$, $\boldsymbol{S}_K = \{\}$, $l = 1$

Keep until convergence
  $\boldsymbol{U}_{2K} = \max_{2K}\ |\boldsymbol{a}_j^H\ \boldsymbol{y}_{\perp}|\ /\ \|\boldsymbol{a}_j\|_2^2$
  $\boldsymbol{T} = \boldsymbol{S}_K \cup \boldsymbol{U}_{2K}$
  $\boldsymbol{\alpha} = \boldsymbol{T}^{\dagger}\ \boldsymbol{y}$
  $\boldsymbol{\alpha}' \leftarrow \boldsymbol{\alpha}, \quad \boldsymbol{T}' \leftarrow \boldsymbol{T}$
  $\boldsymbol{S}_K = \boldsymbol{T}'$
  $\boldsymbol{y}_{\perp} = \boldsymbol{y} - \boldsymbol{T}'\ \boldsymbol{\alpha}'$
  $l = l + 1$

Output: $(\boldsymbol{\alpha}',\ \boldsymbol{S}_K)$
```

Steps of the CoSaMP algorithm is presented as a pseudo-code in Table 2.2. Unlike the OMP where a selected vector cannot be dropped in future iterations, the main feature of CoSaMP is that the solution support is allowed to change at each iteration. With this property, CoSaMP is able to correct previously made errors in the estimated support, whereas OMP cannot remove a vector from the solution set even if it does not belong to the actual support.

2.4 Recovery Guarantees

Even though the basis $\boldsymbol{\Psi} \in \mathcal{C}^{N \times N}$ is non-singular, due to sensing matrix decreasing the effective number of samples, the resulting dictionary $\boldsymbol{A} \in \mathcal{C}^{M \times N}$ has a non-empty null space. Assume that a signal $\boldsymbol{y}$ has a sparse decomposition as $\boldsymbol{y} = \boldsymbol{A}\,\boldsymbol{x}$ with $\|\boldsymbol{x}\|_0 = K$. Let $\boldsymbol{u}$ be a vector from the null space as $\boldsymbol{0} = \boldsymbol{A}\,\boldsymbol{u}$. Hence, the following is also valid $\boldsymbol{y} = \boldsymbol{A}\,(\boldsymbol{x} + \boldsymbol{u})$. Since $\boldsymbol{u}$ is a dense vector with $\|\boldsymbol{u}\|_0 = N$, the following relation holds true: $N - K \leq \|\boldsymbol{x} + \boldsymbol{u}\|_0 \leq N$. In order to guarantee the uniqueness of the sparse representation vector $\boldsymbol{x}$, we need $K < N - K$ so that, $\|\boldsymbol{x} + \boldsymbol{u}\|_0 > K$ is always true and there is no other K, or less, sparse representation. Hence, having $K < N/2$ guarantees the uniqueness of the K-sparse representation.

Even though a K-sparse representation of a signal $\boldsymbol{y}$ is guaranteed to be unique when $K < N/2$, this does not necessarily mean that a sparse solver can reveal the correct sparse representation. Fortunately, CS theory provides set of recovery guarantees for different sparse solvers, most of which depends on a measure that is related to the coherence of $\boldsymbol{A}$.

One of the well-known coherence measures is the Restricted Isometry Property (RIP), which is defined as the smallest δ_K that satisfies the following for all $\boldsymbol{x}$ and for all sub-matrix $\boldsymbol{A}_K$ of $\boldsymbol{A}$:

$$(1 - \delta_K)\,\|\boldsymbol{x}\|_2^2 \;\leq\; \|\boldsymbol{A}_K\,\boldsymbol{x}\|_2^2 \;\leq\; (1 + \delta_K)\,\|\boldsymbol{x}\|_2^2. \tag{2.19}$$

If there is a such δ_K, then $\boldsymbol{A}$ said to satisfy the K-RIP with constant δ_K. Given a sub-matrix $\boldsymbol{A}_K$ of $\boldsymbol{A}$, its corresponding δ can be computed as:

$$\delta = \max\{\ \lambda_{max}(\boldsymbol{A}_K^H \boldsymbol{A}_K) - 1,\ 1 - \lambda_{min}(\boldsymbol{A}_K^H \boldsymbol{A}_K)\ \}, \tag{2.20}$$

where λ_{min} and λ_{max} denote the minimum and maximum eigenvalues of the argument, respectively. In order to find δ_K, pass over all possible sub-matrices and computation of (2.20), which has an exponential complexity, is required. NP-hardness of the computation of δ_K is also shown in [40]. Yet, RIP is used widely in the CS theory to analyze the recovery abilities of sparse solvers. In [41], OMP is guaranteed to recover a K-sparse signal in the form of $\boldsymbol{y} = \boldsymbol{A}\,\boldsymbol{x}$, if $\boldsymbol{A}$ satisfies RIP with $\delta_{K+1} \leq \frac{1}{3\sqrt{K}}$ and this bound is improved further in [42]. In [13] it is shown that CoSaMP will result in a bounded recovery error of K-sparse signal if the dictionary satisfies $\delta_{2K} < c$ for some constant c. Some recovery guarantees for ℓ_1 optimization is also provided in [8].

Another commonly used coherence measure is the mutual coherence of the dictionary that is defined as:

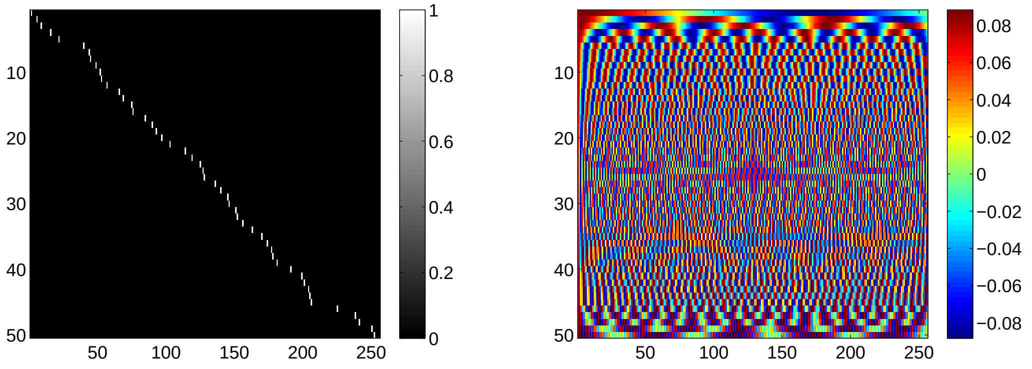
$$\mu(\boldsymbol{A}) = \max_{i \neq j} \frac{|\boldsymbol{a}_i^H\,\boldsymbol{a}_j|}{\|\boldsymbol{a}_i\|_2\,\|\boldsymbol{a}_j\|_2}, \tag{2.21}$$

which is the maximum normalized inner product between two arbitrary columns of the dictionary $\boldsymbol{A}$. If $\mu(\boldsymbol{A}) = 1$, then the dictionary includes two columns which coincide with each other and if $\mu(\boldsymbol{A}) = 0$, then the dictionary is orthogonal. In [43], OMP is shown to recover a K-sparse signal if the dictionary $\boldsymbol{A}$ satisfies $\mu(\boldsymbol{A}) \leq \frac{1}{2K-1}$.

The basic insight about these coherence-based guarantees is that if the columns of the dictionary $\boldsymbol{A}$ become similar to each other, they became indistinguishable under the linear combination. Since the observation $\boldsymbol{y}$ is a linear combination of some columns of $\boldsymbol{A}$, the corresponding columns are required to be as incoherent as possible for a guaranteed recovery. However, the main problem with these guarantees is that they are too pessimistic and try to cover *any* K-sparse signal, including the worst case combinations. Hence, those guarantees do not express the actual capabilities of sparse solvers. As simulated in the following section, OMP is able to recover up to $K = 6$ sparse signals in the given dictionary $\boldsymbol{A}$, whereas mutual coherence based condition guarantees only recovery of a 1-sparse signal since $\mu(\boldsymbol{A}) = 0.53$.

2.5 Simulated Recovery Performances

In this section, performance of the recovery algorithms given in Section 2.3 are simulated. IDCT matrix defined in (2.3) is used as the sparsifying basis. Sensing matrix is constructed by randomly selecting $M = 50$ rows of the $N \times N$ identity matrix. Sensing matrix selected with this scheme corresponds to randomly selecting M elements from the signal of interest $\boldsymbol{s}$. The constructed sensing matrix, which is used in the following simulations, is given in Fig. 2.3(a), and the resulting dictionary $\boldsymbol{A} = \boldsymbol{\Phi\Psi}$ is also shown in Fig. 2.3(b).

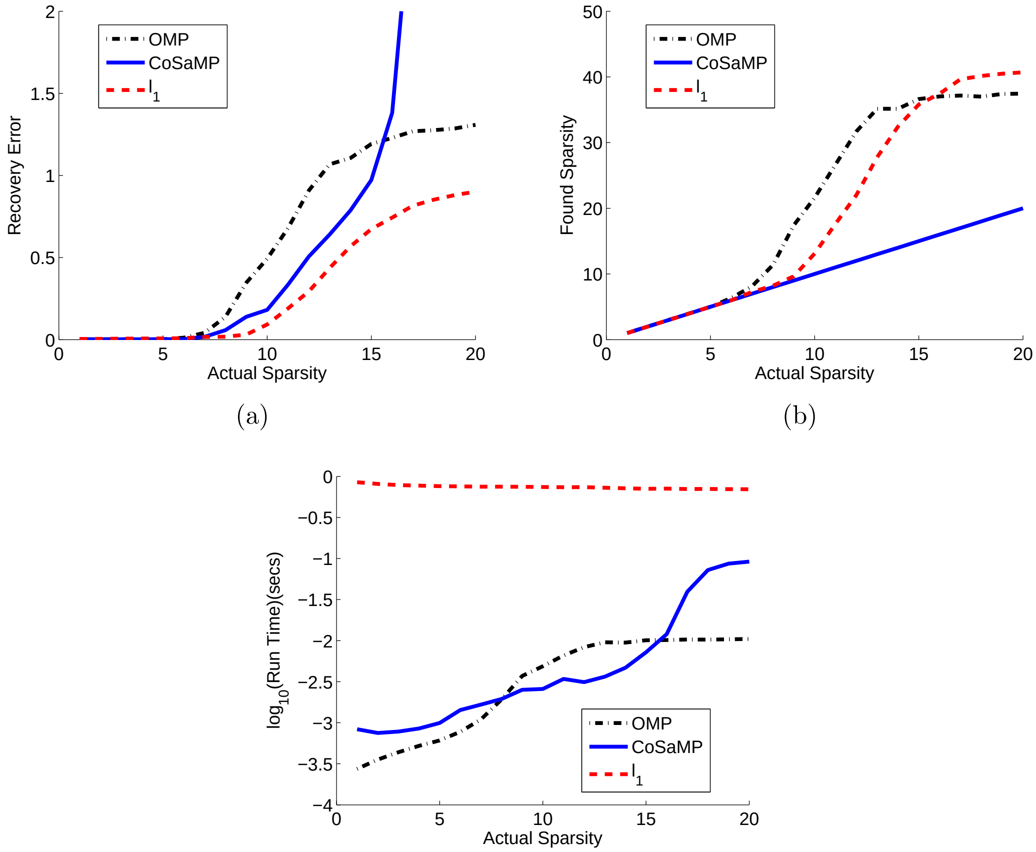


(a)

(b)

Figure 2.3: (a) Sensing matrix $\boldsymbol{\Phi} \in \Re^{M \times N}$ constructed by randomly decimating the rows of an identity matrix, (b) combined dictionary $\boldsymbol{A} = \boldsymbol{\Phi\Psi}$ where $\boldsymbol{\Psi}$ is the IDCT matrix defined in (2.3).

In the simulations, a sparse vector $\boldsymbol{x} \in \Re^N$ is generated randomly with sparsity level changing from 1 to 20. The measurement noise is i.i.d. Gaussian noise with standard deviation $\sigma = 0.001$. The resulting compressed observation vector $\boldsymbol{y}$ is constructed as (2.8). The expected residual level ϵ is selected as $\epsilon = 1.2\,\sigma\,\sqrt{M}$, where factor of 1.2 is used to relax the noise level. The vector $\boldsymbol{y}$ and ϵ is given to solvers and the resulting solutions are compared to the actual solution. This procedure is repeated 100 times to obtain an average recovery performance.



(a)

(b)

(c)

Figure 2.4: Average sparse reconstruction performance of OMP, CoSaMP and ℓ_1 optimization, (a) recovery error, (b) found sparsity, (c) solution time.

Fig. 2.4 shows the recovery performances of OMP, CoSaMP and ℓ_1 optimization at different sparsity levels. Fig. 2.4(a) shows the average error of recovery defined as $\|\boldsymbol{x}^* - \boldsymbol{x}\|_2/\|\boldsymbol{x}\|_2$, where $\boldsymbol{x}^*$ denotes the recovered sparse solution, Fig. 2.4(b) shows the average sparsity of the recovered solutions and Fig. 2.4(c) shows the average time of computation . All algorithms are implemented in MATLAB running on a desktop computer.

Depending on the error levels, sparsity axes can be divided into two: the region where the solutions are correct and the region where solutions are in significant error. This point at which performance changes dramatically is called as threshold sparsity level of the algorithm. When this analysis is extended to different number of measurements, the phase transition of the algorithms is characterized. Such analysis for OMP can be found in Section 3.3.3.3 in Fig. 3.16 and 3.17.

OMP, being the fastest among all three techniques, has the smallest solvable sparsity level of $K = 6$. Even though ℓ_1 optimization can solve signals with sparsity up to $K = 9$, this increase in performance comes with a cost of 3 orders of magnitude more computation time. CoSaMP, on the other hand, performs better than OMP with a very similar computation time. However, main disadvantage of CoSaMP is its requirement to know the actual sparsity level. Having favorable reconstruction performance with significantly lower computation time makes OMP a powerful sparse solver.

Chapter 3

Continuous Signal Spaces and Perturbation Techniques

3.1 Discretization of Continuous Space: Off-Grid Problem

3.1.1 Continuous Parameter Spaces

In many practical systems, a signal of interest can be represented as a linear combination of several different signal sources. More precisely, assuming that there are K components, such a signal is written as:

$$y(t) = \sum_{i=1}^{K} \alpha_{T_i} \, \psi(\boldsymbol{\theta}_{T_i}, t) + n(t), \tag{3.1}$$

where $\psi(\boldsymbol{\theta}_{T_i})$ and α_{T_i} denotes the signal source and the effect of the source, respectively, for the i^{th} component. The measurement noise with known statistical properties is represented with the term $n(t)$.

Although the relation in (3.1) is valid for continuous time domain, in order to process the observations digitally, signal of interest should be sampled. For this purpose let $\boldsymbol{t} \in \Re^M$ be the vector holding the sampling times. In general, the vector $\boldsymbol{t}$ is constructed with equally distant points in order to get uniform samples. However for the rest of this thesis, no such condition will be imposed on the sampling instances and most of the time random sampling will be assumed.

When the continuous-time relation in (3.1) is sampled with the given sampling time vector $\boldsymbol{t}$, the resulting discrete-time relation is expressed as:

$$\boldsymbol{y} = \sum_{i=1}^{K} \alpha_{T_i} \, \boldsymbol{\psi}(\boldsymbol{\theta}_{T_i}; \boldsymbol{t}) + \boldsymbol{n}, \tag{3.2}$$

where $\boldsymbol{y} \in \mathcal{C}^M$ denotes the samples of the signal of interest, $\boldsymbol{n} \in \mathcal{C}^M$ denotes the sampled noise sequence and $\boldsymbol{\psi}(\boldsymbol{\theta}_{T_i}; \boldsymbol{t}) \in \mathcal{C}^M$ denotes the sampled signal source with parameter $\boldsymbol{\theta}_{T_i}$.

The main purpose in inverse problems is to identify the different components from their linear measurements. For a given set of measurements $\boldsymbol{y}$, assuming a linear relation as in (3.2), the inverse problem can be written as the following minimization problem:

$$\arg\min_{\alpha_i, \boldsymbol{\theta}_i} \left\| \boldsymbol{y} - \sum_{i=1}^{K} \alpha_i \, \boldsymbol{\psi}(\boldsymbol{\theta}_i; \boldsymbol{t}) \right\|_2 \quad \text{s.t.} \quad \boldsymbol{\theta}_i \in \mathcal{P}, \tag{3.3}$$

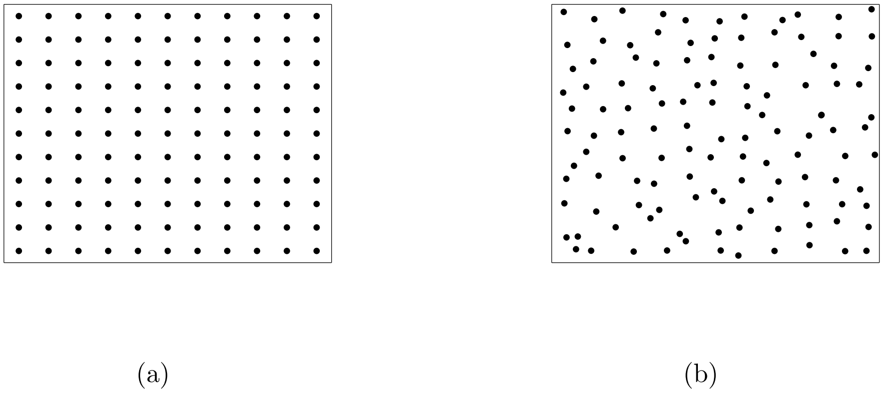
where $\mathcal{P}$ denotes the continuous domain of parameter of interest.

3.1.2 Use of Compressive Sensing Algorithms

In many signal processing applications, the signal of interest consists of linear combination of smaller number of sources, compared to the size of the measurements. This point rises the applicability of sparse signal processing techniques for the solution of the inverse problem in (3.3). As discussed in Chapter 2, Compressive Sensing theory provides a set of algorithms to recover a sparse signal $\boldsymbol{x}$ from the set of measurements in the form of $\boldsymbol{y} = \boldsymbol{A}\,\boldsymbol{x} + \boldsymbol{n}$.

The use of CS techniques for (3.3) is not straightforward due to the different constructions of the inverse problem in (3.3) and the sparse reconstruction problem in (2.9). CS techniques operate on a dictionary $\boldsymbol{A}$ and the main goal is to find a best sparse representation, which means selecting the best sub-matrix of $\boldsymbol{A}$. Even though the optimal solution for such a problem is overwhelmingly difficult, it is still a search problem in a finite-size solution set. On the other hand, the inverse problem in (3.3) operates on a continuous parameter space, hence more difficult to solve in general. From another point of view, the sparse recovery in classical CS theory can be thought of as a detection problem whereas the problem in (3.3) is an estimation problem.

In order to use sparse solvers for (3.3), construction of a dictionary is required. For this purpose, one can choose N different parameter points from the continuous domain $\mathcal{P}$ creating a set of parameters as $\mathcal{B} = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_N\}$. In the discretization process there is no assumption on the griding scheme. Even though general practice is to discretize the basis with equal spacing as in Fig. 3.1(a), arbitrary selection of parameters as in Fig. 3.1(b) is also an acceptable practice.



(a)

(b)

Figure 3.1: Discretization of the continuous parameter domain $\mathcal{P}$ with (a) uniform, (b) arbitrary griding.

Given the discrete parameter points, required dictionary $\boldsymbol{A}$ can be constructed as a collection of the signal sources evaluated at the discrete parameter points. Hence, i^{th} column of the dictionary, denoted as $\boldsymbol{a}_i$, is found as:

$$\boldsymbol{a}_i = \boldsymbol{\psi}(\boldsymbol{\theta}_i; \boldsymbol{t}), \tag{3.4}$$

where $\boldsymbol{t}$ denotes the vector of sampling times. Repeating (3.4) for each discrete point, the required dictionary can be constructed as:

$$\boldsymbol{A} = \begin{bmatrix}\boldsymbol{a}_1 \, \boldsymbol{a}_2 \, \dots \, \boldsymbol{a}_N\end{bmatrix} \in \mathcal{C}^{M\times N}. \tag{3.5}$$

With the construction of the dictionary as in (3.5), the linear continuous-parameter relation in (3.2) can be converted to the standard CS formulation as:

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{n}, \tag{3.6}$$

where $\boldsymbol{x}$ is the vector holding the weight coefficients of the signal sources, that is i^{th} element of $\boldsymbol{x}$ is α_{T_i} if $\boldsymbol{\theta}_i = \boldsymbol{\theta}_{T_i}$. Since there are N discrete parameter points, $\boldsymbol{x}$ is of length N, i.e. $\boldsymbol{x} \in \mathcal{C}^N$, however, the model in (3.2) includes only K components, i.e. $\|\boldsymbol{x}\|_0 = K$. Having $K < M < N$, the relation in (3.6) is identical to one in (2.8), hence any sparse recovery algorithm studied in CS theory can be utilized to solve it.

3.1.3 The Off-Grid Problem

Even though the discretization of the parameter space provides a way of converting the continuous-domain problem into the regular CS formulation, the model in (3.6) can be substituted for the actual relation in (3.2) under a very fundamental assumption: the actual parameters, $\boldsymbol{\theta}_{T_i}$, of the relation must coincide with one of the selected discrete points $\boldsymbol{\theta}_i$. Unless this assumption of parameters being “on-grid” holds true, model in (3.6) is a mere approximation to the main problem in (3.2).

Since the actual parameters of the system can be located at any point in the continuous domain, they typically do not coincide with the selected grid points $\boldsymbol{\theta}_i$. If the actual parameters are assumed to be located randomly in the continuous space, then on-grid assumption holds true with probability zero. Since this requirement of being "on-grid" cannot be satisfied, its well-known effects are named as the "off-grid problem".

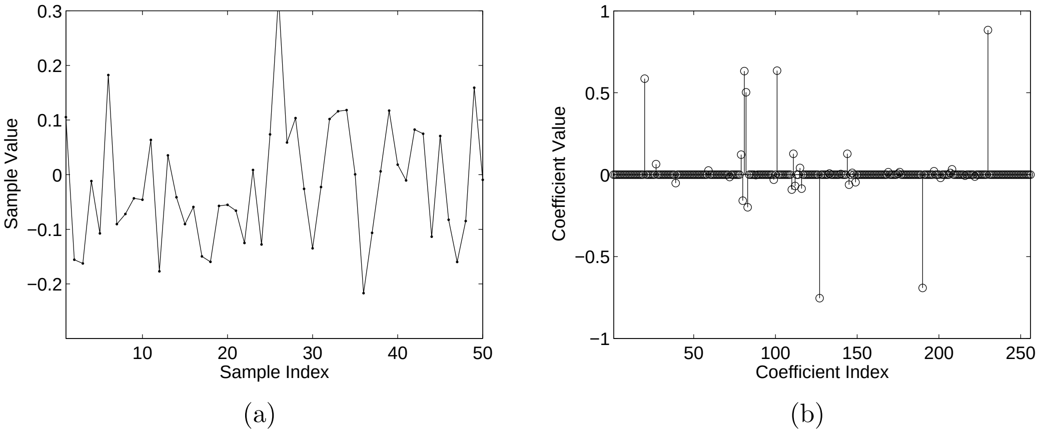


Figure 3.2: (a) A signal composed with off-grid parameters, (b) its corresponding solution in the IDCT domain defined in (2.3) with OMP.

The root cause of the off-grid problem is that a signal with a sparse decomposition in a continuous parameter domain may not have a sparse decomposition in the discretized domain. Let us demonstrate this problem on the IDCT basis given in (2.3) with the sensing matrix given in Fig. 2.3(a). Except for the first column, dictionary $\boldsymbol{A}$ can be thought of as a collection of cosines with the following form:

$$\boldsymbol{a}_k = \sqrt{\frac{2}{N}} \cos\left(2\,\pi\, f_k\, \boldsymbol{t}\right), \tag{3.7}$$

where $\boldsymbol{a}_k$ denotes the k^{th} column of $\boldsymbol{A}$, $f_k = (k-1)/2$ and $\boldsymbol{t} \in \Re^{50}$ is the sampling times in the range [0, 1]. From this perspective, the signal in Fig. 2.1 is a linear combination of cosines at frequencies 9.5, 40, 50, 63, 94.5, 114.5 all in Hertz, all of which are on-grid. If the component at 40Hz was at 40.2Hz, which is off the chosen grid, the signal sensed through $\boldsymbol{\Phi}$ and the resulting sparse recovery by OMP are provided in Fig. 3.2. Even though the decimated time domain signal in Fig. 3.2(a) has a 6-sparse representation in the continuous frequency domain, sparsity cannot be revealed in the provided reconstruction.

Due to the need for discretization, applications including target localization [18], radar [19, 20], time delay and Doppler estimation, beamforming [21, 22] or shape detection [23] are all subject to off-grid problems in the compressive sensing framework. In the literature, the effect of this basis mismatch has been observed and analyzed in some applications such as radar [19, 24] and beamforming [25].

In problems due to parameter space discretization, a simplistic approach is to use multi-resolution refinement and decrease the grid size. As long as there is a set of finite number of discrete parameters, decreasing the grid size is not a direct solution to the off-grid problem. Also it increases the coherence between dictionary columns, violating the Restricted Isometry Property (RIP) [26] and other coherency depended recovery guarantees.

3.2 Parametric Perturbations

In this section of the thesis, a novel solution technique to the forenamed off-grid problem is presented. Proposed algorithm is based on the OMP and it employs controlled perturbations on the selected parameters, hence columns, so that better fit to the observation signal is achieved [44]. The algorithm is simulated on delay-Doppler radar formulation and successful sparse recovery performances are observed. This study is also presented in [45].

3.2.1 PPOMP Algorithm

The main purpose of the presented PPOMP algorithm is to find a solution to the inverse problem in (3.3) under the compressive sensing framework. Therefore, it requires a dictionary $\boldsymbol{A}$ constructed via discretization of the continuous parameter space. Main perturbation stage uses a gradient-descend based optimization technique to find the best perturbations/updates for the selected support set.

Sparse representation of a signal of interest in a dictionary requires identification of parameter points at which the signal sources are present. This is equivalent to the identification of the support set of the signal among the columns of given dictionary $\boldsymbol{A}$. For this purpose OMP is utilized for the estimate of the support set. Interested reader may refer to Section 2.3.2.1 for the details of OMP.

Note that at the k^{th} iteration of the OMP algorithm, the measured signal can be represented in a k-sparse manner as a linear combination of the k support vectors as:

$$\boldsymbol{y} = \boldsymbol{y}_{\perp} + \sum_{i=1}^{k} \alpha_i \, \boldsymbol{a}(\boldsymbol{\theta}_i), \tag{3.8}$$

where $\boldsymbol{y}_{\perp}$ is the orthogonal residual of $\boldsymbol{y}$ within the span of the k chosen support vectors $\boldsymbol{a}(\boldsymbol{\theta}_i)$, $i = 1, \ldots, k$ and $\boldsymbol{a}(\boldsymbol{\theta}_i)$ is a column of dictionary $\boldsymbol{A}$ with grid parameters $\boldsymbol{\theta}_i$. Hence at each iteration of OMP, the vectors in the support set, their coefficients α_i, and the orthogonal residual, $\boldsymbol{y}_{\perp}$, are obtained.

In general, a signal source with parameter $\boldsymbol{\theta}_{T_i}$ may not be located at a grid node but is positioned within a grid area with an unknown perturbation from the closest grid node as:

$$\boldsymbol{\theta}_{T_i} = \boldsymbol{\theta}_i + \delta\boldsymbol{\theta}_i, \tag{3.9}$$

where $\boldsymbol{\theta}_i$ is the nearest grid node parameters, $|\delta\boldsymbol{\theta}_i| < \Delta_{\boldsymbol{\theta}}/2$ with $\Delta_{\boldsymbol{\theta}}$ defining the grid dimensions. If there were no noise, the measurement vector $\boldsymbol{y}$ would be in the span of $\boldsymbol{a}(\boldsymbol{\theta}_{T_i})$, but not in the span of $\boldsymbol{a}(\boldsymbol{\theta}_i)$. Our goal is to perturb the selected grid parameters, hence the corresponding column vectors in $\boldsymbol{A}$, so that a better fit to the measurements can be accomplished. This goal can be formulated as the following optimization problem:

$$\arg \min_{\alpha_i,\, \delta\boldsymbol{\theta}_i} \left\| \boldsymbol{y} - \sum_{i=1}^{k} \alpha_i \, \boldsymbol{a}(\boldsymbol{\theta}_i + \delta\boldsymbol{\theta}_i) \right\|_2 \quad \text{s.t.} \quad |\delta\boldsymbol{\theta}_i| < \Delta_{\boldsymbol{\theta}}/2. \tag{3.10}$$

Solution of the problem in (3.10) provides the perturbation parameters $\delta\boldsymbol{\theta}_i$ and the representation coefficients α_i for the selected set of k column vectors.

Assume that there exist a solver for the problem, namely $\mathbb{S}(\cdot)$, which takes the measurement vector $\boldsymbol{y}$ and the initial grid points, then returns the solution of the problem in (3.10). In an abstract sense, this solver can be written as:

$$\Big(\boldsymbol{\alpha}\ ,\ [\delta\boldsymbol{\theta}_1\ \dots\ \delta\boldsymbol{\theta}_k]\Big) = \mathbb{S}\Big(\boldsymbol{y}\ ,\ [\boldsymbol{\theta}_1\ \dots\ \boldsymbol{\theta}_k]\Big). \tag{3.11}$$

Note that solver $\mathbb{S}(\cdot)$ is not dependent on the OMP technique itself. Therefore, it is possible to integrate $\mathbb{S}(\cdot)$ into any algorithm that provides a suitable estimation of the grid points. In this study OMP is preferred due to its simplicity. When such a solver is utilized within the OMP iterations, an "ideal" parameter perturbed OMP (I-PPOMP) procedure, which is provided in Table 3.1, can be implemented.

Table 3.1: Ideal Parameter Perturbed-OMP (I-PPOMP) Algorithm

```
Inputs: $\big(\boldsymbol{y},\ \boldsymbol{A},\ \epsilon\big)$

Initialization: $\boldsymbol{y}_{\perp,0} = \boldsymbol{y}$, $\boldsymbol{T}_0 = \{\}$, $e = \|\boldsymbol{y}_{\perp,0}\|_2$, $k = 1$
Keep iterating until $e < \epsilon$
    $j^* = \arg\max_{1\le j\le N} |\boldsymbol{a}(\boldsymbol{\theta}_j)^H\ \boldsymbol{y}_{\perp,k-1}|\ /\ \|\boldsymbol{a}(\boldsymbol{\theta}_j)\|_2$
    $\boldsymbol{T}_k = \boldsymbol{T}_{k-1} \bigcup \{\boldsymbol{\theta}_{j^*}\}$
    $\Big(\boldsymbol{\alpha}\ ,\ [\delta\boldsymbol{\theta}_1\ \dots\ \delta\boldsymbol{\theta}_k]\Big) = \mathbb{S}\Big(\boldsymbol{y}\ ,\ \boldsymbol{T}_k\Big)$
    $\boldsymbol{y}_{\perp,k} = \boldsymbol{y} - \sum_{i=1}^{k}\ \alpha_i\ \boldsymbol{a}(\boldsymbol{\theta}_i + \delta\boldsymbol{\theta}_i)$
    $e = \|\boldsymbol{y}_{\perp,k}\|_2$
    $k = k + 1$

Output: $\Big(\boldsymbol{\alpha}\ ,\ [\delta\boldsymbol{\theta}_1\ \dots\ \delta\boldsymbol{\theta}_k]\ ,\ \boldsymbol{T}_k\Big)$
```

Since the optimization problem defined in (3.10) is non-convex, it may not be possible to obtain an ideal solver as specified in (3.11). Hence we propose to use a gradient descent optimization of the cost function in (3.10). Therefore starting from the grid nodes, selected parameters will be gradually perturbed until a convergence criteria is met. To simplify the iterations further α_i's and $\delta\boldsymbol{\theta}_i$'s will be sequentially updated in the following way.

First initialize $\boldsymbol{\theta}_{i,1} = \boldsymbol{\theta}_i$, $i = 1, \ldots, k$, to grid centers and obtain an initial representation coefficient vector $\boldsymbol{\alpha}_1$ as:

$$\boldsymbol{\alpha}_1 = \arg\min_{\boldsymbol{\alpha}} \left\| \boldsymbol{y} - \sum_{i=1}^{k} \alpha_i \, \boldsymbol{a}(\boldsymbol{\theta}_{i,1}) \right\|_2 . \tag{3.12}$$

Starting from $l = 1$, until convergence, perform updates:

$$\boldsymbol{\theta}_{i,l+1} = \boldsymbol{\theta}_{i,l} + \delta\boldsymbol{\theta}_{i,l},$$

where l represents the perturbation index, i represents the target index and

$$\left[\delta\boldsymbol{\theta}_{1,l} \ldots \delta\boldsymbol{\theta}_{k,l}\right] = \arg \min_{\delta\theta_i : |\delta\theta_i| \leq \Delta_\theta/2} \left\| \boldsymbol{y} - \sum_{i=1}^{k} \alpha_{i,l} \, \boldsymbol{a}(\boldsymbol{\theta}_{i,l} + \delta\boldsymbol{\theta}_i) \right\|_2 , \tag{3.13a}$$

$$\boldsymbol{\alpha}_l = \arg\min_{\boldsymbol{\alpha}} \left\| \boldsymbol{y} - \sum_{i=1}^{k} \alpha_i \, \boldsymbol{a}(\boldsymbol{\theta}_{i,l}) \right\|_2 . \tag{3.13b}$$

The problem defined in (3.13b) is a standard least squares formulation, however obtaining solution to the constrained nonlinear optimization problem in (3.13a) is not practical for many applications. Linearization of the cost function in (3.13a) around $\boldsymbol{\theta}_{i,l}$ significantly reduces the complexity of the optimization. For this purpose, $\boldsymbol{a}(\boldsymbol{\theta}_{i,l} + \delta\boldsymbol{\theta}_i)$ can be approximated by using the first order Taylor series as:

$$\boldsymbol{a}(\boldsymbol{\theta}_{i,l} + \delta\boldsymbol{\theta}_i) \approx \boldsymbol{a}(\boldsymbol{\theta}_{i,l}) + \nabla_{\boldsymbol{\theta}} \, \boldsymbol{a}(\boldsymbol{\theta}_{i,l}) \, \delta\boldsymbol{\theta}_i . \tag{3.14}$$

where $\nabla_{\boldsymbol{\theta}} \, \boldsymbol{a}(\boldsymbol{\theta}_{i,l}) \in \mathcal{C}^{M \times p}$ is the Jacobian of $\boldsymbol{a}(\boldsymbol{\theta}_{i,l})$ with respect to parameter vector $\boldsymbol{\theta}$ and p is the number of parameters, i.e. $\boldsymbol{\theta} \in \Re^{p \times 1}$.

By using (3.14), and ignoring the constraints on the perturbations, problem in (3.13a) can be re-written as:

$$\left[\delta\boldsymbol{\theta}_{1,l} \ldots \delta\boldsymbol{\theta}_{k,l}\right] = \arg\min_{\boldsymbol{u}} \left\| \boldsymbol{r}_l - \boldsymbol{B}_l \, \boldsymbol{u} \right\|_2 , \tag{3.15}$$

where $\boldsymbol{r}_l = \boldsymbol{y} - \sum_{i=1}^{k} \alpha_{i,l} \; \boldsymbol{a}(\boldsymbol{\theta}_{i,l})$ is the orthogonal residual from the least squares in (3.13b), $\boldsymbol{B}_l \in \mathcal{C}^{M\times pk}$ is the matrix holding the weighted Jacobians at the linearization point and is defined as:

$$\boldsymbol{B}_l = \Big[\alpha_{1,l}\ \nabla_{\boldsymbol{\theta}}\, \boldsymbol{a}(\boldsymbol{\theta}_{1,l})\ \boldsymbol{D},\ \alpha_{2,l}\ \nabla_{\boldsymbol{\theta}}\, \boldsymbol{a}(\boldsymbol{\theta}_{2,l})\ \boldsymbol{D},\ \ldots\ \alpha_{k,l}\ \nabla_{\boldsymbol{\theta}}\, \boldsymbol{a}(\boldsymbol{\theta}_{k,l})\ \boldsymbol{D}\Big], \tag{3.16}$$

and $\boldsymbol{u} = [\delta\boldsymbol{\theta}_1{}^T,\, \delta\boldsymbol{\theta}_2{}^T,\, \ldots\, \delta\boldsymbol{\theta}_k{}^T]^T \in \mathcal{R}^{pk\times 1}$ is the dummy vector variable containing updates in the l^{th} iteration on the corresponding parameters. Each Jacobian in $\boldsymbol{B}_l$ is scaled with a diagonal matrix $\boldsymbol{D}$, with corresponding grid size of each parameter on the diagonal, so that corresponding updates become unitless. Notice that $\boldsymbol{B}_l$ is different in each iteration since the linearization points $\boldsymbol{\theta}_{i,l}$ are updated. A new linearization is made at each updated parameter point.

Due to errors in linearization, direct solution of (3.15) will not produce the desired parameter perturbations. Instead we adapt a gradient descent type algorithm to solve (3.15) and take a small step in the direction that decreases the norm the most, i.e., direction of negative gradient. Then the new parameter point will be used in the next iteration and so on until the convergence. Let $J(\boldsymbol{u}) = \|\boldsymbol{r}_l - \boldsymbol{B}_l\, \boldsymbol{u}\|_2^2$ and negative of the gradient of J will be $-\nabla_{\boldsymbol{u}} J(\boldsymbol{u}) = 2\boldsymbol{B}_l^H(\boldsymbol{r}_l - \boldsymbol{B}_l\boldsymbol{u})$. Since we have intention of taking a small step from the linearization point, we need the gradient of $J(\boldsymbol{u})$ at $\boldsymbol{u} = \boldsymbol{0}$. Therefore, $-\nabla_{\boldsymbol{u}} J(\boldsymbol{u})|_{\boldsymbol{u}=\boldsymbol{0}} = 2\boldsymbol{B}_l^H\ \boldsymbol{r}_l$. Remember that both $\boldsymbol{B}_l$ and $\boldsymbol{r}_l$ are complex valued whereas perturbations need to be real. When solution is forced to be real, step direction is found to be as $\mathrm{Re}\{-\nabla_{\boldsymbol{u}} J(\boldsymbol{u})|_{\boldsymbol{u}=0}\} = \mathrm{Re}\{2\boldsymbol{B}_l^H\ \boldsymbol{r}_l\}$. With this important modification, alternating gradient descend solution of the main problem in (3.10) can be written as;

$$\begin{aligned} \boldsymbol{\alpha}_l &= \Big[\boldsymbol{a}(\boldsymbol{\theta}_{1,l})\ \boldsymbol{a}(\boldsymbol{\theta}_{2,l})\ \ldots\ \boldsymbol{a}(\boldsymbol{\theta}_{k,l})\Big]^{\dagger}\ \boldsymbol{y}, && (3.17\text{a}) \\ \boldsymbol{\theta}_{i,l+1} &= \boldsymbol{\theta}_{i,l} + \boldsymbol{D}\ \boldsymbol{\Lambda}(\boldsymbol{\mu}_{i,l})\ \mathrm{Re}\{\boldsymbol{B}_l^H\ \boldsymbol{r}_l\}, && (3.17\text{b}) \end{aligned}$$

where $\boldsymbol{\mu}_{i,l}$ is the step size and $\boldsymbol{\Lambda}(\cdot)$ is a diagonal matrix with the specified vector on the diagonals. To keep the updated points within the grid, the algorithm will also check that the total perturbations will not exceed the grid size. (3.17) defines the main update iterations of the proposed gradient based perturbation solver (GS) - $\widehat{\mathbb{S}}(\cdot)$ for (3.10) which is summarized in Table 3.2. Notice that, when $\mathbb{S}(\cdot)$ in Table 3.1 is replaced with the $\widehat{\mathbb{S}}(\cdot)$, proposed PPOMP algorithm is obtained.

For the gradient based parameter perturbation solver in Table 3.2, one could make several selections for the stopping criteria and the step size μ. It is possible to monitor the residual, $\boldsymbol{r}_l$, during the iterations, and terminate the solver in the l^{th} iteration if the residual norm $\|\boldsymbol{r}_l\|_2$, or amount of decrease $\|\boldsymbol{r}_l\|_2 - \|\boldsymbol{r}_{l-1}\|_2$, or rate of decrease $\|\boldsymbol{r}_l\|_2/\|\boldsymbol{r}_{l-1}\|_2$ is below a certain threshold. It is also possible to observe the parameters $\boldsymbol{\theta}_{i,l}$ and terminate the iterations when $|\boldsymbol{\theta}_{i,l} - \boldsymbol{\theta}_{i,l-1}|$ is below a certain threshold. Also iterations can be terminated, when norm of the gradient $\|\boldsymbol{B}_l^H \boldsymbol{r}_l\|_2$ is smaller than a given threshold value or when a maximum iteration count is reached. Several metrics can also be used in conjunction with the stopping criteria. In the presented results, the iterations were terminated when the rate of decrease of the residual is less than a certain threshold.

Table 3.2: Proposed Solver $\widehat{\mathbb{S}}(\cdot)$

```
**Inputs**: $(\{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_k\}, \boldsymbol{y}, \boldsymbol{\mu})$

**Initialize**: $l = 0$, $\boldsymbol{\theta}_{i,0} = \boldsymbol{\theta}_i$ for $1 \le i \le k$

Until stopping condition met,
  $\boldsymbol{A}_l = \Big[\boldsymbol{a}(\boldsymbol{\theta}_{1,l})\ \boldsymbol{a}(\boldsymbol{\theta}_{2,l})\ \ldots\ \boldsymbol{a}(\boldsymbol{\theta}_{k,l})\Big]$,
  $\boldsymbol{\alpha}_l = \boldsymbol{A}_l^{\dagger}\,\boldsymbol{y}$,
  $\boldsymbol{r}_l = \boldsymbol{y} - \boldsymbol{A}_l\,\boldsymbol{\alpha}_l$,
  $\boldsymbol{B}_l = \Big[\alpha_{1,l}\,\nabla_{\boldsymbol{\theta}}\,\boldsymbol{a}(\boldsymbol{\theta}_{1,l})\,\boldsymbol{D},\ \ldots\ \alpha_{k,l}\,\nabla_{\boldsymbol{\theta}}\,\boldsymbol{a}(\boldsymbol{\theta}_{k,l})\,\boldsymbol{D}\Big]$,
  $\big[\boldsymbol{g}_{1,l},\ \ldots,\ \boldsymbol{g}_{k,l}\big] = \mathrm{Re}\{\boldsymbol{r}_l^H\,\boldsymbol{B}_l\}$,

  For all $i$, $1 \le i \le k$,
    $\boldsymbol{\theta}_{i,l+1} = \boldsymbol{\theta}_{i,l} + \boldsymbol{D}\,\boldsymbol{\Lambda}(\boldsymbol{\mu}_{i,l})\,\boldsymbol{g}_{i,l}^T$,
    Check if $\boldsymbol{\theta}_{i,l+1}$ is within grid
    $\delta\boldsymbol{\theta}_i = \boldsymbol{\theta}_{i,l+1} - \boldsymbol{\theta}_{i,0}$,

  $l = l + 1$,

**Output:** $(\boldsymbol{\alpha}_l, \{\delta\boldsymbol{\theta}_1, \delta\boldsymbol{\theta}_2, \ldots, \delta\boldsymbol{\theta}_k\})$
```

For the selection of step size μ there are several possibilities. It is possible to use a fixed step size μ, that is $\mu_{i,l} = \mu$. If μ is small enough, after sufficient number of iterations convergence can be achieved. A more efficient approach is to use step sizes with constant rate of decrease, that is $\mu_{i,l} = \gamma\, \mu_{i,l-1}$ where γ is fixed and $0 < \gamma < 1$. If the gradient of a function is Lipschitz continuous with a constant L, gradient descent steps converges to a local optima by using constant step size that satisfies $\mu < 2/L$ [38, 39]. In addition, line search techniques can also be used to select locally optimal step sizes that guarantees convergence with at least linear convergence rates [39].

The additional computational complexity of PPOMP algorithm compared to OMP is the calculation of the gradient directions, and this requires a matrix vector multiplication which can be performed significantly faster than solving constrained nonlinear problem in (3.13a).

3.2.2 Compressive Delay-Doppler Radar

Coherent radar systems transmit a sequence of pulses with known phases and processes the received echoes to perform clutter suppression and detection at each angle of interest. Excellent references on the operation of radar receivers are available in the literature [46, 47]. In this application, a classical pulse doppler radar with a co-located receiver and a transmitter is considered. Although it is not investigated in here, MIMO radar systems can also be considered within CS framework [22, 48]. Let radar transmits $s(t)$, a coherent train of N_p pulses:

$$s(t) = \sum_{i=0}^{N_p-1} p\big(t - i\, T_{PRI}\big)\, e^{j2\pi f_c t}, \tag{3.18}$$

where, $p(t)$ is the individual pulse waveform, T_{PRI} is the uniform pulse repetition interval and f_c is the radar carrier frequency. Assuming K dominant targets with delays of τ_{T_m} and Doppler shifts of ν_{T_m}, $1 \leq m \leq K$, the received signal following the baseband down-conversion can be expressed as:

$$y(t) = \sum_{m=1}^{K} \alpha_m \; s(t - \tau_{T_m}) \, e^{j2\pi \nu_{T_m} t} + n(t), \tag{3.19}$$

where α_m is the complex reflectivity of the individual targets and $n(t)$ is the measurement noise. The above relation between the received signal and target parameters are expressed in terms of the measurable quantities of delay and Doppler. These parameters are related to the range and radial velocity of the m^{th} target as:

$$\tau_{T_m} = \frac{2R_m}{c} \quad , \quad \nu_{T_m} = \frac{2f_c}{c} v_m, \tag{3.20}$$

where R_m is the range and v_m is the radial velocity of the m^{th} target. At this point, the common practice is to employ matched filtering to individual uniformly spaced samples of pulse returns and use FFT across the delay aligned matched filter outputs. This way the returns are compressed in time and frequency sequentially [47].

In compressive sensing formulation, a sampled version of the measurement relation given in (3.19) is adapted to a linear matrix-vector relationship in delay-Doppler domain. For this purpose 2 dimensional delay-Doppler domain which lies in the product space $[\tau_o, \tau_f] \times [\nu_o, \nu_f]$ must be discretized where initial and final values of τ_0 and τ_f are determined by the range and ν_0 and ν_f are determined by the velocity of the potential targets. Discretization generates a finite set of N target points $\mathcal{B} = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_N\}$, where each $\boldsymbol{\theta}_j$ representing a grid node of (τ_j, ν_j), hence length of the parameter vector $p = 2$. For each grid node $\boldsymbol{\theta}_j$ the data model can be calculated from (3.19) as:

$$\boldsymbol{\psi}_j = s(\boldsymbol{t} - \tau_j) \circ e^{j2\pi\nu_j \boldsymbol{t}}. \tag{3.21}$$

where $\boldsymbol{t} \in \Re^{N_t \times 1}$ is the vector holding the time samples and operator "$\circ$" corresponds to Hadamard product. N_t is the number of time samples.

Repeating (3.21) at each $\boldsymbol{\theta}_j = (\tau_j, \nu_j)$ generates the dictionary $\boldsymbol{\Psi}$ where the j^{th} column of $\boldsymbol{\Psi}$ is $\boldsymbol{\psi}_j$. The size of the dictionary $\boldsymbol{\Psi}$ becomes $N_t \times N$. If the true target parameters (τ_{T_m}, ν_{T_m}) falls exactly on the grid points (τ_j, ν_j) then a linear system of equations can be formed as:

$$\boldsymbol{y}_s = \boldsymbol{\Psi}\,\boldsymbol{x} + \boldsymbol{n}, \tag{3.22}$$

where $\boldsymbol{y}_s$ is the sampled measurement vector and $\boldsymbol{x}$ is a reflectivity vector defining the delay-Doppler space, i.e., if there is a target at $\boldsymbol{\theta}_j$, the value of the j^{th} element of $\boldsymbol{x}$ should be α_j, otherwise zero. If there are K targets in the scene then the vector $\boldsymbol{x}$ should be a K sparse vector, that is $\|\boldsymbol{x}\|_0 = K$.

In the CS formulation, a very small fraction of the samples obtained at the Nyquist rate carry enough information to represent a sparse signal. Thus a sub-Nyquist sampling can be done and a random subset of M measurements at random times t_s can be measured in CS. In general these new measurements can be represented as $\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{y}_s$ where $\boldsymbol{\Phi}$ is an $M \times N_t$, $M < N_t$ measurement matrix constructed by randomly selecting M rows of an $N_t \times N_t$ identity matrix. The general linear relation is then:

$$\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{\Psi}\,\boldsymbol{x} + \boldsymbol{n} = \boldsymbol{A}\,\boldsymbol{x} + \boldsymbol{n}. \tag{3.23}$$

When the sub-sampling is integrated into the data model, basis atom corresponding to parameter $\boldsymbol{\theta}_j = (\tau_j, \nu_j)$ can be written in a compact form as:

$$\boldsymbol{a}(\boldsymbol{\theta}_j) = s(\boldsymbol{t} - \tau_j) \circ e^{j2\pi\nu_j \boldsymbol{t}}, \tag{3.24}$$

where $\boldsymbol{t} \in \Re^{M\times 1}$ is the vector holding the random time samples. Hence, the required gradient computations and the Jacobian simplify further as:

$$\begin{aligned} \frac{\partial\,\boldsymbol{a}(\boldsymbol{\theta})}{\partial\,\tau}\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_j} &= e^{j2\pi\nu_j \boldsymbol{t}} \circ \frac{d}{d\tau} s(\boldsymbol{t}-\tau)\Big|_{\tau=\tau_j} = -e^{j2\pi\nu_j \boldsymbol{t}} \circ \frac{d\,s(t)}{d\,t}\Big|_{t=\boldsymbol{t}-\tau_j} \\ \frac{\partial\,\boldsymbol{a}(\boldsymbol{\theta})}{\partial\,\nu}\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_j} &= j2\pi\boldsymbol{t} \circ \boldsymbol{a}(\boldsymbol{\theta}_j) \end{aligned} \tag{3.25}$$

$$\nabla_{\boldsymbol{\theta}}\,\boldsymbol{a}(\boldsymbol{\theta}_j) = \left[\begin{array}{cc} \frac{\partial\,\boldsymbol{a}(\boldsymbol{\theta})}{\partial\,\tau}\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_j} & \frac{\partial\,\boldsymbol{a}(\boldsymbol{\theta})}{\partial\,\nu}\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_j} \end{array}\right]. \tag{3.26}$$

Note that for pre-computed and stored $ds(t)/dt$ values, calculation of these partial derivatives require only component-wise multiplication of vectors that has M multiplications each. Hence Jacobian can be computed very efficiently. For such construction of the Jacobian, the required diagonal normalization matrix $\boldsymbol{D}$ should be selected as:

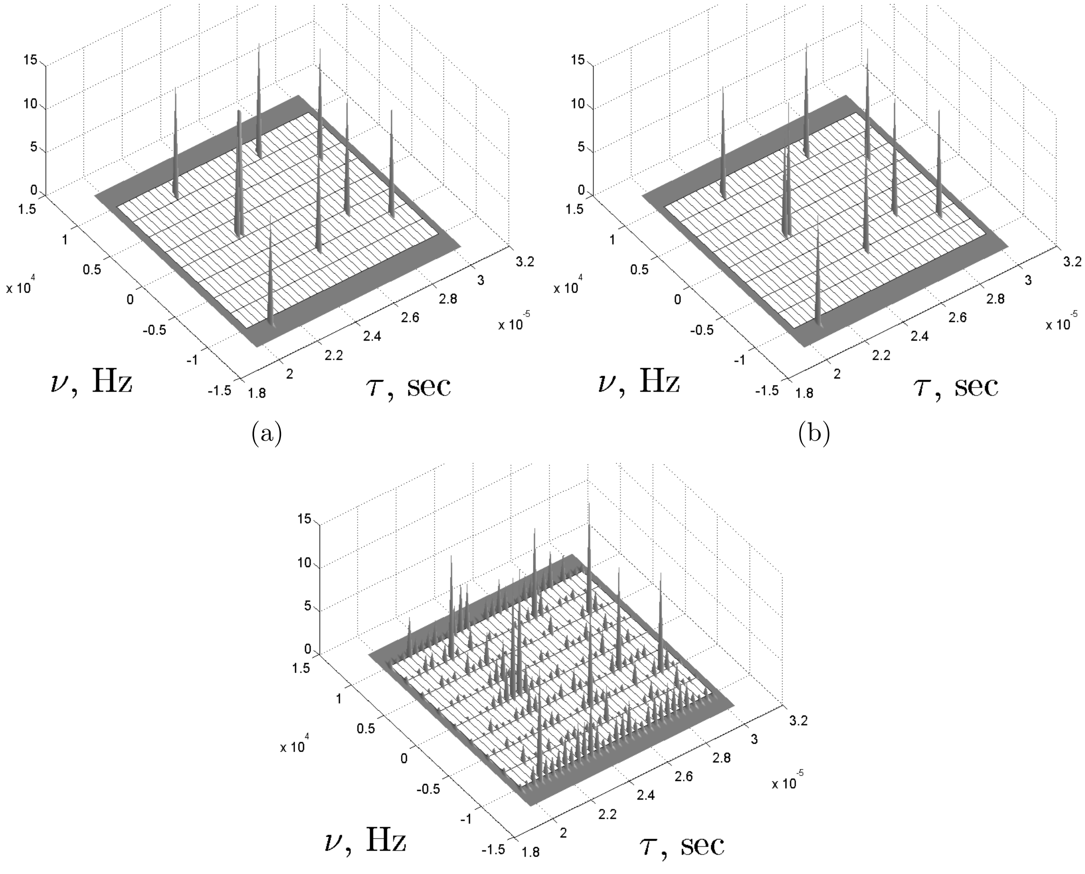
$$\boldsymbol{D} = \begin{pmatrix} \Delta_\tau & 0 \\ 0 & \Delta_\nu \end{pmatrix}, \tag{3.27}$$

where Δ_τ and Δ_ν are the Rayleigh resolution spacing in delay and Doppler, respectively.

As shown in Appendix A.1, for a delay-Doppler radar with a linear chirp signal, normalized form of the non-linear objective function in (3.13a) is Lipschitz continuous with $L = 10\,\pi^2$, therefore gradient descent stage of the proposed algorithm is guaranteed to converge to a local minima. In the presented results, step size is selected as $\mu_{i,l} = z\gamma_{i,l}$, where z is a pre-selected value of $z = 0.01$. For the i^{th} point, $\gamma_{i,l}$ is the ratio of the norm of the gradient in the l^{th} iteration to the maximum observed norm of the gradient through the perturbation iterations. As a result, $\gamma_{i,l} \leq 1$ and decreases as the norm of the gradient decreases. With this selection, smaller steps are taken as the gradient decreases when approaching a local minima. Notice that $\mu_{i,l} \leq 0.01 < 2/L \approx 0.02$, thus our selection of the step size is guaranteed to converge to a local minima.

3.2.2.1 Simulations

In the simulations, a classical single receiver-single transmitter pulsed-Doppler radar transmitting a linear chirp signal $p(t)$ with bandwidth of $B = 1.5MHz$ and pulse width of $T_p = 20\mu s$ is considered. In the coherent processing, a pulse train of $N_p = 8$ pulses are used with $T_{PRI} = 50\mu s$. The delay and Doppler space is chosen as the maximum unambiguous ranges of $[T_p,\ T_{PRI} - T_p]$ in delay and $[-1/(2\,T_{PRI}),\ 1/(2\,T_{PRI})]$ in Doppler. To create the forward linear model the space is discretized to grids with Rayleigh resolution spacing in both parameter axis which is $\Delta_\nu = 1/(N_p T_{PRI})$ in Doppler and $\Delta_\tau = 1/(2B)$ in delay. For the simulated case this discretization creates a total of $N = 279$ grid nodes. Sparse target scene is modeled as $K = 9$ point reflectors that are generated with delay and Doppler parameters randomly selected from the defined continuous delay-Doppler space where none of them exactly coincides with the chosen grid nodes. The complex reflectivity of the parameters are selected randomly with magnitudes selected from a normal distribution of $N(5, 1)$ and phases selected uniformly from $[0, 2\pi]$. For $M = 2N/3 = 186$ randomly spaced time samples in $[0,\ N_p T_{PRI}]$, the received signal is computed using (3.19). If the samples are taken at the Nyquist rate, total number of samples is $(N_p T_{PRI})(2B) = 1200$. Therefore M corresponds to only 15% of the Nyquist rate samples. Measurement noise corresponding to an SNR of 27.3dB is added to the computed time samples. Here SNR is defined as $20 \log_{10}(\|\boldsymbol{y}_0\|_2 / \|\sigma \boldsymbol{n}\|_2)$ where $\sigma \boldsymbol{n}$ is the noise component in the measurements.



(a)

(b)

(c)

Figure 3.3: (a) True delay-Doppler space reflectivity with $K = 9$ off the grid targets. Reconstruction results with (b) PPOMP, (c) OMP.

The actual target reflectivity and its reconstruction by the proposed PPOMP technique are shown in Figure 3.3(a) and (b), respectively. It can be seen that even for off-grid targets, PPOMP could provide accurate reconstruction of the sparse target scene. Note that PPOMP result is obtained in the absence of prior information about the target sparsity level. OMP technique using the same measurements and the same termination criteria with PPOMP generated the result shown in Fig. 3.3(c). Due to presence of off grid targets, OMP generates large number of significant peaks resulting in excessively many false target detections even at high level of detection threshold.

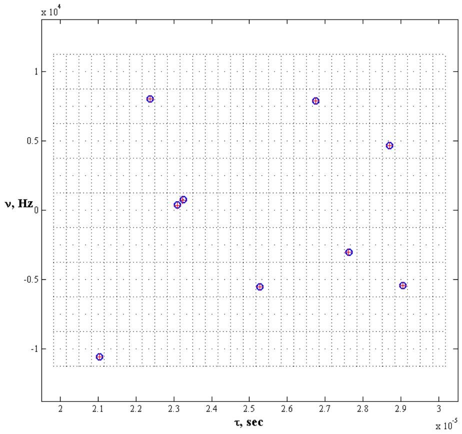


Figure 3.4: Actual and reconstructed target positions in the delay-Doppler domain. Circles ('o') correspond to the actual target parameters where plus signs ('+') correspond to the reconstructed target parameters by the proposed PPOMP technique.

Figure 3.4 shows the same simulation result as a 2D image with underlying grids and their centers. It can be seen that PPOMP could find all the target parameters very close to their actual values. Figure 3.5(a) shows the gradient based steps taken for one of the targets starting from the grid center. It can be seen that with decreasing step sizes the algorithm converges to the actual target parameters. Similarly, 3.5(b) shows gradient steps taken for two closely spaced targets. Note that the separation of these two targets is closer than a grid size corresponding to the classical Rayleigh resolution limit both in delay and Doppler axis. While a matched filter won't be able to resolve these two targets, the proposed PPOMP technique could detect and identify their actual parameters accurately. This shows the high resolution capability of the proposed PPOMP technique, which is an attribute of other sparse signal reconstruction techniques as well [49, 50]. Here this phenomenon is also observed for off-grid targets.

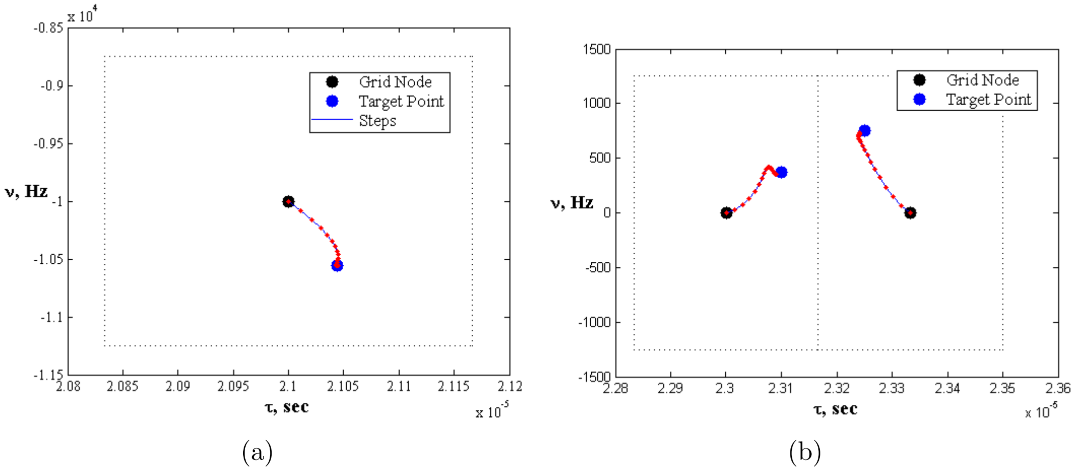


Figure 3.5: Gradient based steps taken within the PPOMP algorithm at (a) one of the target grids , with (b) two targets grids where the two target parameters are closer than a grid size in both τ and ν. Grid node corresponds to a discritized point as in (3.21) and Target Point corresponds to the actual off-grid target point.

3.2.2.2 How to Measure the Error?

One of the important problems of standard CS based reconstruction techniques is that in the presence of off-grid targets, they tend to generate non-sparse reconstructions. In such a case, the reconstruction error should be carefully defined. One approach would be to match the closest points in the correct scene and the reconstructed one, then compute the parameter error between them. However, when sparsity levels do not match, this error criterion is not appropriate. Hence it is a necessity to find a suitable metric in order to compare the parameter estimation performance of the sparse reconstruction techniques. Here, we propose to use Kullback-Leibler Divergence (KLD) between the actual and reconstructed target scenes, which is defined as follows:

$$D(p||q) \triangleq \int_{-\infty}^{\infty} p(\boldsymbol{x}) \, \ln\Big(\frac{p(\boldsymbol{x})}{q(\boldsymbol{x})}\Big) \, d\boldsymbol{x}, \tag{3.28}$$

where $p(\boldsymbol{x})$ and $q(\boldsymbol{x})$ are probability density functions of the corresponding scenes. Even though a given target scene has no probabilistic behavior, we can consider it as a 2-Dimensional Gaussian Mixture Model(GMM), where each mixture element has the following covariance matrix:

$$\boldsymbol{C} = \begin{pmatrix} {\Delta_\tau}^2 & 0 \\ 0 & {\Delta_\nu}^2 \end{pmatrix}, \tag{3.29}$$

where Δ_τ and Δ_ν are the resolutions of delay-Doppler grid. Hence, if a scene has K targets with parameters τ_{T_i} and ν_{T_i}; reflection coefficients with α_i for $i = 1, \ldots, K$, then we define its corresponding GMM as:

$$p(\boldsymbol{x}) = \sum_{i=1}^{K} \frac{\alpha'_i}{2\pi \, |\boldsymbol{C}|^{\frac{1}{2}}} \exp\Big(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_i)^T \, \boldsymbol{C}^{-1} \, (\boldsymbol{x} - \boldsymbol{\mu}_i)\Big), \tag{3.30}$$

where α'_i are the normalized coefficients such that $\alpha'_i = |\alpha_i| / \sum_j |\alpha_j|$ and $\boldsymbol{\mu}_i$ are the corresponding delay-Doppler parameters such that $\boldsymbol{\mu}_i = [\tau_{T_i} \; \nu_{T_i}]^T$. Using the definition as in (3.30), $p(\boldsymbol{x})$ becomes a valid pdf, hence KLD defined in (3.28) can be used.

For a single Gaussian, a closed form of the KLD is available in terms of defining parameters. However, for GMM, there is no closed form solution of the integral in (3.28). An efficient approximation can be obtained by using Monte Carlo techniques since KLD defined in (3.28) can also be considered as an integral to compute the expectation of $\ln\Big(\frac{p(\boldsymbol{x})}{q(\boldsymbol{x})}\Big)$ under the distribution of $p(\boldsymbol{x})$. Therefore, it can be written as:

$$D(p||q) \triangleq \int_{-\infty}^{\infty} p(\boldsymbol{x}) \, \ln\Big(\frac{p(\boldsymbol{x})}{q(\boldsymbol{x})}\Big) \, d\boldsymbol{x} = E_{p(\boldsymbol{x})}\Big[\ln\Big(\frac{p(\boldsymbol{x})}{q(\boldsymbol{x})}\Big)\Big]. \tag{3.31}$$

Sample mean can be used to approximate the actual value of the expectation relying on the law of large numbers as:

$$D(p||q) \simeq \frac{1}{Z} \sum_{j=1}^{Z} \ln \left(\frac{p(\boldsymbol{x}_j)}{q(\boldsymbol{x}_j)} \right), \tag{3.32}$$

where each $\boldsymbol{x}_j$ is drawn independently and identically from $p(\boldsymbol{x})$ as defined in (3.30). In the following simulations, each KLD is approximated by using $Z = 10^6$ samples.

3.2.2.3 Average Performance

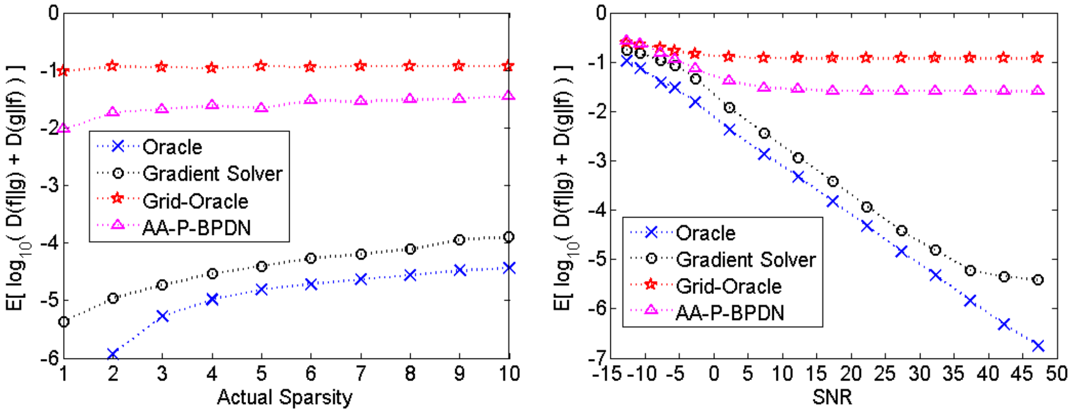
First, we would like to define an oracle estimator that would be the lower bound for the reconstruction performance of the tested algorithms. We assume that oracle estimator exactly knows the off-grid target parameters and hence the oracle estimation of the coefficients is given as a Least-Squares(LS) solution as:

$$\widehat{\boldsymbol{x}} = \left[\boldsymbol{a}(\tau_{T_1}, \nu_{T_1}) \; \boldsymbol{a}(\tau_{T_2}, \nu_{T_2}) \; \ldots \; \boldsymbol{a}(\tau_{T_K}, \nu_{T_K}) \right]^{\dagger} \boldsymbol{y}. \tag{3.33}$$

We also define a "grid-oracle" solution, which is the LS solution not on the actual parameter points, but on the grid points closest to the actual off-grid ones. This provides the lower bound for the unperturbed techniques.

To illustrate the performance of the proposed GS algorithm defined in Table 3.2, reconstructions are compared with the oracle solution , grid-oracle solution and AA-P-BPDN algorithm proposed in [32] while actual sparsity is changing from 1 to 10 at a fixed SNR of 27.3dB and measurement number of $M = 186$. In this simulation all techniques are given the grid points closest to the actual ones, basically to measure how well the proposed perturbation technique and alternatives can estimate the actual parameters given the correct grid points compared to oracle performances. For a fair comparison, AA-P-BPDN is modified and ℓ_1 reconstruction in the iterations is replaced with the LS solution on the given grid location. Iterations in AA-P-BPDN is terminated when normalized norm of the difference between the solutions in two consecutive iterations are smaller than 10^{-2}. It is expected that both algorithms will be better than the grid-oracle due to their perturbation mechanism, yet oracle solution will still be the lower bound.

For the error metric, KLD as defined in (3.32) is used. Since KLD is not symmetric, we consider the difference between two radar scenes, namely p and q, as $d(p,q) = D(p||q) + D(q||p)$. For the average reconstruction performance, simulations were repeated 300 times at each sparsity level with independent delay-Doppler domain target scenes. The average of base-10 logarithm of the $d(p,q)$ distances is provided in Figure 3.6(a). It can be observed that as the sparsity level increases, reconstruction performance decreases. However, performance of the proposed gradient solver follows the oracle performance closely with similar performance gap for sparsity ranging from 3 to 10. AA-P-BPDN, on the other hand, has a better performance than the grid-oracle solution but significantly inferior than the proposed gradient solver and the oracle solution. This is partly because AA-P-BPDN linearizes the basis functions only on the grid points and the accuracy of its approximation decreases for larger perturbations. However, GS updates its approximation at each iteration with a gradient descent update which converges to a local minima.



(a)

(b)

Figure 3.6: Mean of the KLD metric for tested techniques in comparison with the oracle result at different (a) sparsity levels, (b) SNR levels.

To investigate the effect of noise level, different SNR levels in the range of

-15dB to 50dB are tested for a fixed sparsity level of $K = 5$ and number of samples $M = 186$. Figure 3.6(b) shows the average KLD in the logarithmic scale for the tested algorithms. It can be observed that oracle KLD decreases linearly with increasing SNR. Proposed GS closely follows oracle performance with a small performance gap until an SNR up to 35dB and levels of after that due to termination of the iterations. Although AA-P-BPDN has a better performance than grid oracle, it performs worse than proposed GS for all SNR levels. Results presented in Figure 3.6 show that if the closest grid locations are given to the proposed perturbation procedure, a close performance to the oracle solution can be obtained. Now we would like to test the total PPOMP algorithm when the perturbation is done on the grid locations selected by OMP.

In the following simulations, performance of the proposed PPOMP algorithm is compared to AA-P-BPDN, standard OMP, ℓ_1 reconstruction and the oracle solution for varying sparsity, measurement number and SNR levels. At each test case 50 independent random trails are performed. Noise fit level for all suitable techniques is set to be $\epsilon = 1.3\,\|\sigma \boldsymbol{n}\|_2 / \|\boldsymbol{y}\|_2$.

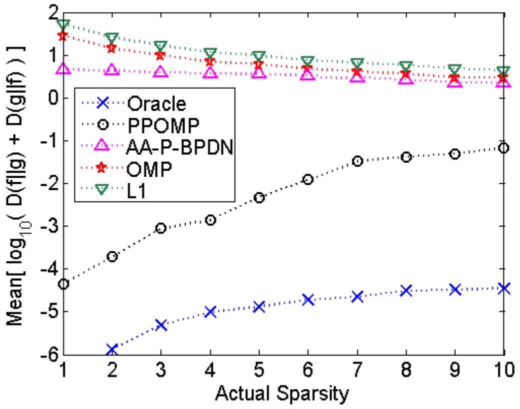


Figure 3.7: Mean of the KLD metric for tested techniques in comparison with the oracle result at varying sparsity levels.

Figure 3.7 shows the mean of logarithm base-10 of KLD metric for varying sparsity levels. PPOMP is closer to the oracle performance, whereas other techniques are significantly inferior compared to the proposed PPOMP technique. When compared to the small gap between the gradient solver and the oracle performance in Figure 3.6, gap between PPOMP and oracle solution in Figure

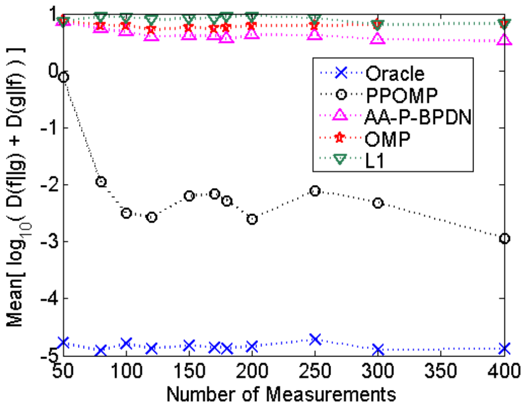


Figure 3.8: Mean of the KLD metric for tested techniques in comparison with the oracle result at varying number of measurements.

3.7 is larger. This is because OMP iterations within the PPOMP technique are not always able to provide correct grid points. This performance gap is more apparent in the less sparse region since OMP iterations are more prone to fail in that range. The same case is also valid for the AA-P-BPDN; AA-P-BPDN is inferior to its oracle counter-part. It is important to notice that since PPOMP and AA-P-BPDN have perturbation mechanism, both are superior to the classical unperturbed techniques.

Figure 3.8 shows a similar comparison for a range of number of measurements. It can be seen that after a minimum required number of measurements which seems to be around 100 for this case, performance of PPOMP do not increase with the increase in number of measurements. Even if there is performance gap between the PPOMP and the oracle solution, PPOMP is significantly superior to the other compared techniques.

Figure 3.9 shows the mean of logarithm base-10 of KLD for a range of SNR levels. While PPOMP performs closer to oracle for high SNR, its performance degrades and becomes similar to OMP or ℓ_1 for lower SNR regime. Even though the proposed GS is able to follow the oracle performance for varying SNR as in Figure 3.6(b), PPOMP performs worse in low SNR regime since OMP is not able to provide the correct grid points to the gradient solver in that regime. As a result, we observe a degraded performance in the overall PPOMP algorithm. At

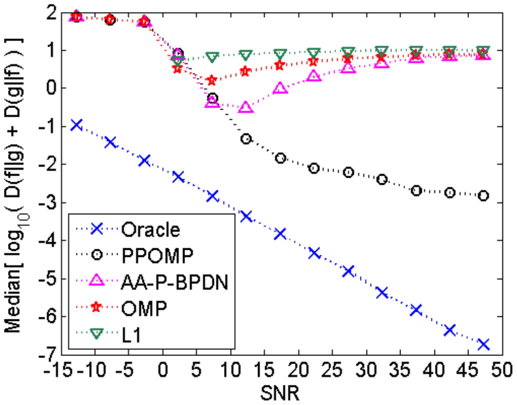


Figure 3.9: Mean of the KLD metric for tested techniques in comparison with the oracle result at varying SNR levels.

lower SNR, impressions due to off-grid error are washed out by noise, hence results of all the investigated techniques are at about the same level of performance. For the investigated application whose results are shown in Figure 3.9, the benefit of using the proposed technique becomes noticeable beyond SNR of 10dB, which is commonly encountered in practice. Also notice that in the analysis, AA-P-BPDN performs better than both OMP and ℓ_1 minimization due to its perturbation scheme. However, the performance gap between AA-P-BPDN and PPOMP is significant for high SNR.

In addition to the higher performance of PPOMP, it is also less computationally complex compared to AA-P-BPDN. Since in each iteration of AA-P-BPDN, one ℓ_1 optimization and one constraint least-squares problems are are solved, the computational complexity is significantly higher than the proposed PPOMP technique. The reported simulations performed on a workstation with Intel E5450 processor using CVX toolbox [39]. While AA-P-BPDN takes approximately 360 seconds in average for a single reconstruction, PPOMP converges in approximately 37 seconds in average for the same test problem of $K = 5$, $M = 186$, $SNR = 27.3$dB.

3.3 General Perturbations

Section 3.1 discusses sparse representation in a continuous parameter spaces and the off-grid problem introduced in the process of discretization of the continuous space. A novel solution to this type of problem is also presented in Secion 3.2. It is important to notice that the off-grid problem is a type of the more general basis mismatch problem. In addition to these structured perturbations, random time jitter in A/D conversion, modeling errors in construction of the dictionary $\boldsymbol{A}$ create perturbations on the dictionary columns. Hence, in general, the signal $\boldsymbol{y}$ will be sparse in an unknown basis $\widehat{\boldsymbol{A}} = \boldsymbol{A} + \boldsymbol{P}$ where $\boldsymbol{A}$ is the adopted dictionary and $\boldsymbol{P}$ is the unknown perturbation matrix.

In this section, a novel Perturbed Orthogonal Matching Pursuit (POMP) algorithm, that recovers unstructured perturbations, is presented. In the proposed POMP algorithm, controlled perturbation mechanism is applied on the selected columns. The selected column vectors are perturbed in directions that decrease the orthogonal residual at each iteration. Proven limits on perturbations are obtained. The proposed method is fast, simple to implement and successful in recovering sparse signals under random basis perturbations. A preliminary form of this approach has been presented in [51] and full version is published in [52].

3.3.1 Perturbation by Rotation

When $\boldsymbol{A}$ is not known precisely, $\boldsymbol{y}$ may not be sparse in the assumed $\boldsymbol{A}$. However, sparsity of the signal can be revealed under a certain perturbation, $\delta\boldsymbol{A}$, on the given $\boldsymbol{A}$. In this case, the ℓ_0 problem can be recast as,

$$\min \|\boldsymbol{x}\|_0 \qquad \text{s.t.} \qquad \min_{\delta\boldsymbol{A} \in \boldsymbol{\Delta}} \left\|\boldsymbol{y} - (\boldsymbol{A} + \delta\boldsymbol{A})\,\boldsymbol{x}\right\|_2 < \epsilon, \tag{3.34}$$

where $\boldsymbol{\Delta}$ is some bounded perturbation space. This problem can be viewed as a generalized version of the problem given in [31], in which $\delta\boldsymbol{A}$ is considered as a Taylor series or polar approximation of $\boldsymbol{A}$ and $\boldsymbol{\Delta}$ is the sufficient limits, also relaxing ℓ_0 with ℓ_1. The solution to this general ℓ_0 problem is also combinatoric in nature, and infeasible in practice.

To reduce the complexity of the ℓ_0 problem of (3.34), sub-optimal greedy techniques can be developed as well. In these greedy approaches, the support set of the reconstruction is iteratively increased until the constraints are satisfied. Assuming that at iteration k, the support set contains k columns of $\boldsymbol{A}$, which we will call as $\boldsymbol{S}_k = \{\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_k\}$ for simplicity. At the $(k+1)^{th}$ iteration, new vector $\boldsymbol{s}_{k+1}$ is obtained from the solution of the following optimization problem:

$$\boldsymbol{s}_{k+1} = \arg \max_{\boldsymbol{u} \in \boldsymbol{S}'_k} \left\{ \max_{\delta \boldsymbol{S}_k, \delta \boldsymbol{u}} \left\| \boldsymbol{P}_{\left([\boldsymbol{S}_k\ \boldsymbol{u}]+[\delta \boldsymbol{S}_k\ \delta \boldsymbol{u}]\right)}\ \boldsymbol{y} \right\|_2 \text{s.t.} \quad [\delta \boldsymbol{S}_k\ \delta \boldsymbol{u}] \in \Delta \right\}, \quad (3.35)$$

where matrix $\boldsymbol{P}$ is the projection operator to the column space of perturbed $[\boldsymbol{s}_1, \ldots, \boldsymbol{s}_k, \boldsymbol{u}]$ and $\boldsymbol{S}'_k = \boldsymbol{A} \setminus \boldsymbol{S}_k$ is the set of all basis vectors that are not contained in $\boldsymbol{S}_k$.

For each $\boldsymbol{u} \in \boldsymbol{S}'_k$, this perturbation problem can be solved by using the technique given in [53]. However, due to its associated gradient descent based iterations, the complexity of solution is still a practical limitation for large N. In this work, we propose a simpler non-iterative perturbation for each $\boldsymbol{s}_i \in \boldsymbol{S}_k$, to maximize the projection under bounded perturbations. At any iteration k, the measurement $\boldsymbol{y}$ can be decomposed as:

$$\boldsymbol{y} = \boldsymbol{y}_{\perp} + \boldsymbol{y}_{//}, \quad (3.36)$$

where $\boldsymbol{y}_{//}$ is the projection of $\boldsymbol{y}$ onto the span of vectors in $\boldsymbol{S}_k$ and $\boldsymbol{y}_{\perp}$ is the orthogonal residual. Since vectors in $\boldsymbol{S}_k$ are linearly independent, this projection can be uniquely expressed as:

$$\boldsymbol{y}_{//} = \sum_{i=1}^{k} \alpha_i\, \boldsymbol{s}_i, \quad (3.37)$$

where α_i is the weight of the corresponding i^{th} column vector. In the proposed approach, as shown in Fig. 3.10, the $\boldsymbol{s}_i$'s will be perturbed by rotating them towards to $\widehat{\boldsymbol{y}_\perp}$ by an angle ϕ_i, where $\widehat{\boldsymbol{y}_\perp} = \boldsymbol{y}_\perp / \|\boldsymbol{y}_\perp\|_2$ is the normalized residual:

$$\boldsymbol{s}_{i,p}(\phi_i) = \boldsymbol{s}_i \cos(\phi_i) + \widehat{\boldsymbol{y}_\perp}\, \mathrm{sgn}(\alpha_i) \sin(\phi_i). \tag{3.38}$$

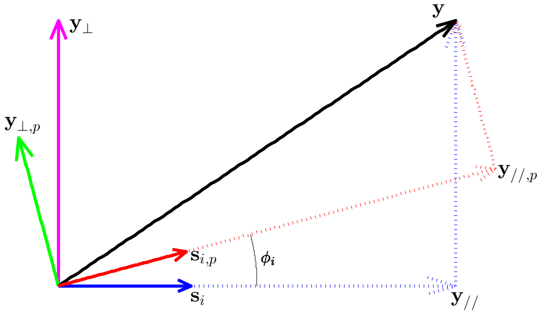


Figure 3.10: One dimensional example for perturbed and unperturbed column vectors. As the basis vector rotates, residual decreases.

Since both $\boldsymbol{s}_i$ and $\widehat{\boldsymbol{y}_\perp}$ have unit norms and they are orthogonal to each other, $\boldsymbol{s}_{i,p}$, which is the perturbed version of $\boldsymbol{s}_i$, has also unit norm. If the angle of rotation, or equivalently, the allowed amount of perturbation on $\boldsymbol{s}_i$ is large enough, then $\boldsymbol{s}_{i,p}$ can be aligned with $\boldsymbol{y}$ and there will be no residual left in the perturbed basis. If the $\boldsymbol{s}_i$'s are rotated more than adequate, $\boldsymbol{s}_{i,p}$'s can overlap with each other. When more than one vector span the overlapping region, uniqueness of the projection is lost. In order to avoid such overlaps, rotation of $\boldsymbol{s}_i$ should be limited to the half of the minimum angle between $\boldsymbol{s}_i$ and other $\boldsymbol{s}_j$ for $j \neq i$. More precisely, the maximum perturbation angle, ϕ_{max}, for a vector $\boldsymbol{a}_i$ should satisfy:

$$\cos(2\phi_{max}) \leq \mu(\boldsymbol{A}), \tag{3.39}$$

where $\mu(\boldsymbol{A})$ is the mutual coherence of $\boldsymbol{A}$ defined in (2.21) and $\phi_{max} = \cos^{-1}\big(\mu(\boldsymbol{A})\big)/2$. This case is illustrated in Fig. 3.11, where the maximum allowed perturbation of a vector is such that the cones around the columns of $\boldsymbol{A}$ do not overlap with each other. Perturbations beyond this limit generate switch-over between the chosen set of vectors and cause non-unique projection. Therefore, at each step of the proposed approach, only perturbations satisfying this limit will be considered.

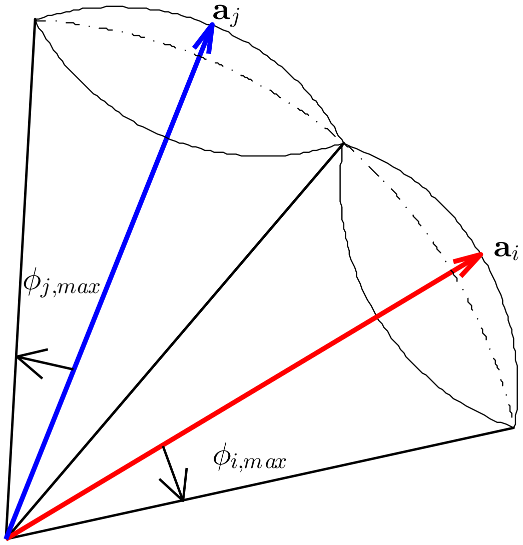


Figure 3.11: Each unit column of $\boldsymbol{A}$ has a maximum perturbation angle so that the perturbed vectors do not overlap with each other.

The approach of perturbations embedded in the iterations of Orthogonal Matching Pursuit(OMP) provides a practical technique for sparse reconstruction when basis mismatch is present. In the following discussion, we will provide a detailed theoretical investigation of this approach that will be referred to as Perturbed-OMP, or POMP in short.

3.3.2 Theoretical Investigation and POMP Algorithm

One immediate question is the optimality of the proposed perturbation scheme, which is answered by the following theorem:

Theorem 1. *Given any perturbation angle satisfying,* $0 \leq \phi_i \leq \frac{\pi}{2}$*, as long as* $\sum_{i=1}^{k} |\alpha_i| \tan(\phi_i) \leq \|\boldsymbol{y}_\perp\|_2$ *is satisfied, optimal perturbation direction for* $\boldsymbol{s}_i$ *is* $\widehat{\boldsymbol{y}_\perp} \operatorname{sgn}(\alpha_i)$.

Proof. Perturbations according to arbitrary directions $\boldsymbol{u}_i$ with $\|\boldsymbol{u}_i\|_2 = 1$ can be written as:

$$\boldsymbol{s}_{i,p}(\phi_i) = \boldsymbol{s}_i \cos(\phi_i) + \boldsymbol{u}_i \sin(\phi_i). \tag{3.40}$$

therefore, $\boldsymbol{s}_i$ can be written as:

$$\boldsymbol{s}_i = \boldsymbol{s}_{i,p}(\phi_i)\sec(\phi_i) - \boldsymbol{u}_i \tan(\phi_i). \tag{3.41}$$

Replacing this decomposition of $\boldsymbol{s}_i$ in terms of $\boldsymbol{s}_{i,p}$ and $\boldsymbol{y}_\perp$ in Eq. 3.37, measurements can be written as:

$$\boldsymbol{y} = \boldsymbol{y}_\perp + \sum_{i=1}^{k} \alpha_i \bigg(\boldsymbol{s}_{i,p}(\phi_i)\sec(\phi_i) - \boldsymbol{u}_i \tan(\phi_i) \bigg), \tag{3.42}$$

which can be regrouped to obtain:

$$\boldsymbol{y} = \sum_{i=1}^{k} \alpha_i\, \boldsymbol{s}_{i,p}(\phi_i)\ \sec(\phi_i) + \boldsymbol{y}_\perp - \sum_{i=1}^{k} \boldsymbol{u}_i\, \alpha_i\ \tan(\phi_i). \tag{3.43}$$

First part of (3.43) is in the span of the perturbed basis. In order to achieve the minimum residual, norm of the remaining terms should be minimized. Hence optimal perturbation directions can be found as the solution of the following problem:

$$\min_{\boldsymbol{u}_i} \left\| \boldsymbol{y}_\perp - \sum_{i=1}^{k} \boldsymbol{u}_i\, \alpha_i\ \tan(\phi_i) \right\|_2^2 \quad \text{s.t.} \quad \|\boldsymbol{u}_i\|_2 = 1, \tag{3.44}$$

which has a unique global minimum at $\boldsymbol{u}_i = \dfrac{\boldsymbol{y}_\perp}{\|\boldsymbol{y}_\perp\|_2}\,\mathrm{sgn}(\alpha_i\ \tan(\phi_i))$ if $\displaystyle\sum_{i=1}^{k} |\alpha_i|\ \tan(\phi_i) \leq \|\boldsymbol{y}_\perp\|_2$ is satisfied. Since $\tan(\phi_i)$ is positive for ϕ_i of interest, optimal rotation directions can be written as:

$$\boldsymbol{u}_i = \frac{\boldsymbol{y}_\perp}{\|\boldsymbol{y}_\perp\|_2}\,\mathrm{sgn}(\alpha_i). \tag{3.45}$$

□

With the optimal selection of the perturbation directions, following theorem guarantees the decrease in the residual after the perturbation stage.

Theorem 2. *For the largest perturbation angle satisfying* $\phi_i < \phi_{max} < \frac{\pi}{2}$ *and* $\sum_{i=1}^{k} |\alpha_i| \tan(\phi_i) \leq \|\boldsymbol{y}_\perp\|_2$*, the perturbed support vectors* $\boldsymbol{s}_{i,p}$ *defined in Eq. 3.38, has an orthogonal residual* $\boldsymbol{y}_{\perp,p}$ *whose* ℓ_2 *norm is upper bounded as:* $\|\boldsymbol{y}_{\perp,p}\|_2 \leq \|\boldsymbol{y}_\perp\|_2 \left(1 - \frac{1}{\left\|\boldsymbol{y}_\perp\right\|_2} \sum_{i=1}^{k} |\alpha_i| \tan(\phi_i)\right)$.

Proof. From (3.38), similar to (3.41) and (3.42), observation vector can be decomposed as:

$$\boldsymbol{y} = \boldsymbol{y}_\perp \left(1 - \frac{1}{\left\|\boldsymbol{y}_\perp\right\|_2} \sum_{i=1}^{k} |\alpha_i| \tan(\phi_i)\right) + \sum_{i=1}^{k} \alpha_i \, \boldsymbol{s}_{i,p}(\phi_i) \sec(\phi_i). \tag{3.46}$$

In this decomposition of $\boldsymbol{y}$, the second term is in the span of the perturbed vectors $\boldsymbol{s}_{i,p}$. Therefore, the orthogonal decomposition of $\boldsymbol{y}$ in the perturbed basis is:

$$\boldsymbol{y} = \boldsymbol{y}_{\perp,p} + \boldsymbol{y}_{//,p}, \tag{3.47}$$

where $\boldsymbol{y}_{//,p}$ is the projection of $\boldsymbol{y}$ onto the span of the perturbed basis vectors, and $\boldsymbol{y}_{\perp,p}$ is the corresponding residual. Thus,

$$\boldsymbol{y}_{\perp,p} = \boldsymbol{P}_{[\boldsymbol{s}_{i,p},\ldots,\boldsymbol{s}_{k,p}]} \, \boldsymbol{y}_\perp \left(1 - \frac{1}{\left\|\boldsymbol{y}_\perp\right\|_2} \sum_{i=1}^{k} |\alpha_i| \tan(\phi_i)\right). \tag{3.48}$$

Since, the ℓ_2 norm of the projection operation is less than one, we have:

$$\begin{aligned} \|\boldsymbol{y}_{\perp,p}\|_2 &\leq \left\| \boldsymbol{y}_\perp \left(1 - \frac{1}{\left\|\boldsymbol{y}_\perp\right\|_2} \sum_{i=1}^{k} |\alpha_i| \tan(\phi_i)\right) \right\|_2 \\ &= \|\boldsymbol{y}_\perp\|_2 \left| 1 - \frac{1}{\left\|\boldsymbol{y}_\perp\right\|_2} \sum_{i=1}^{k} |\alpha_i| \tan(\phi_i) \right|. \end{aligned} \tag{3.49}$$

From the hypothesis, $\sum_{i=1}^{k} |\alpha_i| \tan(\phi_i) \leq \|\boldsymbol{y}_\perp\|_2$, thus we obtain the desired upper bound:

$$\|\boldsymbol{y}_{\perp,p}\|_2 \leq \|\boldsymbol{y}_\perp\|_2 \left(1 - \frac{1}{\left\|\boldsymbol{y}_\perp\right\|_2} \sum_{i=1}^{k} |\alpha_i| \tan(\phi_i)\right), \text{ q.e.d.} \tag{3.50}$$

□

The proof of Theorem 2 also reveals the following fact:

Corollary 1. *The upper bound derived in Theorem 2 is monotonically decreasing as a function of* ϕ_i *as long as the condition* $\sum_{i=1}^{k} |\alpha_i| \tan(\phi_i) \leq \|\boldsymbol{y}_\perp\|_2$ *is satisfied.*

Proof. In Theorem 2, it is shown that:

$$\|\boldsymbol{y}_{\perp,p}\|_2 \leq \|\boldsymbol{y}_\perp\|_2 \left(1 - \frac{1}{\big\|\boldsymbol{y}_\perp\big\|_2} \sum_{i=1}^{k} |\alpha_i| \tan(\phi_i)\right).$$

As long as $\sum_{i=1}^{k} |\alpha_i| \tan(\phi_i) \leq \|\boldsymbol{y}_\perp\|_2$, the right hand side is a valid non-negative upper bound for $\|\boldsymbol{y}_{\perp,p}\|_2$, and is a continuous function of all $\phi_i < \phi_{i,max}$, $1 \leq i \leq k$, with partial derivatives:

$$\frac{-|\alpha_i|}{1 + \tan^2(\phi_i)} < 0, \qquad 1 \leq i \leq k, \tag{3.51}$$

which are all strictly negative for $|\alpha_i| \neq 0$. Thus, the upper bound decreases monotonically as a function of ϕ_i, $1 \leq i \leq k$, q.e.d.

□

The first implication of Corollary 1 is that once a candidate support set is available and $\sum_{i=1}^{k} |\alpha_i| \tan(\phi_i) \leq \|\boldsymbol{y}_\perp\|_2$, rather than searching for a perturbation that minimizes $\|\boldsymbol{y}_{\perp,p}\|_2$, one can minimize its upper bound simply by perturbing each vector in the support up to its allowed limit.

The second implication of Corollary 1 is that if the perturbation angles are increased, i.e., more freedom for the adjustment of the basis exists, the derived upper bound decreases. Therefore, for a certain amount of perturbation, this raises the possibility of driving the upper bound to zero.

Corollary 2. *Let* $\phi_i = \phi^*$, $1 \leq i \leq k$. *Then if* $\phi^* = \tan^{-1}\left(\dfrac{\|\boldsymbol{y}_\perp\|_2}{\|\boldsymbol{\alpha}\|_1}\right)$, *perturbation of the support vectors up to* ϕ^*, *results in* $\|\boldsymbol{y}_{\perp,p}\|_2 = 0$, *yielding a* k*-sparse reconstruction with no residual error.*

Proof. Simply replace $\tan(\phi_i) = \tan(\phi^*) = \dfrac{\|\boldsymbol{y}_\perp\|_2}{\sum_{i=1}^{k} |\alpha_i|}$ in the upper bound given in the Eq. 3.50 to get:

$$0 \leq \|\boldsymbol{y}_{\perp,p}\|_2 \leq \|\boldsymbol{y}_\perp\|_2 \left(1 - \frac{1}{\|\boldsymbol{y}_\perp\|_2} \sum_{i=1}^{k} |\alpha_i| \, \tan(\phi^*)\right) = 0, \text{ q.e.d.} \tag{3.52}$$

□

It is important to note that Theorem 2 and Corollaries 1 and 2 are valid for any set of linearly independent k vectors from the columns of $\boldsymbol{A}$. Even though chosen subset $\boldsymbol{S}_k$ does not include any component from the correct support, this theorem guarantees that by using perturbation, the residual can be decreased. If $\boldsymbol{S}_k$ is the correct support, or some subset of the correct support, then $\boldsymbol{y}^* = \boldsymbol{S}_k {\boldsymbol{S}_k}^\dagger \boldsymbol{y}$ becomes a good approximation of $\boldsymbol{y}$. In this case, $\|\boldsymbol{y} - \boldsymbol{y}^*\|_2$ becomes smaller and hence ϕ^*, the angle of perturbation at which the upper bound becomes zero, is a smaller angle.

Proposition 1. *A perturbed set of support vectors* $\{\boldsymbol{s}_{i,p}(\phi), 1 \leq i \leq k\}$ *will have* $\|\boldsymbol{y}_{\perp,p}\|_2 = 0$ *if and only if the upper bound in Eq. 3.50 is zero.*

Proof. Using perturbations up to ϕ^*, the residual will be zero; hence we can expand $\boldsymbol{y}$ as:

$$\boldsymbol{y} = \sum_{i=1}^{k} \beta_i \, \boldsymbol{s}_{i,p}(\phi^*). \tag{3.53}$$

Assume there is another angle $\widetilde{\phi} < \phi^*$ resulting also a zero residual. Hence, we can expand $\boldsymbol{y}$ as:

$$\boldsymbol{y} = \sum_{i=1}^{k} \gamma_i \, \boldsymbol{s}_{i,p}(\widetilde{\phi}). \tag{3.54}$$

If we subtract Eq. 3.53 from Eq.3.54 term by term, we get the following:

$$\begin{aligned} 0 \;&=\; \sum_{i=1}^{k} \Big(\beta_i \, \boldsymbol{s}_{i,p}(\phi^*) \;-\; \gamma_i \, \boldsymbol{s}_{i,p}(\widetilde{\phi}) \Big), \qquad (3.55) \\ &=\; \sum_{i=1}^{k} \Big(\big(\beta_i \cos(\phi^*) - \gamma_i \cos(\widetilde{\phi})\big) \boldsymbol{s}_i + \big(\beta_i \sin(\phi^*) - \gamma_i \sin(\widetilde{\phi})\big) \widehat{\boldsymbol{y}_\perp} \operatorname{sgn}(\alpha_i) \Big). \end{aligned}$$

Since $\boldsymbol{s}_i$ and $\widehat{\boldsymbol{y}_\perp}$ are orthogonal to each other, $\beta_i \sin(\phi^*) = \gamma_i \sin(\widetilde{\phi})$ and $\beta_i \cos(\phi^*) = \gamma_i \cos(\widetilde{\phi})$. If we divide both equations term by term, we get $\tan(\phi^*) = \tan(\widetilde{\phi})$. Since we only consider acute angles, $\phi^* = \widetilde{\phi}$, which contradicts the assumption. Hence, the perturbation angle which results upper bound in Eq. 3.50 to be zero is unique, q.e.d.

□

Although in the proposed perturbation approach norm of the residual can be made zero for large enough perturbation angles, a relaxed constrained optimization problem, where $\|\boldsymbol{y}_{\perp,p}\|_2 < \epsilon$, is more appropriate when there is noise in $\boldsymbol{y}$. Under this relaxation, more sparse reconstructions can also be obtained. In Fig. 3.12 as an illustrative case, the error residual is shown as a function of perturbation angle. It is seen that the constraint line $\|\boldsymbol{y}_{\perp,p}\| = \epsilon$ intersects with the error norm curve at ϕ^*_ϵ, which is the first intersecting angle smaller than ϕ^*.

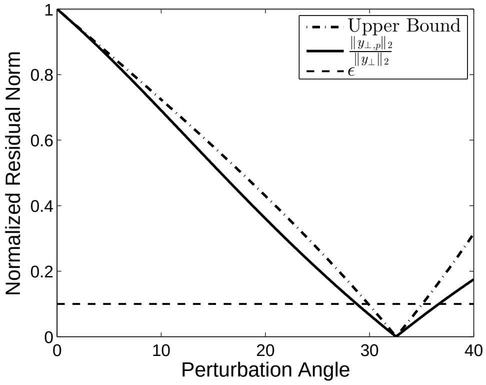


Figure 3.12: Upper bound and $\|\boldsymbol{y}_{\perp,p}\|_2/\|\boldsymbol{y}_\perp\|_2$ as a function of ϕ.

The following theorem provides an estimate for ϕ^* by using the intersection of $\|\boldsymbol{y}_{\perp,p}\|_2 = \epsilon$ with the upper bound in Eq. 3.50.

Theorem 3. *If the perturbation angles of support vectors in $\boldsymbol{S}_k$ are all chosen as $\phi^*_\epsilon = \tan^{-1}\left(\frac{\|\boldsymbol{y}_\perp\|_2 - \epsilon}{\|\boldsymbol{\alpha}\|_1}\right)$, where $\boldsymbol{y} = \boldsymbol{y}_\perp + \sum_{i=1}^{k} \alpha_i\, \boldsymbol{s}_i$, then $\|\boldsymbol{y}_{\perp,p}\|_2 \leq \epsilon$.*

Proof. If ϕ_i in Eq. 3.50 is replaced with ϕ^*_ϵ, then the upper bound becomes:

$$\begin{aligned}
\|\boldsymbol{y}_{\perp,p}\|_2 &\leq \|\boldsymbol{y}_\perp\|_2 \left(1 - \frac{1}{\big\|\boldsymbol{y}_\perp\big\|_2} \sum_{i=1}^{k} |\alpha_i| \, \tan(\phi^*_\epsilon)\right) \\
&= \|\boldsymbol{y}_\perp\|_2 \left(1 - \frac{\|\boldsymbol{\alpha}\|_1}{\|\boldsymbol{y}_\perp\|_2} \frac{\big(1 - \epsilon/\|\boldsymbol{y}_\perp\|_2\big)\|\boldsymbol{y}_\perp\|_2}{\|\boldsymbol{\alpha}\|_1}\right) \\
&= \|\boldsymbol{y}_\perp\|_2 \frac{\epsilon}{\|\boldsymbol{y}_\perp\|_2} = \epsilon, \text{ q.e.d.} \qquad (3.56)
\end{aligned}$$

□

Note that for small enough ϵ, the residual error of projection onto the perturbed support vectors, i.e., $\|\boldsymbol{y}_{\perp,p}\|_2$, the above bound will also be tight and $\|\boldsymbol{y}_{\perp,p}\|$ will be close to ϵ. Therefore, for the k^{th} iteration of the proposed POMP approach, there are three limits on the perturbation angles: first, ϕ_{max}, which is the limit predefined by the coherence of $\boldsymbol{A}$ as defined in Eq. 3.39; second ϕ^*_ϵ, beyond which the residual error norm is lower than ϵ; and third $\boldsymbol{\phi}'$, the user-defined limit. A user may want a solution with larger or smaller perturbations and, therefore may want $\phi_i \leq \phi'_i$. Hence, the allowable perturbation angle ϕ_i for the i^{th} support vector should be

$$\phi_i = \min\{\phi_{max}, \phi^*_\epsilon, \phi'_i\}. \tag{3.57}$$

In general, ϕ_{max} is a larger angle compared to other two. In the early phases of POMP, where the number of chosen support vectors are lower than actual sparsity level, ϕ^*_ϵ is expected to be larger than the allowed perturbation, ϕ'. Therefore, in the early phases of the iterations, the residual error norm of the perturbed support vectors is typically larger than the termination criterion ϵ. Yet, it is expected that as new columns are added to the current support during the iterations, ϕ^*_ϵ will decrease and will be less than ϕ' eventually. Once $\phi_i = \min\{\phi_{max}, \phi^*_\epsilon, \phi'_i\} = \phi^*_\epsilon$, iterations will stop, since after the perturbation the residual will have a norm less than ϵ. The following theorem, whose proof is provided in the Appendix A.2, states this expectation formally.

Theorem 4. *Let $\boldsymbol{S}_k$ be the support estimation in the k^{th} iteration and ϕ^*_k be the required perturbation angle as derived in Corollary 2 and let $\boldsymbol{s}_{k+1}$ be the basis vector chosen in the $(k+1)^{th}$ iteration. If $\|\boldsymbol{S}^\dagger_k \boldsymbol{s}_{k+1}\|_1 \leq 1 + \gamma$, then, $\phi^*_{k+1} < \phi^*_k$, where $\gamma > 0$.*

Since $\boldsymbol{A} \in \Re^{M \times N}$ is full rank, iterations stop when k reaches M without any perturbation in the worst case. Therefore, ϕ^*_M is zero. According to Theorem 4, when the required condition is satisfied, ϕ^*_k produces a monotonically decreasing sequence throughout the iterations. Since we know that $\phi^*_M = 0$, there exists a $K \leq M$ such that $\phi^*_K \leq \phi'$ due to monotonicity of the angles. Therefore, Theorem 4 indirectly guarantees the termination of iterations.

It is important to note that Tropp's Exact Recovery Condition for OMP [43], which simply states that if $\|\boldsymbol{S}^\dagger_k \boldsymbol{a}_i\|_1 \leq 1$ for all un-selected basis vectors, OMP will select one of the correct support vectors in the next iteration, is a stricter version of Theorem 4. Hence, we can safely guarantee that maximum required angle, ϕ^*_ϵ, will always decrease throughout the iterations as long as OMP is guaranteed to provide exact reconstruction. This point will be revisited with numerical simulations in Section 3.3.3.

Table 3.3: Perturbed-OMP (POMP) Algorithm

```
**Inputs**: $(\boldsymbol{y}, \boldsymbol{A}, \epsilon, \boldsymbol{\phi}')$

**Initialization**: $\boldsymbol{y}_{\perp,0} = \boldsymbol{y}$, $S_0 = \{\}$, $e = \|\boldsymbol{y}_{\perp,0}\|_2$, $k = 1$
Keep iterating until $e < \epsilon$

  **Orthogonal Pursuit:**
  $\boldsymbol{U}_k = \boldsymbol{A} \setminus \boldsymbol{S}_{k-1}$
  $j^* = \arg\max_j |\boldsymbol{u}_j^T \boldsymbol{y}_{\perp,k-1}|$
  $\boldsymbol{S}_k = \boldsymbol{S}_{k-1} \bigcup \{\boldsymbol{u}_{j^*}\}$
  $\boldsymbol{x}_k = \boldsymbol{S}_k^\dagger \boldsymbol{y}$
  $\boldsymbol{y}_{\perp,k} = \boldsymbol{y} - \boldsymbol{S}_k \boldsymbol{x}_k$

  **Perturbation Procedure:**
  If $\|\boldsymbol{y}_{\perp,k}\|_2 > \epsilon$
    $\widehat{\boldsymbol{y}_{\perp,k}} = \boldsymbol{y}_{\perp,k} / \|\boldsymbol{y}_{\perp,k}\|_2$
    $\phi_k^* = \tan^{-1}\left(\frac{\|\boldsymbol{y}_{\perp,k}\|_2 - \epsilon}{\|\boldsymbol{x}_k\|_1}\right)$
    $\boldsymbol{\phi}^k = \min(\boldsymbol{\phi}', \phi_k^*)$,
    For $i = 1$ to $k$;
      $\boldsymbol{s}_{i,p} = \boldsymbol{s}_i \cos(\phi_i^k) + \widehat{\boldsymbol{y}_{\perp,k}}\ \mathrm{sgn}(x_{k,i}) \sin(\phi_i^k)$
    $\boldsymbol{S}_{k,p} = \{\boldsymbol{s}_{i,p}, 1 \leq i \leq k\}$
    $\boldsymbol{y}_{\perp,k} = (\boldsymbol{I} - \boldsymbol{S}_{k,p}\ {\boldsymbol{S}_{k,p}}^\dagger)\, \boldsymbol{y}$
    $e = \|\boldsymbol{y}_{\perp,k}\|_2$
Return with $k = k + 1$.

**Output:** $\boldsymbol{x}^* = {\boldsymbol{S}_{k,p}}^\dagger \boldsymbol{y}$
```

The steps of the proposed Perturbed Orthogonal Matching Pursuit (POMP) algorithm is detailed in Table 3.3. Starting with a set of unit norm vectors, in the k^{th} iteration, POMP searches over the dictionary to find the vector providing the largest absolute inner product with the residual. After the selection of the new vector, POMP computes the projection of the measurement, $\boldsymbol{y}$, onto the new larger support and finds the residual. OMP continues with the next iteration here.

However, POMP proceeds with the perturbation. Given ϵ and ϕ', ϕ is computed according to Eq. 3.57. It is assumed that user provides an angle, ϕ', less than the mutual coherence of the basis, $\phi' \leq \cos^{-1}\big(\mu(\boldsymbol{A})\big)/2$, otherwise perturbations are limited by the basis itself. After that POMP starts to perturb each vector in the current support as given in Eq. 3.38. Then, the measurement vector, $\boldsymbol{y}$, is projected onto the perturbed support and the new residual is found. If the norm of the residual is less than ϵ, iterations are terminated, otherwise POMP continues with the next iteration.

One important characteristic of POMP is the promise of the sparser solutions. This property can be revealed as follows. Assume that POMP has produced a K-sparse solution. Iterations can be terminated due to two reasons. If the observation $\boldsymbol{y}$ is already sparse in $\boldsymbol{A}$, then at the K^{th} iteration we get $\|\boldsymbol{y}_\perp\|_2 \leq \epsilon$. Since OMP and POMP have the same selection criteria, OMP also chooses the very same $\boldsymbol{S}_K$ and obtains the same K sparse solution. However, if $\boldsymbol{y}$ is not sparse in $\boldsymbol{A}$, POMP obtains the K sparse solution using the perturbation. Since $\|\boldsymbol{y}_{\perp,p}\|_2 \leq \epsilon$ but $\|\boldsymbol{y}_\perp\|_2 > \epsilon$, OMP iterates at least one more time resulting a denser solution. Therefore, for any observation $\boldsymbol{y}$, OMP produces denser, or equally sparse at best, solutions than POMP.

OMP is preferred in many applications due to its computational efficiency. In the perturbation stage of the proposed algorithm, the inverse tangent operation can be well approximated by using tables and low order power series expansion, therefore the computational order is determined by the least-squares solution on the perturbed basis, which has a complexity of $O(k^2M)$ in the k^{th} iteration, which is the same as the standard OMP algorithm. If the algorithm terminates in the K^{th} iteration, overall complexity becomes $O(K^3M)$ for both POMP and OMP. In the worst case, algorithm will terminate eventually in the M^{th} iteration, which results in complexity of $O(M^4)$.

3.3.3 Performance of POMP

In this section, performance of POMP algorithm will be investigated and compared with alternative techniques under random basis mismatch. For this purpose, sparse reconstruction of sinusoids from their time samples will be considered. In this example, following dictionary vectors are used:

$$\boldsymbol{a}_i = \cos(2\pi f_i \boldsymbol{t}) \qquad 1 \leq i \leq N, \tag{3.58}$$

where f_i is the frequency of the i^{th} dictionary vector and $\boldsymbol{t}$ is the vector of time samples at which the signal is sampled. However, in practice, due to time jitter in the sampling, the observed signal $\boldsymbol{y}$ is not sampled at the nominal sampling times, resulting in a dictionary mismatch:

$$\boldsymbol{y} = \sum_{i=1}^{K} x_i \cos\left(2\pi f_i(\boldsymbol{t} + \boldsymbol{t}^j)\right) = \sum_{i=1}^{K} \boldsymbol{x}_i \, \widetilde{\boldsymbol{a}}_i, \tag{3.59}$$

where $\boldsymbol{t}^j$ can be modeled as a vector of independent and uniformly distributed random variables in the range of $[-\delta T/2, \delta T/2]$ where δT represents the level of jitter. The overall effect of time-jitter can be considered as a random perturbation on the dictionary $\boldsymbol{A} = [\boldsymbol{a}_1, \ldots, \boldsymbol{a}_N]$. In this more realistic scenario, $\boldsymbol{y}$ is typically non-sparse in the assumed dictionary $\boldsymbol{A}$.

To compare the overall performance in the above described set of simulations, the following metrics are used: the normalized signal reconstruction error, $\|\boldsymbol{x} - \boldsymbol{x}^*\|_2/\|\boldsymbol{x}\|_2$; the level of sparsity, $\|\boldsymbol{x}^*\|_0$; the distance between signal supports, $1 - \dfrac{|\boldsymbol{S}^* \cap \boldsymbol{S}|}{\max\{|\boldsymbol{S}^*|, |\boldsymbol{S}|\}}$; and the normalized residual norm, $\|\boldsymbol{y} - \boldsymbol{S}^* \, x^*\|_2/\|\boldsymbol{y}\|_2$, where $(\boldsymbol{x}, \boldsymbol{S})$ is the correct signal and its support, and $(\boldsymbol{x}^*, \boldsymbol{S}^*)$ is the obtained solution and its corresponding support [34].

3.3.3.1 Recovery from Time-Jittered Samples

In the first simulation, the dictionary is constructed by using frequencies $f_i = i \cdot \delta_f + 100$ Hz for $1 \leq i \leq 200$ and frequency separation of $\delta_f = 1$Hz. The sampling time vector $\boldsymbol{t}$ is created over the $[0,1]$ second interval with randomly chosen $M = 180$ time samples. A 25-sparse signal $\boldsymbol{x}$ is randomly generated with non-zero coefficients selected uniformly random from the range $[-10,-1]\cup[1,10]$. Observations, $\boldsymbol{y}$, are generated with time jitter of $\delta T = 8.3\mu$sec. The same termination criteria of $\epsilon = 10^{-3}$ is used for all compared techniques.

To select a proper maximum perturbation angle for POMP, the expected value of the perturbation angles are calculated. Based on the following, the normalized inner products of $\boldsymbol{a}_i$ and $\widetilde{\boldsymbol{a}}_i$:

$$\begin{aligned}\cos(\phi_i) &= \frac{\boldsymbol{a}_i^T\,\widetilde{\boldsymbol{a}}_i}{\|\boldsymbol{a}_i\|_2\,\|\widetilde{\boldsymbol{a}}_i\|_2} \\ &= \frac{\displaystyle\sum_{k=1}^{M} 0.5\,\cos\left(2\,\pi\,f_i\,(2t_k + t_k^j)\right) + 0.5\,\cos\left(2\,\pi\,f_i\,t_k^j\right)}{M\;\boldsymbol{a}_{i,RMS}\;\widetilde{\boldsymbol{a}}_{i,RMS}}.\end{aligned}$$

Since the sampling jitter is modeled as an i.i.d. sequence, $E\Big\{\dfrac{1}{M}\displaystyle\sum_{k=1}^{M} 0.5\,\cos\big(2\,\pi\,f_i\,(2t_k + t_k^j)\big)\Big\} \simeq 0$ and $E\big\{\boldsymbol{a}_{i,RMS}\big\} \simeq E\big\{\widetilde{\boldsymbol{a}}_{i,RMS}\big\} \simeq \sqrt{0.5}$ for large M. Therefore, the expected value of the cosine of the perturbation angles can be approximated as:

$$E\Big\{\cos(\phi_i)\Big\} \simeq \frac{E\Big\{0.5\cos(2\pi f_i t_k^j)\Big\}}{\sqrt{0.5}\,\sqrt{0.5}} = E\Big\{\cos(2\pi f_i t_k^j)\Big\}. \tag{3.60}$$

Using the small angle approximation of cosines, Eq. 3.60 can be further simplified as:

$$E\Big\{\cos(\phi_i)\Big\} \;\simeq\; E\Big\{1 - 2\,\pi^2\,f_i^2\,{t_k^j}^2\Big\} = 1 - 2\,\pi^2\,f_i^2\sigma_j^2,$$

where σ_j^2 is the jitter variance. Since jitter is assumed to be uniform in the interval $[-\delta T/2, \delta T/2]$, $\sigma_j^2 = \delta T^2/12$. Finally, using a small angle approximation for ϕ_i, we obtain,

$$E\big\{\phi_i\big\} \approx 60^\circ\,\sqrt{3}\,f_i\,\delta T. \tag{3.61}$$

Therefore, in the implementation of POMP, the allowed perturbation angles are selected according to Eq. 3.61 for each column, respectively. For the i^{th} column of $\boldsymbol{A}$ that corresponds to $f_i = i + 100$ Hz, the maximum perturbation angle is $\phi_i = (i + 100) \cdot 8.63° \cdot 10^{-4}$.

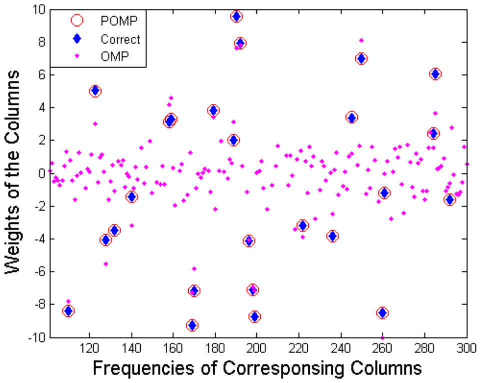


Figure 3.13: A realization of the reconstruction problem under time-jitter where OMP drastically fails and produces 167-sparse solution.

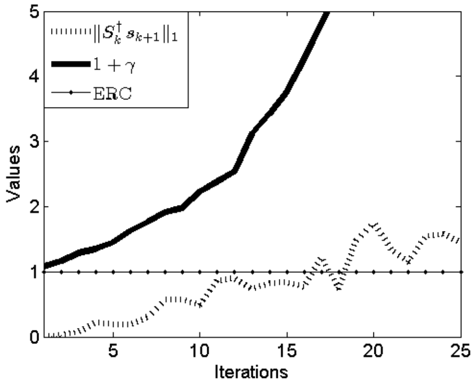


Figure 3.14: $\|\boldsymbol{S}_k^\dagger \, \boldsymbol{s}_{k+1}\|_1$ and its bound $(1 + \gamma)$ as a function of iterations. Even though ERC is not satisfied for $k > 16$, condition in Theorem 4 is satisfied due to large value of γ, ensuring the decrease of maximum perturbation angle during the iterations.

Figure 3.13 shows actual and reconstructed signal coefficients for OMP and POMP techniques for a random realization of jitter. In this specific case, while POMP correctly reconstructs the sparse signal coefficients, OMP generates a highly non-sparse signal. For this realization, Figure 3.14 shows $\|\boldsymbol{S}_k^\dagger \, \boldsymbol{s}_{k+1}\|_1$, the Tropp's Exact Recovery Condition (ERC) [43] and the proposed bound of Theorem 4. Even though ERC is not satisfied for $k > 16$, the proposed conditions of Theorem 4 are satisfied for this case. Therefore, this example shows that, the guarantees given by Theorem 4 provide a larger regime in which POMP is successful.

3.3.3.2 Simulated Performance and Theoretical Guarantees

In this section of the simulations, we will investigate performance of POMP when the condition of the Theorem 4 is not satisfied. For this purpose, we conduct a large set of Monte-Carlo simulations in excess of 10^6 trials in which the sparsity level K and number of measurement M are swept from 2 to 85 and K to 200, respectively. For each (K, M) pair, 100 cases are simulated. In each of these simulations, cases where the perturbation angle decrease, i.e. $\phi^*_{k+1} \leq \phi^*_k$, are identified as event E_1, and the sufficiency condition of Theorem 4 is satisfied, i.e. $\|\boldsymbol{S}_k{}^\dagger\, \boldsymbol{s}_{k+1}\|_1 \leq 1 + \gamma$, are identified as E_2. Out of all the runs at each sparsity level, the cases where only E_1 is valid and both E_1 and E_2 are valid are counted separately. In Figure 3.15, the statistics of the cases are shown for all the experiments conducted.

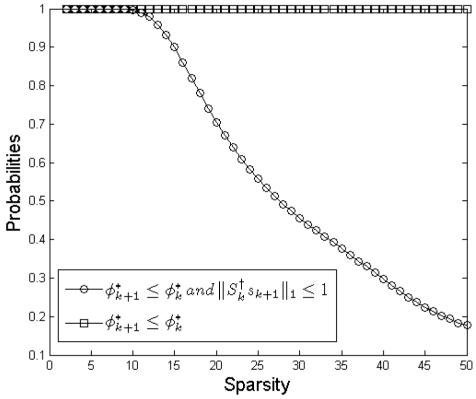
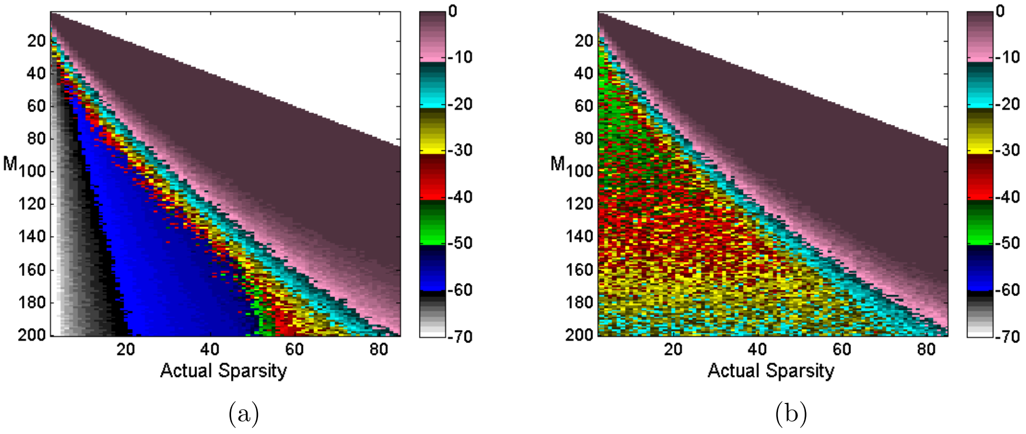


Figure 3.15: Empirical probabilities of E_1 is valid and (E_1, E_2) is jointly valid.

It can be observed that for $K \leq 10$, the case E_2 is satisfied with empirical probability of 1, and as sparsity level increases, the probability of the case E_2 being satisfied decreases. The maximum perturbation angle decreases consistently when the case E_2 is satisfied. Note that, Theorem 4 is only a sufficiency condition and in the simulated scenario for $K \geq 10$ its requirements are not guaranteed to met. Nevertheless, in all the simulations conducted, it is observed that the perturbation angle decreases.

3.3.3.3 Average Reconstruction Performance

In Figures 3.16 and 3.17, the average performance results obtained for OMP and POMP are shown. Figure 3.16 shows the normalized reconstruction error for both OMP and POMP in dB, while the corresponding support distances can be seen in Figure 3.17. For the same number of measurements and range of sparsity levels, it can be observed from Figure 3.16 that both OMP and POMP have similar phase transition curves; however POMP produces significantly lower reconstruction errors. In Figure 3.17, it can be seen that POMP provides more reliable supports as well.



(a)

(b)

Figure 3.16: Sparse signal reconstruction average error is in dB, $20\log_{10}\left(\frac{\|\boldsymbol{x}^{*}-\boldsymbol{x}\|_2}{\|\boldsymbol{x}\|_2}\right)$ as a function of measurement number and sparsity, (a) POMP, (b) OMP.

Instead of using perturbation procedure, one can try to use OMP in a finely discretized frequency domain. This way, the required perturbations on the dictionary vectors can be reduced. However, increasing the density of the basis also increases the coherence of the dictionary which adversely affects performance of

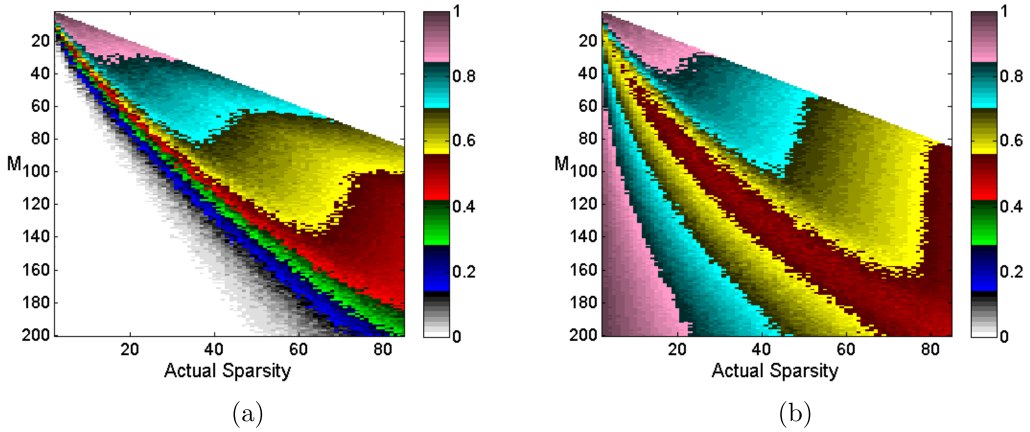
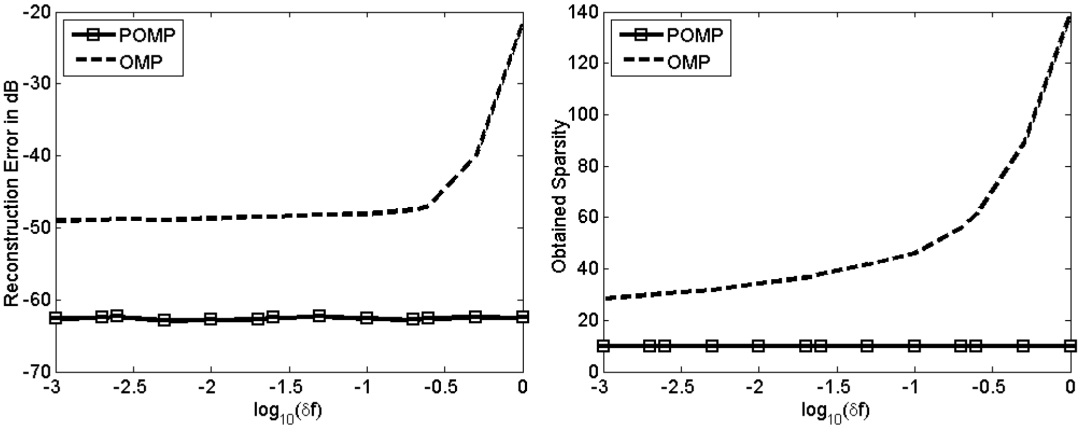


Figure 3.17: Average distances between actual and obtained supports, $1 - \dfrac{|\boldsymbol{S}^* \cap \boldsymbol{S}|}{\max\{|\boldsymbol{S}^*|, |\boldsymbol{S}|\}}$ as a function of measurement number and sparsity, (a) POMP, (b) OMP.

the CS techniques. In the simulated scenario, the effect of increasing frequency density in the dictionary is shown in Figure 3.18. In this case number of measurements is kept constant at $M = 180$, however the size of the dictionary is increased as $N = 199/\delta_f + 1$. The frequency range of $[101,\ 300]$Hz is considered in the simulations. Random time samples are chosen in $[0,\ 1/\delta_f]$ time interval with a jitter of $\delta T = 8.3\mu$sec time jitter. Compared to $\delta_f = 1$, when $\log_{10}(\delta_f) = -3$ is used, this corresponds to approximately 1000 times denser sampled frequency dictionary with $N = 199001$. Sparsity of the signal is kept constant at $K = 10$, and the SNR of the observed signal is kept at 60dB to better observe the effect of using a denser dictionary in the reconstructions. As seen in the Figure 3.18(a), using a denser basis can decrease the reconstruction error up to approximately -50 dB for OMP. However, beyond this point, OMP reaches an error floor. Also, as seen from Figure 3.18(b), obtained sparsity is reduced significantly. On the other hand, reconstruction performance of POMP is almost independent from the density of the dictionary. For all tested cases of N, POMP has a reconstruction error below -60 dB and yields a 10 sparse solution, that matches the actual sparsity. Note that using a denser basis significantly increases the mutual correlation of the dictionary vectors as well as the required number of inner-products. Therefore, rather than using OMP over a larger and denser dictionary, it is advisable to use POMP over a moderate size dictionary. Note that if the frequencies of the signal components were known precisely, the linearization based techniques proposed in [32,33,54] could have been used for the estimation of the jitter in the sampling times.



(a) (b)

Figure 3.18: Reconstruction performance with respect to density of the complex exponentials. (a) Sparse signal reconstruction error in dB, (b) Obtained sparsity level for POMP and OMP algorithms.

In this part of the simulations, we compare POMP with OMP, CoSaMP, Basis Pursuit, and Sparse Total Least Squares (S-TLS) [33] algorithms. Since POMP is based on OMP iterations, they have similar phase transition characteristics as given in Figure 3.16. Hence, in the following simulations, we will stay in the K/M regime in which OMP works successfully.

Unlike OMP and POMP, CoSaMP requires the correct sparsity level of the signal. Since, such information is not available in general, CoSaMP reconstructions obtained at all sparsity levels starting from $K = 1$ until the residual error is below the specified ϵ level.

Basis Pursuit (BP) which is also known as ℓ_1 reconstruction is implemented using the convex optimization toolbox CVX. To induce sparsity, hard thresholding is applied to BP reconstruction results. Let $\widehat{\boldsymbol{x}} = [\widehat{\boldsymbol{x}}_l^T \; \widehat{\boldsymbol{x}}_{l'}^T]^T$ be the BP reconstruction sorted in the absolute sense and $\widehat{\boldsymbol{x}}_l^T$ be the vector containing l largest coefficients. Then, in the reported results here, the threshold, τ, is selected as $\tau = \widehat{\boldsymbol{x}}(l)$ such that $\|\widehat{\boldsymbol{x}}_l\|_2 \geq (1 - 10^{-6})\|\widehat{\boldsymbol{x}}\|_2$.

In [33], two algorithms have been proposed for S-TLS; one finds the global optimum with highly demanding computational cost, and the other one is computationally more efficient but only guaranteed to converge to a local minima. Here, this more efficient technique which is called as coordinate descend (CD) based S-TLS is used for comparison. Since S-TLS has an iterative structure, the stopping criteria is important. To obtain a consistent comparison of algorithms, S-TLS is terminated when the residual error is below the specified ϵ at the end of an iteration. Rather than an all-zero vector, iterations are started with the solution of the obtained BP reconstruction to achieve faster convergence for S-TLS. In the following results, STLS-1 corresponds to $\lambda = 10^{-1}$ and STLS-2 corresponds to $\lambda = 10^{-2}$, where λ is the sparsity-tuning parameter of the CD based S-TLS algorithm.

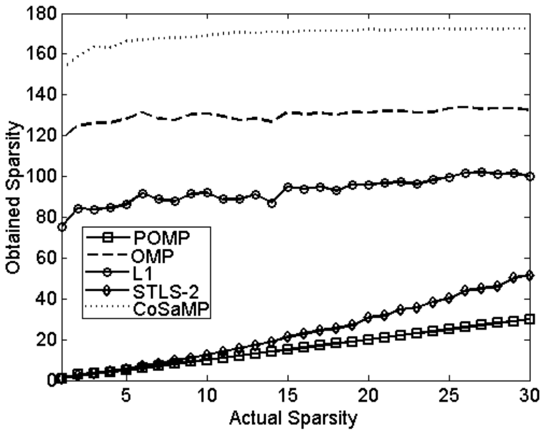


Figure 3.19: Obtained sparsity of the reconstructed signal as a function of actual sparsity.

For all the techniques compared, Figure 3.19 and 3.20 shows the obtained sparsity levels and support distances as a function of sparsity, respectively. POMP and S-TLS, being perturbation based techniques, can achieve the correct sparsity level for smaller K. As the actual sparsity increases, S-TLS fails to obtain the correct sparsity, whereas POMP successfully finds the support up to $K = 30$. Techniques that do not employ dictionary perturbation provided results that are significantly inferior than POMP and S-TLS at all sparsity levels. The performance of these techniques saturates and produce solutions at about the same sparsity, irrespective of the actual sparsity level. Since the observed signal is not sparse in the assumed dictionary $\boldsymbol{A}$ with $rank(\boldsymbol{A}) = M = 180$, OMP goes up to two-third of the rank whereas CoSaMP uses approximately 90% of the rank. On the other hand, POMP obtains the correct sparsity level and the support as shown in Figure 3.20. S-TLS gradually produces higher support distances due to denser solutions obtained in less sparse signals. Since techniques that do not employ perturbation has a saturated sparsity estimate, $\max\{|\boldsymbol{S}^*|, |\boldsymbol{S}|\} = |\boldsymbol{S}^*|$, and $|\boldsymbol{S}^* \cap \boldsymbol{S}|$ increases gradually as $|\boldsymbol{S}|$ increases. Therefore, their distance metric produces smaller values. However, this situation should not be considered as performance improvement for less sparse signals. It is because they simply fail to recover the correct support at all tested sparsity levels.

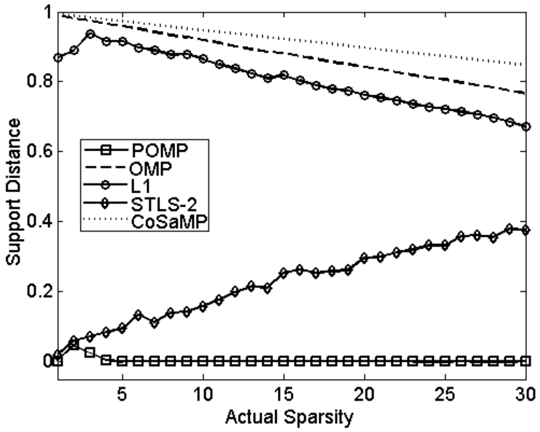


Figure 3.20: Distances between actual and obtained supports as a function of actual sparsity.

Figure 3.21 shows the average normalized reconstruction error in dB as a function of true sparsity for each of the compared reconstruction techniques. As expected, BP performs better than OMP. However, even though S-TLS employs perturbation, it has very similar performance to BP. POMP, on the other hand, has a significantly lower, -20dB, signal reconstruction error as compared to S-TLS and BP. CoSaMP has the worst performance among all the compared techniques with significantly higher reconstruction errors.

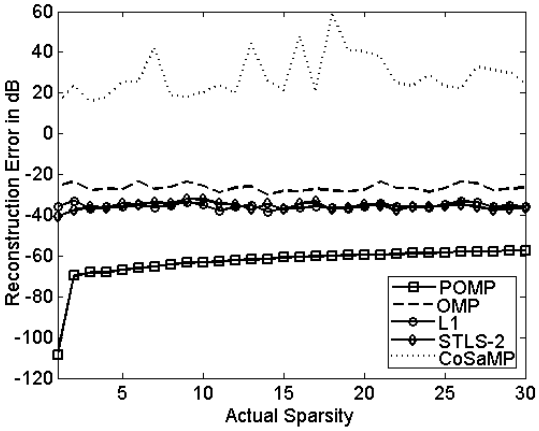


Figure 3.21: Signal reconstruction error in dB as a function of actual sparsity.

3.3.3.4 Reconstruction and Noise-to-Perturbation Ratio

In the above simulations, termination criteria of all algorithms are determined by the residual norm level ϵ. We assume that observation $\boldsymbol{y}$ has the form of $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{n} = \boldsymbol{y}_0 + \boldsymbol{n}$. If we write the residual as,

$$\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2 = \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}_0 - \boldsymbol{n}\|_2 \simeq E\{\|\boldsymbol{n}\|_2\}, \tag{3.62}$$

hence $\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2 / \|\boldsymbol{A}\boldsymbol{x}\|_2 \simeq 1/SNR$ if noise variance is small. Therefore, $\epsilon = 1/SNR$ can be chosen. More specifically, if the noise has a variance σ^2, then $\epsilon = \sigma\sqrt{M}/\|\boldsymbol{y}_0\|_2$. Since the signals are error free in previous simulations, $\epsilon = 10^{-3}$ is chosen. However, if the signal comes from a perturbed dictionary, then $\boldsymbol{y} = (\boldsymbol{A} + \delta\boldsymbol{A})\boldsymbol{x} + \boldsymbol{n}$, hence,

$$\begin{aligned} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2 &= \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{A}\boldsymbol{x} - \delta\boldsymbol{A}\boldsymbol{x} - \boldsymbol{n}\|_2 \\ &\simeq \sqrt{E\{\|\boldsymbol{n}\|_2^2\} + E\{\|\delta\boldsymbol{A}\boldsymbol{x}\|_2^2\}}, \end{aligned} \tag{3.63}$$

where noise and perturbations are assumed to be independent. It is appropriate to define

$$SPR = \|\boldsymbol{A}\,\boldsymbol{x}\|_2 / \|\delta\boldsymbol{A}\,\boldsymbol{x}\|_2, \tag{3.64}$$

as the Signal-to-Perturbation Ratio (SPR). Therefore according to Eq. 3.63, $\|\boldsymbol{Ax}-\boldsymbol{y}\|_2/\|\boldsymbol{y}\|_2 \simeq \sqrt{1/SNR+1/SPR}$. In our framework, $\|\boldsymbol{a}_{i,p}^T\,\boldsymbol{a}_i\|_2 = \cos(\phi_i)$, since $\|\boldsymbol{a}_{i,p}\|_2 = \|\boldsymbol{a}_i\|_2 = 1$. Using simple geometry, the perturbation amount can be written as $\|\boldsymbol{a}_{i,p}-\boldsymbol{a}_i\|_2 = 2\sin(\phi_i/2)$. Hence, the perturbation amount is $E\{\|\delta\boldsymbol{Ax}\|_2\} \simeq \|\boldsymbol{y}\|_2 \times E\{\phi\}$, where $E\{\phi\} = 2\sin(60^\circ\,\sqrt{3}\,E\{f\}\,\delta T/2)$. Therefore, $SPR_{dB} = 20\log\big(\|\boldsymbol{y}\|_2/(E\{\phi\}\,\|\boldsymbol{y}\|_2)\big) = 20\log(1/E\{\phi\})$. If the perturbations are small, it can be approximated as:

$$SPR_{dB} = -20\log\big(\pi\,E\{f\}/\sqrt{3}\big) - 20\log\big(\delta T\big). \tag{3.65}$$

In the following simulations, we aim to analyze the SNR and SPR regions in which POMP and compared techniques can successfully work. For this purpose sparsity is fixed at $K = 10$ and average reconstruction performance is found for varying SNR and SPR levels.

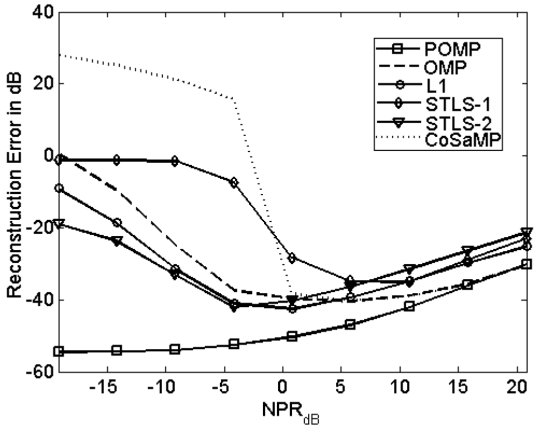


Figure 3.22: Signal reconstruction error in dB for fixed $SPR_{dB} = 40$dB and varying SNR from 20dB to 60dB.

In Figure 3.22, the perturbation level is kept constant at $SPR_{dB} = 40$dB, which corresponds to $\delta T = 25\,\mu$sec according to Eq. 3.65, and the SNR is varied from 20dB to 60dB. In Figure 3.23, we keep the noise level constant at $SNR_{dB} = 50$dB and the SPR_{dB} is varied from 30dB to 70dB. In both figures, the reconstruction performances are shown as a function of the Noise-to-Perturbation Ratio (NPR) defined as $NPR_{dB} = SPR_{dB} - SNR_{dB}$. It is clear that two distinct performance regions are determined by the sign of NPR_{dB}.

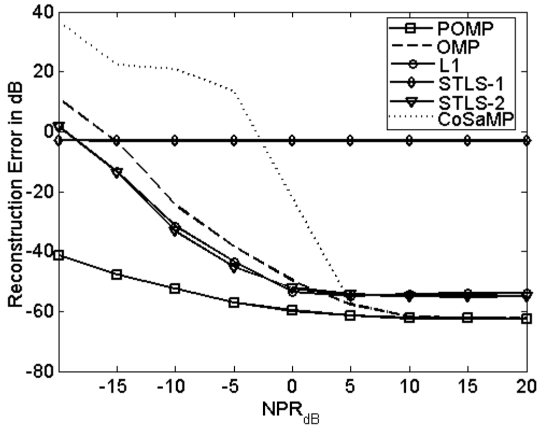


Figure 3.23: Signal reconstruction error in dB for fixed SNR_{dB} = 50dB and varying SPR from 30dB to 70dB.

As shown in Figure 3.22, although POMP has the lowest reconstruction error, all compared techniques have acceptable performances for positive NPR_{dB}. In that regime, noise power dominates the perturbation on the dictionary. Since the perturbations are relatively insignificant, OMP and CoSaMP can achieve similar results to POMP. However, when NPR_{dB} is negative, the perturbation on the dictionary is more dominant than the noise level. OMP, BP and CoSaMP have no mechanism to handle these perturbations, hence they treat the perturbations as additional noise. However, in this case it is highly improbable to decrease the residual error to the actual noise level. Thus, they saturate and produce reconstructions with larger errors. On the other hand, even if perturbations dominate the noise, POMP can handle perturbations and produces reconstruction with significantly lower errors. Figure 3.23 displays similar results, where SNR is fixed and SPR is varied. For high NPR regime, all algorithms result in similar errors, whereas for low NPR, only POMP produces acceptable reconstructions. Higher perturbation causes larger reconstruction errors in POMP. However, it is much more robust to dictionary mismatch than the compared techniques.

Chapter 4

Recursive Compressive Sensing Framework

The classical use of CS techniques starts with a set of N column vectors, that are generally constructed with computation of the signal source at some discrete parameter as in (3.4). Thus, length of the sparse signal $\boldsymbol{x}$ is fixed to N. Therefore, number of samples is considered as a free parameter. The proven reconstruction criteria of sparse solvers are between the number of measurements and the sparsity level: sparser signals require fewer number of measurements.

Unlike the classical approach, in practical problems analysis starts with a given set of observations $\boldsymbol{y}$, hence M is fixed. As discussed in Section 3.1, discretization is a required step to convert the problem into CS formulation. Since the resolution of the discretization determines the length of the sparse signal, N becomes a design parameter. Since less sparse signals require lower correlation in the basis [41, 43], the minimum resolution limit is dictated by M and sparsity level K. Therefore, for less sparse signals there will be fewer components in the signal dictionary. Thus, the off-grid problem becomes more severe when the signal is less sparse.

In this chapter of the thesis, a novel recursive framework for CS applications, in which the continuous parameter space is adaptively discretized, is presented. In the presented recursive framework any CS reconstruction technique can be used. As illustrated over commonly used CS applications, the estimated parameters of the sparse signal components almost achieve the Cramer-Rao lower bound. A preliminary form of this approach has been presented in [55] and full version is in [56].

4.1 Adaptive Discretization

As discussed in Section 3.1, a sampled data vector with a sparse representation in a continuous parameter space can be written as:

$$\boldsymbol{y} = \sum_{i=1}^{K} \alpha_i \, \psi(\boldsymbol{\theta}_{T_i}; \boldsymbol{t}) + \boldsymbol{n} \quad \text{s.t.} \quad \boldsymbol{\theta}_{T_i} \in \mathcal{P}. \tag{4.1}$$

This form can be reduced to commonly used compressive sensing setup by discretizing the continuous and bounded parameter space, $\mathcal{P}$. For a more abstract formulation that will be useful in the presentation of the proposed approach, let $d(\cdot, \cdot)$ be a functional that takes the continuous space $\mathcal{P}$ and a discretization interval λ_θ and returns a set of discrete parameter points:

$$\{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \dots, \boldsymbol{\theta}_N\} = d(\mathcal{P}, \lambda_\theta). \tag{4.2}$$

This discretization provides N grid points, $\boldsymbol{\theta}_i \in \mathcal{P}$, for $1 \leq i \leq N$. For each $\boldsymbol{\theta}_i$, the M-dimensional corresponding signal atom is computed using the given sampling times as:

$$\boldsymbol{a}_i = \psi(\boldsymbol{\theta}_i; \boldsymbol{t}). \tag{4.3}$$

By using (4.3) for each $\boldsymbol{\theta}_i$, $M \times N$ dimensional sensing matrix is constructed as $\boldsymbol{A} = [\boldsymbol{a}_1, \boldsymbol{a}_2, \dots, \boldsymbol{a}_N]$. Note that the dictionary composed with the discretization of the parameter space has to guarantee the reconstruction of an arbitrary K-sparse signal in the column space of $\boldsymbol{A}$. Each CS technique has its own guarantees for the recovery. For simplicity in the development, we will assume that OMP is used as the sparse solver. For an alternative CS reconstruction technique, it is straightforward to make proper changes in the following development.

Recovery guarantees of OMP has been discussed in the literature. In [43], a sufficient condition for the recovery of a K-sparse signal in terms of mutual coherence is provided as:

$$\mu(\boldsymbol{A}) \leq \frac{1}{2K-1}, \tag{4.4}$$

where $\mu(\boldsymbol{A})$ is defined in (2.21). Notice that mutual coherence is a functional of the basis vectors defined in (4.3), hence it has to be computed accordingly. To illustrate the effect of sufficient recovery condition on the discretization of the continuous parameter space, we will concentrate on the frequency estimation problem, hence complex exponentials with the following structure will be used as the basis functions:

$$\psi(\theta; t) = e^{-j2\pi f t}. \tag{4.5}$$

Let the discretization interval between two adjacent discrete parameter points be λ_f, hence $f_{l+k} - f_l = k\,\lambda_f$. In this case, normalized inner product of basis vectors corresponding two grid points can be computed as:

$$\begin{aligned} \frac{|\boldsymbol{a}_k^H \boldsymbol{a}_l|}{\|\boldsymbol{a}_k\|_2\,\|\boldsymbol{a}_l\|_2} &= \frac{|\,\boldsymbol{a}(f_k;\boldsymbol{t})^H \boldsymbol{a}(f_l;\boldsymbol{t})\,|}{\|\boldsymbol{a}(f_k;\boldsymbol{t})\|_2\,\|\boldsymbol{a}(f_l;\boldsymbol{t})\|_2} = \frac{1}{M}\left|\sum_{i=1}^{M} e^{j2\pi(k-l)\lambda_f t_i}\right| \\ &\approx \left|\,\mathrm{sinc}\left((k-l)\lambda_f\,T\right)\right|, \end{aligned} \tag{4.6}$$

which is a close approximation when the sampling instants have uniform distribution in $[0,\,T]$. Since $|\,\mathrm{sinc}(x)| \geq |\,\mathrm{sinc}(nx)|$ holds true for all non-zero real x and non-zero integer n, mutual coherence of the dictionary is found as:

$$\mu(\boldsymbol{A}) = \max_{k \neq l}\left|\mathrm{sinc}\left((k-l)\lambda_f\,T\right)\right| = |\,\mathrm{sinc}(\lambda_f\,T)\,|. \tag{4.7}$$

In order to guarantee that OMP will recover a K-sparse solution, the discretization interval λ_f has to satisfy the condition given in (4.4) resulting in the following inequality:

$$\mu(\boldsymbol{A}) = |\,\mathrm{sinc}(\lambda_f\,T)\,| \leq \frac{1}{2K-1}. \tag{4.8}$$

Assuming $0 \leq \lambda_f T \leq 1$, we can safely take the inverse of the sinc function. Thus the finest possible discretization interval is:

$$\lambda_f(K) = \frac{1}{T}\,\mathrm{sinc}^{-1}\left(\frac{1}{2K-1}\right). \tag{4.9}$$

Since $\mathrm{sinc}^{-1}(x)$ is a monotonic decreasing function for $0 \leq x \leq 1$, for larger K values $\lambda_f(K)$ becomes larger as well. This implies that for less sparse signals, we have to use a coarser discretization in the continuous parameter space resulting in more severe performance degradation due to off-grid problem.

In the two dimensional case, if the resolutions in both dimensions are denoted as λ_{f_1} and λ_{f_2}, then (4.8) can be written as:

$$\mu(\boldsymbol{A}) = \mathrm{sinc}(\lambda_{f_1} T)\ \mathrm{sinc}(\lambda_{f_2} T) \leq \frac{1}{2K-1}. \tag{4.10}$$

In the more general sense, if the basis functions are p-dimensional and consist of set of separable parameters $\theta_i, \ldots, \theta_p$ in the form as:

$$a(\boldsymbol{\theta};\ [t_{k_1}\ t_{k_2}\ \ldots\ t_{k_p}]) = \prod_{i=1}^{p} a_i(\theta_i;\ t_{k_i}), \tag{4.11}$$

and the correlation depends only on the difference between the corresponding parameters, i.e. $a_i^*(\theta_i^1;\ t_{k_i})\, a_i(\theta_i^2;\ t_{k_i}) = f_i(\theta_i^1 - \theta_i^2;\ t_{k_i})$, then the mutual coherence of the basis can expressed as:

$$\mu(\boldsymbol{A}) = \prod_{i=1}^{p} \mu_i(\lambda_{\theta_i}), \tag{4.12}$$

where $\mu_i(\cdot)$ is the function that relates the resolution of the i^{th} parameter to the corresponding correlation. If there are more than one parameters in the signal atoms, the required resolution in the corresponding parameters that satisfies the reconstruction with OMP is not unique. In 2D case, set of all allowed resolutions for a given sparsity level lies on a curve. Fig. 4.1 shows this resolution curves for several sparsity levels. In general, if the basis functions can be represented as in

(4.11), set of allowed minimum discretization intervals form a $p-1$ dimensional manifold.

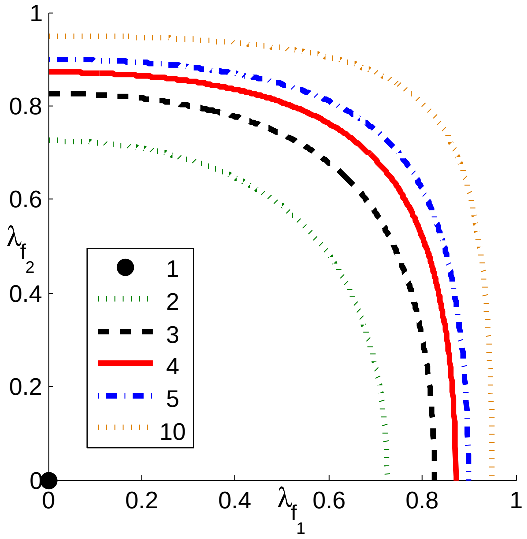


Figure 4.1: Necessary discretization interval curves that guarantees the recovery of the sparse signal in the two dimensional spectrum estimation problem.

Typically, in the discretization of 2D continuous Fourier space, equal spacing is used in both dimensions. Then, corresponding mutual correlation becomes $\mu(\boldsymbol{A}) = \text{sinc}^2(\lambda_f T)$. Fig. 4.2 shows the feasible set of discretization intervals for 1D and 2D Fourier spaces.

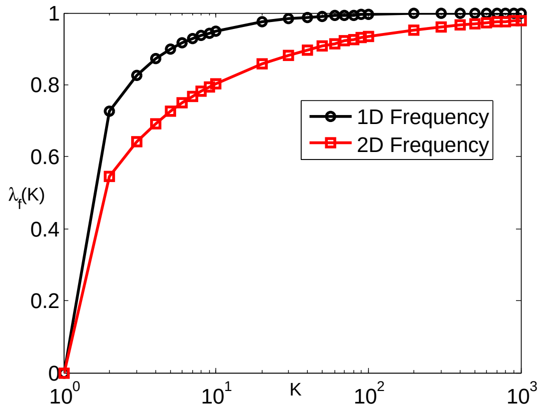


Figure 4.2: Lower bound of feasible discretization interval with respect to sparsity level that guarantees the recovery of the sparse signal for 1D and 2D Fourier spaces.

The interval defined in (4.9) is the finest discretization that guarantees the recovery of K-sparse signal with OMP as the solver. Therefore, the continuous model in (4.1) has to be discretized accordingly. As expected, the discretization interval is an increasing function of K. Also, we observe that the discretization interval is always less than $1/T$. It is known that if a system with classical sampling procedure takes samples in the $[0,\ T]$ time range, its frequency resolution is $1/T$. Having a resolution smaller than $1/T$ in the least sparse case is consistent with this fact. Finally, we observe that the allowed interval for 1-sparse case is 0. In other words, for 1-sparse signals discretization interval can be set arbitrarily small, yet the recovery is still guaranteed. This last fact will provide the foundation of the proposed recursive approach presented in the following section.

It is important to note that (4.9) provides the allowed minimum interval for the frequency estimation problem when OMP is used as the sparse solver. The allowed interval for another solver or basis function may differ. In a more general and abstract sense, the required resolution can be represented as follows:

$$\lambda_\theta = f(M, K). \tag{4.13}$$

Here the function $f(\cdot,\cdot)$ can be computed analytically or numerically depending on the recovery condition imposed by the solver of choice on the basis functions. However, the observation based on the OMP case are valid for some other solvers.

4.2 Proposed Recursive Compressive Sensing Framework

For the model given in (4.1), for a given measurement vector $\boldsymbol{y} \in \mathcal{C}^M$ and a provisional estimate of the sparsity level K, compressive sensing solver can be written in the following abstract form:

$$[\boldsymbol{\alpha}^*, \boldsymbol{\theta}^*] = \mathcal{S}(\boldsymbol{y}, K, \mathcal{P}), \tag{4.14}$$

where $\boldsymbol{\theta}^*$ is the estimated parameter values, $\boldsymbol{\alpha}^*$ are the corresponding representation coefficients and $\mathcal{P}$ is the bounded and continuous parameter space. A sensing matrix is required in order to utilize CS reconstruction techniques in the solver in (4.14). At this step first thing is to determine the finest resolution dictated by (4.13). Then the parameter space $\mathcal{P}$ is discretized accordingly and the sensing matrix $\boldsymbol{A}$ is constructed using (4.3). In this way, we can define the problem as $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{n}$ and solve for the K-sparse reconstruction of the signal for a selected sparse solver of choice yielding $\boldsymbol{\alpha}^*$ and $\boldsymbol{\theta}^*$. Using $[\boldsymbol{\alpha}^*, \boldsymbol{\theta}^*]$, the observation vector can be represented as:

$$\boldsymbol{y} = \sum_{i=1}^{K} \alpha_i^* \, \psi(\theta_i^*; \boldsymbol{t}) + \boldsymbol{n}, \tag{4.15}$$

where $\boldsymbol{n}$ corresponds to the fit error of the sparse solver. The important thing to notice is that the matrix-vector model is an approximate relationship due to discretization of the parameter space. However, if the problem is highly sparse, this approximate relationship has a relatively high accuracy since the allowed discretization interval in (4.13) is a decreasing function of the sparsity. Therefore our main purpose is to split the K-sparse problem into set of smaller problems with higher sparsity. For this purpose, assume that we divide the problem into c partitions as follows:

$$\boldsymbol{y} = \sum_{i=1}^{K_1} \alpha_{1,i}^* \, a(\theta_{1,i}^*; \boldsymbol{t}) + \ldots + \sum_{i=1}^{K_c} \alpha_{p,i}^* \, a(\theta_{p,i}^*; \boldsymbol{t}) + \boldsymbol{n}, \tag{4.16}$$

where $\sum_{j=1}^{c} K_j = K$, $\bigcup_{j,i} \alpha_{j,i}^* = \boldsymbol{\alpha}^*$ and $\bigcup_{j,i} \theta_{j,i}^* = \boldsymbol{\theta}^*$. This process also partitions the given parameter space $\mathcal{P}$ into disjoint sets such that $\mathcal{P} = \bigcup_j \mathcal{P}_j$ and $\mathcal{P}_j \cap \mathcal{P}_k = \emptyset$ if $j \neq k$ with $\theta_{j,i}^* \in \mathcal{P}_j$. E-M framework provides an effective solution for the partitioned problem [57, 58]. Assuming the estimates of the last $c-1$ partitions' parameters are precise enough, we can construct the observation vector corresponding to those partitions and then find the partial observation vector corresponding to the first partition. This is the E-step of the framework. The M-step is to solve the problem with partial observation vector in its corresponding domain.

In general, in the E-step for an arbitrary l^{th} partition, the corresponding partial observation vector is written as:

$$\boldsymbol{y}_l = \boldsymbol{y} - \sum_{\substack{j=1 \\ j \neq l}}^{c} \sum_{i=1}^{K_j} \alpha^*_{j,i}\, a(\theta^*_{j,i}; \boldsymbol{t}). \tag{4.17}$$

In the M-step, in order to estimate parameters with higher accuracy, we solve the sparse reconstruction problem of observation vector $\boldsymbol{y}_l$ with sparsity estimation K_l in the domain $\mathcal{P}_l$. Therefore, M-step of the framework is written as the solution of the following problem,

$$[\boldsymbol{\alpha}^*_l, \boldsymbol{\theta}^*_l] = \boldsymbol{\mathcal{S}}(\boldsymbol{y}_l, K_l, \mathcal{P}_l). \tag{4.18}$$

Iteratively solving (4.17) and (4.18) from $l = 1$ to $l = c$ realizes one pass of the EM approach. The most important thing to notice is the transformation of (4.14) to (4.18) under the EM framework. The problems are identical to each other in the structural sense: both take an observation vector, provisional sparsity level and a parameter space to operate on. Thus, the approach used in the solution of (4.14) can also be applied to (4.18). Successively applying the very same approach to each sub-problem, the main problem splits itself into smaller, and sparser, sub-problems with denser discretization of the parameter space in a recursive manner. The remarkable advantage of this approach lies in the reduction of the sparsity levels in the sub-problems. In the fragmentation from (4.14) to (4.18), the immediate observation is that $K_l < K$ for all $1 \leq l \leq c$ provided that $c > 1$. Due to decreasing characteristics of (4.13), finer discretization of corresponding parameter spaces can be performed in the sub-problems, which improves the accuracy of the parameter estimations.

The proposed EM based recursive solution approach is summarized in Table 4.1. Some of the steps are not defined precisely so that the framework can be utilized in a wide range of problems with different characteristics specific to implementation of those steps. In the following sub-sections, we will elaborate on the individual steps for clarity.

Table 4.1: EM Based Recursive Algorithm

```
Signature: $[\widehat{\boldsymbol{\alpha}},\ \widehat{\boldsymbol{\theta}}] = \mathcal{S}(\boldsymbol{y}, K, \mathcal{P})$

1) $\lambda_\theta = f(M, K)$, find the required resolution,
2) $\{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N\} = d(\mathcal{P}, \lambda_\theta)$, discretization,
3) $\boldsymbol{A} = [\psi(\boldsymbol{\theta}_1; \boldsymbol{t}) \ \ldots \ \psi(\boldsymbol{\theta}_N; \boldsymbol{t})]$, basis matrix,
4) $[\boldsymbol{\alpha}, \boldsymbol{\theta}] = SparseSolver(\boldsymbol{y}, \boldsymbol{A}, K)$, a solution,
5) While $[\boldsymbol{\alpha}, \boldsymbol{\theta}]$ is not a satisfactory solution,
  5.1) $\Big\{\big(\boldsymbol{\alpha}_1, \boldsymbol{\theta}_1, \mathcal{P}_1\big), \ldots, \big(\boldsymbol{\alpha}_c, \boldsymbol{\theta}_c, \mathcal{P}_c\big)\Big\} = Partition(\boldsymbol{\alpha}, \boldsymbol{\theta}, \mathcal{P})$,
  5.2) For each partition $l$ from 1 to $c$
    5.2.1) $\boldsymbol{y}_l = \boldsymbol{y} - \sum_{\substack{j=1 \\ j \neq l}}^{c} \sum_{i=1}^{K_j} \alpha_{j,i}\, \psi(\boldsymbol{\theta}_{j,i}; \boldsymbol{t})$
    5.2.2) $[\boldsymbol{\alpha}_l,\ \boldsymbol{\theta}_l] = \mathcal{S}(\boldsymbol{y}_l, K_l, \mathcal{P}_l)$
  5.3) $\boldsymbol{\alpha} = \bigcup_{j,i} \alpha_{j,i},\ \boldsymbol{\theta} = \bigcup_{j,i} \boldsymbol{\theta}_{j,i}$, combine,
6) $\widehat{\boldsymbol{\alpha}} = \boldsymbol{\alpha}$ and $\widehat{\boldsymbol{\theta}} = \boldsymbol{\theta}$, finalize the solution.
```

4.2.1 Base Case of the Recursion

Separation of the problem will be terminated in a finite amount of recursive calls since the sparsity of the main problem K is finite. The most interesting case, which is also the base case of the recursion, happens when the sparsity of that partition reduces to one. Amenity of this case is the result of the following fact $\lambda_\theta(1) = 0$. When an instance of (4.14) is initialized with $K = 1$, the solver is allowed to discretize the parameter space arbitrarily dense. However in the noisy setting, partition size is lower-bounded by the Cramér-Rao lower bound (CRLB) which is the universal lower bound for the non-random parameter estimation. This case will be illustrated on the frequency estimation problem using basis functions given in (4.5). In this model, observations are in the following structure in the 1-sparse case:

$$\boldsymbol{y} = \alpha \; e^{j\phi} \; e^{j\omega \boldsymbol{t}} + \boldsymbol{n}, \tag{4.19}$$

where α, ϕ and ω are the unknown amplitude, phase and angular frequency, respectively; $\boldsymbol{n}$ is a zero mean circularly symmetric i.i.d. complex Gaussian noise with variance σ^2, and $\boldsymbol{t}$ is the vector of sampling times.

In the frequency estimation problem, 1-sparse case is similar to the single tone frequency estimation problem. In the regular sampling, this is a well studied problem [59]. However, to best of our knowledge there is no reported study on the CRLB of the single tone frequency estimation under random sampling.

In (4.19), important difference from the regular sampling is that time sampling is also random. Therefore, expectation of the Fisher information matrix over the distribution of the sampling times should be obtained. As derived in Appendix A.3, the CRLB for the single tone frequency estimation under the random sampling are:

$$\begin{aligned} var(\widehat{\alpha}) &\geq \frac{\sigma^2}{M}, \\ var(\widehat{\phi}) &\geq \frac{(\sigma/\alpha)^2}{M} \frac{E[t_i^2]}{var(t_i)}, \\ var(\widehat{\omega}) &\geq \frac{(\sigma/\alpha)^2}{M} \frac{1}{var(t_i)}, \end{aligned} \tag{4.20}$$

where $var(t_i)$ is the variance and $E[t_i^2]$ is the second order moment of the random time samples. When sampling times are generated from an i.i.d. uniform distribution in the $[0, T]$ range, CRLB of the frequency estimation becomes:

$$var(\widehat{\omega}) \geq J_u = \frac{(\sigma/\alpha)^2}{M} \frac{12}{T^2}, \tag{4.21}$$

which is asymptotically equivalent to the CRLB in the regular sampling case [59]. The difference of the random sampling is the dependency of CRLB on the variance of the sampling distribution. Even though uniform distribution results in asymptotically equivalent CRLB of the regular sampling, it is possible to choose a distribution with larger variance to reduce the corresponding CRLB below the CRLB corresponding to the uniform sampling. For instance, Bernoulli random variable having values at 0 and T with equal probabilities has a variance of $T^2/4$, hence achieve one-third of the CRLB of the regular sampling!

If we use an unbiased estimator of the parameters, square-root of the CRLB can be thought as the finest partition size that the estimator can achieve under the assumed noise statistics. Even though (4.13) allows arbitrarily dense discretization in the 1-sparse case, resolution smaller than square-root of the CRLB will not provide any further improvement in the estimation performance. Therefore, treatment to this important case is to re-define (4.9) for $K = 1$ as follows:

$$\lambda_f(1) := \gamma \frac{\sqrt{J_u}}{2\pi} = \gamma \, \frac{\sqrt{3}}{\pi} \, \frac{1}{T\sqrt{M}} \, \frac{\sigma}{\alpha}, \tag{4.22}$$

where 2π is result of the conversion to ordinary from angular frequency and γ is a fraction to absorb the discretization effect in the estimation.

It has been shown for the regular sampling case that spectrum estimation is the ML estimate of the frequency, [59], which is defined as:

$$\widehat{f} = \arg\max_{f} \left| \boldsymbol{y}^H \, \boldsymbol{a}(f; \boldsymbol{t}) \right|. \tag{4.23}$$

The significance of the spectrum estimation is its equivalence to the search step of the OMP algorithm. In the 1-sparse case, if the given parameter space is discretized according to analyzed CRLB limit in (4.22) and OMP is used for the sparse solver, resulting solution will be the ML estimate of the frequency.

4.2.2 Sparse Solver

The proposed framework can be used with any sparse solver of choice. However, algorithms with low complexity is preferable since the proposed framework recursively initiates several instances of the same problem. More importantly, selected sparse solver is expected to be a minimum variance unbiased estimator (MVUE) of the parameters for a sparsity level so that the proposed framework achieves the Cramér-Rao lower bound in the estimation variance.

4.2.3 Satisfactory Solution

In order to be able to complete the recursive calls in an appropriate amount of time, a termination criterion should be defined. One straightforward way is to compute the residual error. Given the provisional solution set $(\boldsymbol{\alpha}, \boldsymbol{\theta})$, the corresponding estimate for the observation vector is computed as:

$$\boldsymbol{y}^* = \sum_{i=1}^{K} \alpha_i \, \psi(\boldsymbol{\theta}_i; \boldsymbol{t}). \tag{4.24}$$

Using a predefined threshold ϵ_1, the provisional solution can be qualified as satisfactory if $\|\boldsymbol{y} - \boldsymbol{y}^*\|_2 \leq \epsilon_1$. Another approach is to monitor the residual norm and terminate the iterations when rate of decrease in the residual is below a certain threshold ϵ_2, $\|\boldsymbol{y}^*_{q-1}\|_2 / \|\boldsymbol{y}^*_q\|_2 \leq \epsilon_2$, where q is the index of iterations. Apart from the residual, iterations may be terminated when the changes in the solution set is lower than a certain threshold. Also, total number of iterations can be bounded. For a robust behavior, some of the discussed metrics can be used in conjunction.

4.2.4 Partitioning

Splitting the problem into self-similar sub-problems requires a partitioning operation on the provisional solution set. Partitioning can be considered as the well-studied problem of clustering [60] in the machine learning theory. In the proposed framework a specific clustering algorithm can be implemented to partition the original problem into sub-problems. However, to reduce the required computational load, we propose to use simple partitioning structures that will be elaborated in the following section.

4.3 Recursion on a Binary Tree

Clustering is an important problem in the proposed framework since it is the initial phase of the EM iterations and will be successively called in each iteration and recursive calls. By selecting fixed $c = 2$, we split the main problem into two distinct parts in each recursive call. Hence, the sub-problems and their relations with the main problem can be represented with a binary tree. However, structure of the tree will depend on partitioning technique. In the following, two different approaches with their implementation and complexity analysis are provided.

4.3.1 Partitioning on an Unbalanced Tree

4.3.1.1 Structure

In the unbalanced tree approach, the provisional solution is partitioned as "*the most dominant component*" and "*the rest*". Thus in each recursive call, problem will be split into two sub-problems with sparsity 1 and $K - 1$. This partitioning approach will generate a solution tree similar to the one in Fig. 4.3(a).

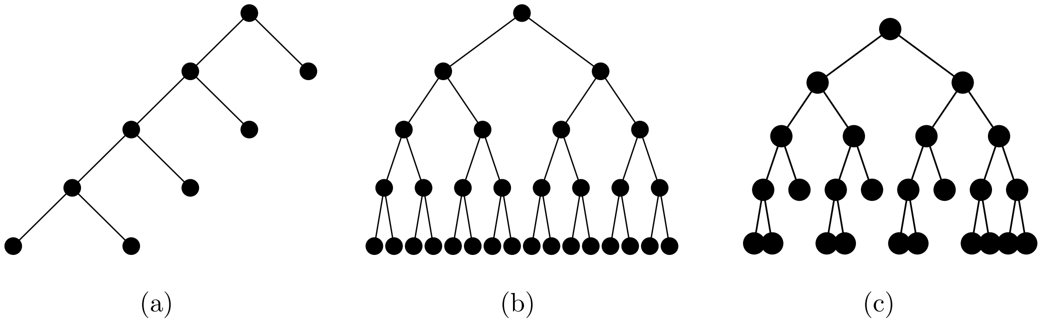

(a) (b) (c)

Figure 4.3: A sample solution path in the (a) unbalanced binary tree with $K = 5$, (b) balanced and symmetric binary tree with $K = 2^4$, (c) balanced binary tree with $K = 13$.

Without loss of generality, under proper permutation of the indices, for (4.15) we can assume $|\alpha_1^*| \geq |\alpha_2^*| \geq \ldots \geq |\alpha_K^*|$, hence the parameter space corresponding to the dominant component can be written as:

$$\mathcal{P}_1 = \mathcal{B}_r(\boldsymbol{\theta}_1^*) \subset \mathcal{P}, \tag{4.25}$$

where $\mathcal{B}_r(\theta_1^*)$ corresponds to a ball center θ_1^* and radius r. If $r < |\lambda_\theta(K)|$, it is only necessary to find a 1-sparse solution in the domain $\mathcal{P}_1$. Therefore, the observation vector corresponding to this partition is written as:

$$\boldsymbol{y}_1 = \boldsymbol{y} - \sum_{i=2}^{K} \alpha_i^* \, \psi(\boldsymbol{\theta}_i^*; \boldsymbol{t}). \tag{4.26}$$

With the construction of the parameter space and the observation vector, an improved estimate for the dominant parameter is obtained as the solution of the following problem with the same form:

$$[\alpha_1^*, \boldsymbol{\theta}_1^*] = \mathcal{S}(\boldsymbol{y}_1, 1, \mathcal{P}_1). \tag{4.27}$$

When an estimate for the dominant component is obtained, the parameter space and the corresponding observation vector for the remaining components are constructed as:

$$\begin{aligned} \mathcal{P}_1' &= \mathcal{P} \setminus \mathcal{P}_1, \\ \boldsymbol{y}_1' &= \boldsymbol{y} - \alpha_1^* \, \psi(\boldsymbol{\theta}_1^*; \boldsymbol{t}). \end{aligned} \tag{4.28}$$

Then, a refinement for the remaining parameters is obtained as the solution of the following problem:

$$[\alpha_2^*, \ldots, \alpha_K^*, \boldsymbol{\theta}_2^*, \ldots, \boldsymbol{\theta}_K^*] = \mathcal{S}(\boldsymbol{y}_1', K-1, \mathcal{P}_1'). \tag{4.29}$$

When the radius of the neighborhood defined in (4.25) is selected as $r \geq |\lambda_\theta(K)|$, components with parameters other than $\boldsymbol{\theta}_1^*$ may also fall into the neighborhood. If we assume that K_0 of the components are in $\mathcal{B}_r(\boldsymbol{\theta}_1^*)$, then the main problem is split into sub-problems with sparsity K_0 and $K - K_0$, which is also expected to generate an unbalanced solution tree.

4.3.1.2 Complexity

When a problem in the form of (4.14) is initiated, the proposed algorithm first solves a K-sparse problem. Then, K parameters are partitioned and 1-sparse and $K-1$-sparse problems are solved. Partitioning and re-solving is repeated during the E-M iterations. As a result, in the solution of (4.14) total computational cost can be written as:

$$C(K) = S(K) + \Big[P(K) + C(1) + C(K-1)\Big]\,\gamma(K), \tag{4.30}$$

where $C(K), S(K), P(K)$ are the computational costs of the solution of K-sparse problem in the proposed framework, solution of K-sparse problem with the sparse solver and partitioning of K parameters, respectively. Since partitioning and solving sub-problems are repeated, $\gamma(K)$ denotes the required number of E-M iterations. When the recursive relation in (4.30) is expanded, cost of K-sparse problem can be found as:

$$C(K) = \sum_{i=0}^{K-2} \Big[\frac{S(K-i)}{\gamma(K-i)} + P(K-i) + C(1)\Big] \prod_{j=0}^{i} \gamma(K-j) + C(1) \prod_{i=0}^{K-2} \gamma(K-i), \tag{4.31}$$

which can be simplified as:

$$C(K) \;\propto\; K \prod_{i=2}^{K} \gamma(i). \tag{4.32}$$

When the E-M iterations are assumed to be constant for all sub-problems, $\gamma(i) = \gamma$, overall complexity is found as $C(K) \;\propto\; K\,\gamma^K$, which is exponential in K.

4.3.1.3 Demonstration of a Solution

In this section, solution of a problem will be demonstrated for a better understanding of the steps of the proposed structure. Assume that we are given an observation vector which has a 4-sparse representation in the 2-dimensional frequency domain. Therefore, the intervals provided in Fig. 4.2 will be used in the discretization. We will assume that the parameters lie in the domain $\mathcal{P} = [100\ 110] \times [100\ 110]$.

In the first step, the proposed framework discretizes the basis with $\lambda_f = 0.693$ due to value of $\text{sinc}^{-1}(\sqrt{1/7})$. Then, OMP algorithm provides a set of estimates using the discretized domain. The first line of Fig. 4.4 shows the actual parameters and the ones estimated by OMP. Due to off-grid problem, inaccuracy in the estimation causes large residual error. Since the solution is not considered as satisfactory, algorithm proceeds to E-M iterations in step 5 of Table 4.1.

In step 5.1 of the framework, partitioning algorithm splits the problem. Since unbalanced binary tree is considered, the component with the highest energy is separated as the first partition. Observation vector is computed with (4.26). In the example given in Fig. 4.4, the most dominant component lies in the domain $\mathcal{P}_1 = [100.4\ 101.2] \times [100.4\ 101.2]$, hence it is considered as the domain of the 1-sparse problem. Since $K = 1$ is the base case of the recursion, given domain is discretized by the CRLB limit dictated by the assumed noise statistics. Notice that, the CRLB limit here is different than (4.22) since the problem here is 2 dimensional frequency estimation. This procedure on the dominant parameter increases the accuracy of the estimation as given on the left of the second line in Fig. 4.4.

After the termination of the 1-sparse sub-problem, main problem starts iteration in the other partition. Notice that in step 5.2.1, the algorithm uses the recently updated parameters of the first part. It is important to notice that 1-sparse and 3-sparse sub-problems do not run in parallel, each sub-problem is initialized successively one after another.

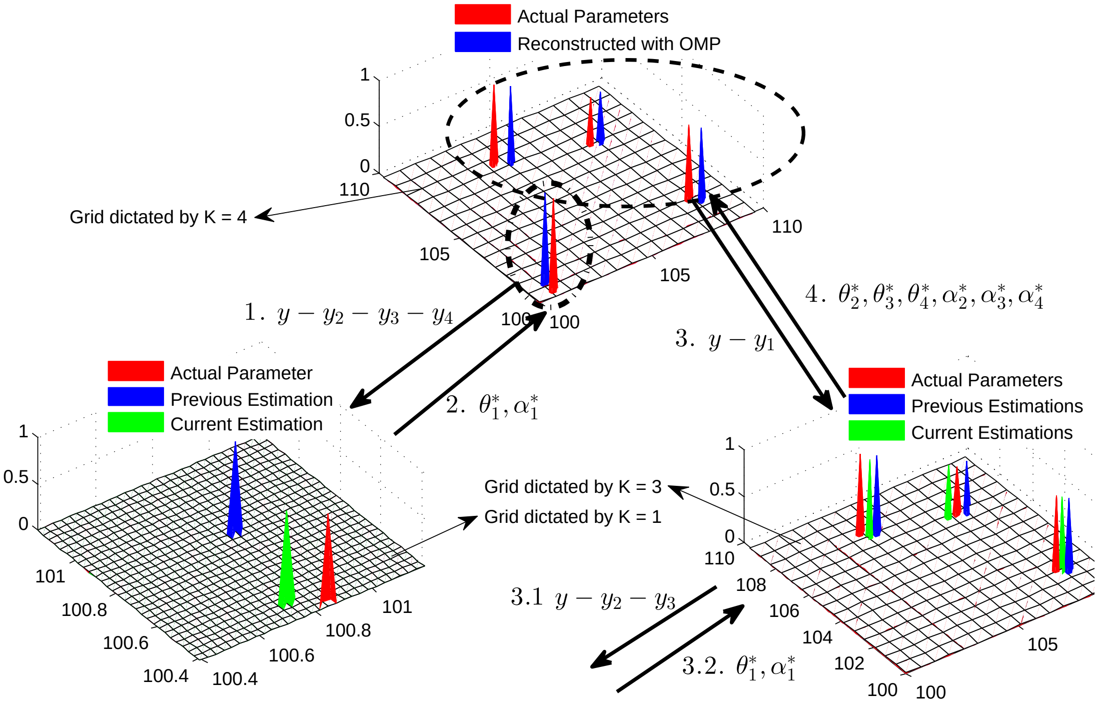


Figure 4.4: Steps of the solution to a sample 4-sparse problem with unbalanced tree partitioning.

In step 5.2.2 another instance of the same algorithm is initialized with $K = 3$, hence $\lambda_f = \text{sinc}^{-1}(\sqrt{1/5}) = 0.642$ is used for the discretization. Domain of the problem is defined as the remaining from the first part $\mathcal{P}_1' = \mathcal{P} \setminus \mathcal{P}_1$ and the observation vector is computed according to (4.28). As the main problem, this sub-problem partitions itself in a recursive manner.

When the 3-sparse sub-problem is terminated, the main problem proceeds to step 5.3 and merges the solution produced by the 1-sparse and 3-sparse sub-problems. Compared to the initial solution, one pass of the algorithm results in increased accuracy in the solution parameter. The natural progression is to repeat this procedure for more improved estimates. With this purpose, the main problem returns to step 5 to check necessity of another pass. If the current set of estimates results in a residual lower than a certain threshold, algorithm terminates; if not, it follows the same steps until one of the termination criteria is met.

4.3.2 Partitioning on a Balanced Tree

4.3.2.1 Structure

In the balanced tree approach, the provisional solution will be split into two parts with equal number of non-zero elements. Recursively repeating this approach, the main problem will be split into all the way down to 1-sparse problems. This partitioning approach will generate a solution tree similar to the one in Fig. 4.3(b) assuming that sparsity level $K = 2^n$ for some integer n. If the sparsity level is not a power of 2, problem will be split into two problems with approximately equal sparsities. Even though the resulting solution tree will not be symmetric, it will be similar to one in Fig. 4.3(c).

Required partitioning of the domain can be obtained as the solution of the following optimization problem:

$$\min_{\mathcal{P}_1} \left| K/2 - \sum_{i=1}^{K} \mathbb{1}_{\mathcal{P}_1}(\theta_i^*) \right|, \tag{4.33}$$

where $\mathbb{1}_{\mathcal{P}}(\theta)$ denotes the indicator function: its value is 1 if $\theta \in \mathcal{P}$, 0 otherwise. Since $\mathcal{P}_1$ is a continuous subset of $\mathcal{P}$, there are infinitely many solutions to the problem in (4.33). It is possible to introduce constraints on the set, or on the parameters, to decrease the number of possible solutions while having partitions with the desired properties. One admissible constraint is to force the partitions to have approximately equal energies. By this way main problem is partitioned equally not only sparsity-wise but also energy-wise. It is also possible to force the domains to be connected or even convex for more reasonable partitions. With such requirements, the problem in (4.33) can be constraint as:

$$\min_{\mathcal{P}_1} \left| K/2 - \sum_{i=1}^{K} \mathbb{1}_{\mathcal{P}_1}(\theta_i^*) \right| \quad \text{s.t.} \quad \left| \sum_{i \in S_1} (\alpha_i^*)^2 - \sum_{i \notin S_1} (\alpha_i^*)^2 \right| \leq \epsilon, \quad \mathcal{P}_1 \text{ is connected}, \tag{4.34}$$

where S_1 is the set of indices of the parameters in the domain $\mathcal{P}_1$, i.e. $i \in S_1$ if $\theta_i^* \in \mathcal{P}_1$. Even though solution to (4.34) is still not unique due to continuity of

$\mathcal{P}_1$, it provides more useful partitions.

Solution to (4.34) is not trivial in general. It is a necessity to use a suitable technique depending on the basis functions and problem model. One can also define a different optimization problem that satisfies the needs of the model.

In the frequency estimation problem, (4.5), the domain $\mathcal{P}$ is an interval constituting the bandwidth of interest, i.e. $\mathcal{P} = [f_{min}\, f_{max}]$. Our purpose is to select a cut frequency f_c such that $\mathcal{P}$ is split into two, i.e. $\mathcal{P}_1 = [f_{min}\, f_c]$ and $\mathcal{P}_1' = [f_c\, f_{max}]$. We can solve for the following problem to obtain a cut frequency:

$$\min_{f_c} \left| \frac{\|\boldsymbol{\alpha}^*\|_2^2}{2} - \sum_{i \text{ s.t. } f_i^* < f_c} (\alpha_i^*)^2 \right|, \tag{4.35}$$

which splits the domain into two energy-wise approximately equal parts. When coefficients, α_i, are assumed to be distributed i.i.d., solution to (4.35) also splits the domain into two sparsity-wise approximately equal parts. Since there are only K frequencies in the provisional solution, (4.35) can be solved very efficiently with a single pass over K parameters. When f_c is obtained, $\mathcal{P}_1$ hence S_1 can be constructed. Therefore, the observation vector corresponding to first partition is written as:

$$\boldsymbol{y}_1 = \boldsymbol{y} - \sum_{i \notin S_1} \alpha_i^* \, \psi(\boldsymbol{\theta}_i^*; \boldsymbol{t}). \tag{4.36}$$

and the following problem is solved to obtain improved estimates of the parameters in domain $\mathcal{P}_1$,

$$[\alpha_1^*, \ldots, \alpha_{K_1}^*, \boldsymbol{\theta}_1^*, \ldots, \boldsymbol{\theta}_{K_1}^*] = \mathcal{S}(\boldsymbol{y}_1, K_1, \mathcal{P}_1), \tag{4.37}$$

where $K_1 = |S_1|$ is the number of frequencies in $\mathcal{P}_1$. When the solution for the first half is obtained, observation vector corresponding to second half is constructed as:

$$\boldsymbol{y}_1' = \boldsymbol{y} - \sum_{i \in S_1} \alpha_i^* \, \psi(\boldsymbol{\theta}_i^*; \boldsymbol{t}). \tag{4.38}$$

and the following problem is solved to obtain improved estimates of the parameters in domain $\mathcal{P}'_1$,

$$[\alpha^*_{K_1+1}, \ldots, \alpha^*_K, \boldsymbol{\theta}^*_{K_1+1}, \ldots, \boldsymbol{\theta}^*_K] = \mathcal{S}(\boldsymbol{y}'_1, K - K_1, \mathcal{P}'_1). \tag{4.39}$$

As in the unbalanced tree partitioning, these iterations are recursively repeated until at least one of the termination criteria is met.

4.3.2.2 Complexity

After the solution to initial K-sparse problem, balanced partitioning approach will split the parameters into two $K/2$-sparse problems and the algorithm is recursively applied on the newly obtained partitions. Assuming $K = 2^n$, total computational cost of the solution of K-sparse problem can be written similar to (4.30) as:

$$C(2^n) = S(2^n) + \Big[P(2^n) + C(2^{n-1}) + C(2^{n-1})\Big]\,\gamma(2^n). \tag{4.40}$$

When the recursive relation in (4.40) expanded, cost of K-sparse problem is found as:

$$C(2^n) = \sum_{i=0}^{n-1} \Big[2^i\,\frac{S(2^{n-i})}{\gamma(2^{n-i})} + P(2^{n-i})\Big] \prod_{j=0}^{i} \gamma(2^{n-j}) + 2^n C(1) \prod_{i=0}^{n-1} \gamma(2^{n-j}), \tag{4.41}$$

which can be simplified as:

$$C(2^n) \;\propto\; n\; 2^{n-1} \prod_{i=2}^{n} \gamma(2^i). \tag{4.42}$$

When E-M iterations are assumed to be constant for all sub-problems, $\gamma(i) = \gamma$, overall complexity is found as $C(K) \;\propto K \log(K)$. Therefore, at the expense of the high-cost partitioning in the balanced tree approach, the exponential complexity in the unbalanced case reduces to $K \log(K)$ in terms of K. This result is also validated experimentally in Section 4.4.

4.4 Simulations

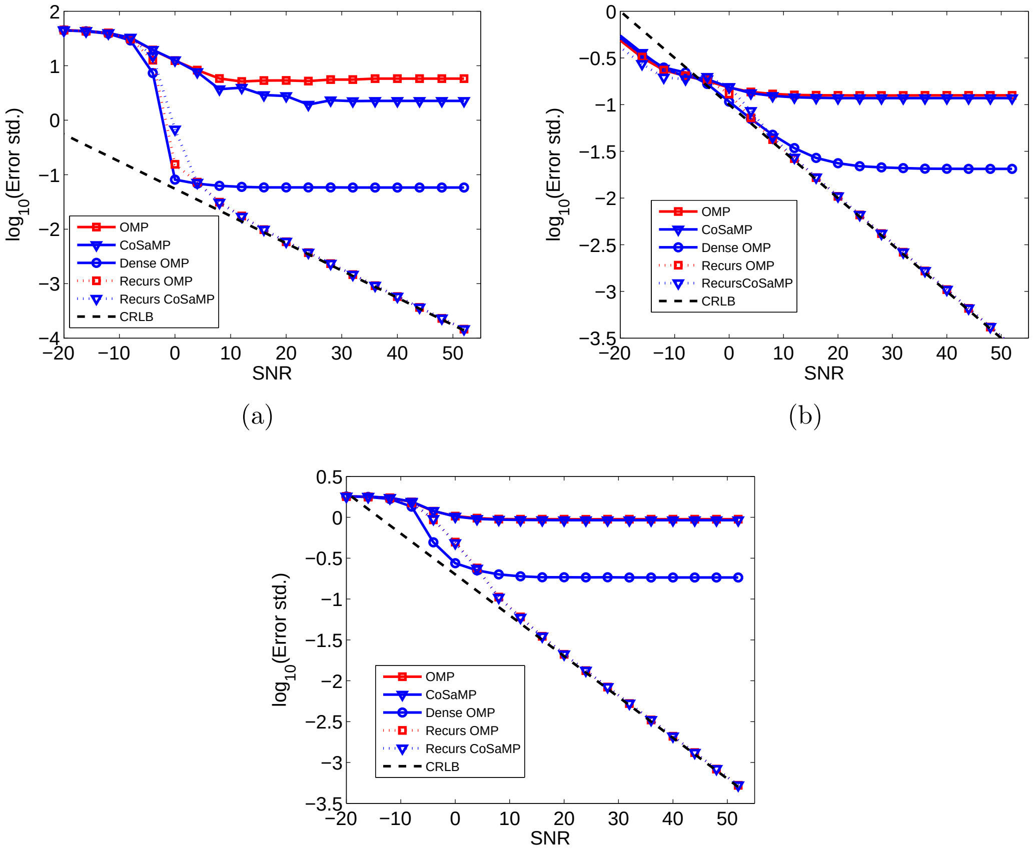
In this section, performance of the proposed framework is investigated in sparse spectral estimation. The observation vector with K-sparse components is constructed as:

$$\boldsymbol{y} = \sum_{i=1}^{K} \alpha_i \, e^{j\phi_i} \, e^{j2\pi f_i \boldsymbol{t}} + \boldsymbol{n}, \tag{4.43}$$

where $\boldsymbol{t} \in \Re^M$ is constructed by selecting time samples uniformly from $[0, T]$ range with $T = 1$ s. ϕ_i's are selected uniformly in $[0, 2\pi]$ range and $\alpha_i = 1$ for $1 \leq i \leq K$. The frequency of the components, f_i, are selected uniformly random in $[100, 300]$ Hz range; $\boldsymbol{n} \in \mathbb{C}^M$ is a circularly symmetric i.i.d. complex Gaussian noise with zero mean and standard deviation σ. In the following, α/σ will be considered as a measure of Signal-to-Noise Ratio (SNR).

To illustrate the performance gain in the proposed framework, OMP and CoSaMP are compared to their recursive counterparts in the proposed framework. Information about the true sparsity level is available to CoSaMP, and OMP is also terminated at the true sparsity level for a fair comparison. Even though OMP with a fine grid has been reported with a limited performance gain [52], the results of OMP with a dense grid is also provided for the comparison purposes. Together with the standard deviations of the error in the estimated parameters, the CRLB's given in (4.20) are presented for comparison purpose.

In the first set of simulations, the sparsity level is fixed to $K = 5$ and $M = 100$ random samples are used, which corresponds to one quarter of the Nyquist rate samples. An observation vector with randomly selected frequencies is generated according to (4.43) and noise with SNR ranging from -20 dB to 50 dB is added. Fig. 4.5 provides the standard deviation of the errors with respect to the simulated SNR levels. Regular solvers and their recursive counterparts have a transition around 0 dB, which corresponds to the case of $\alpha < \sigma$, regular and their recursive counterparts behave similarly with a significant deviation from the CRLB. When SNR is higher than 0 dB, there is a little improvement in the regular solvers. Due to the off-grid problem, solvers do not provide significant improvements even at high SNR. For OMP, a denser grid brings a reduction in



(a)

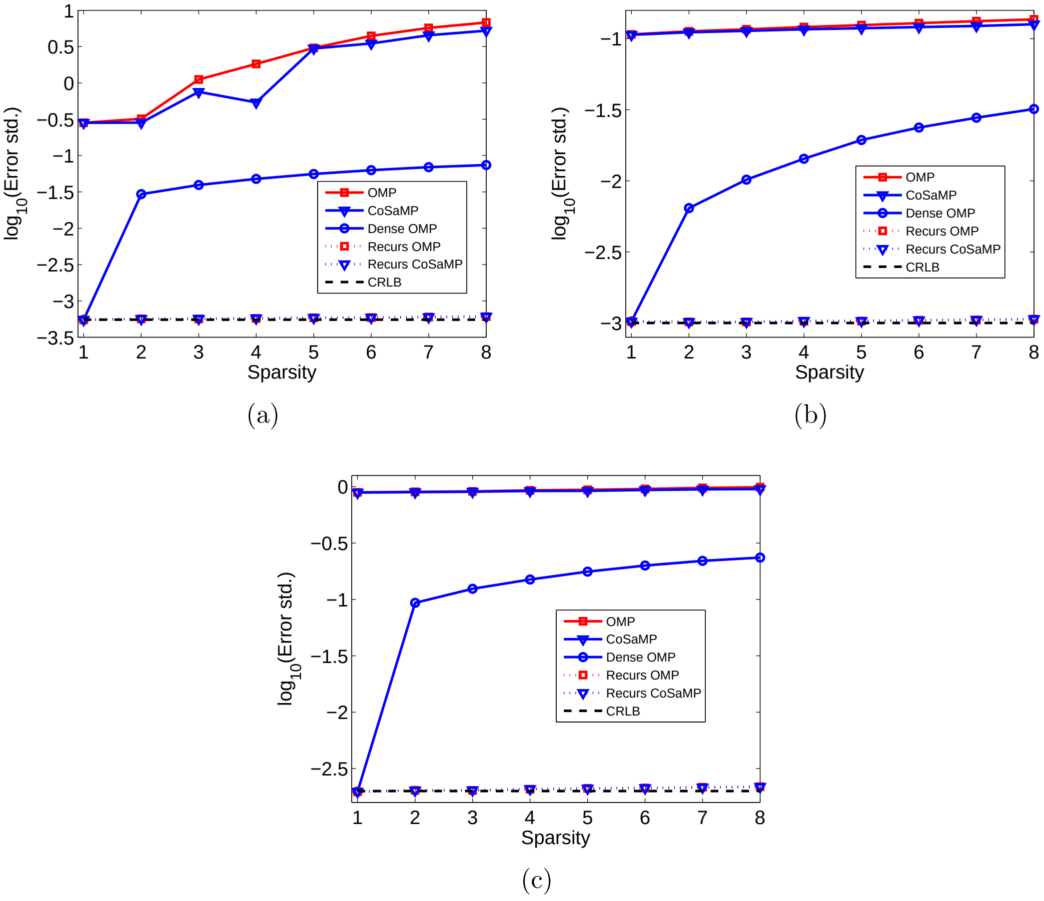
(b)

(c)

Figure 4.5: Standard deviation of the error in the solution of (a) frequency, (b) magnitude, (c) phase of the components with respect to various levels of noise at $K = 5$, $M = 100$.

the error variance. Yet, the improved estimation performance is still far from the CRLB. Though recursive counterparts have a similar break point, their solution accuracy is significantly better at higher SNR with the use of dense grid. Their solution error scales down with the noise variance achieving the CRLB for SNR's greater than 10 dB.

In the second set of simulations, sparsity level of the signal is varied from 1 to 8 while keeping SNR at 40 dB and number of measurement at $M = 100$. Fig. 4.6 provides the standard deviation of the errors with respect to the simulated sparsity levels. One immediate observation is that error variance increases with the number of components. For sparsity $K > 1$, dense grid again provides a limited increase in the estimation performance for OMP. However when there is only one component, OMP with a dense grid is equivalent to spectrum estimation given in (4.23) which is the ML estimate of the parameters and achieves the lower bound. This fact is also observed in Fig. 4.6. Most importantly the recursive solvers provide significantly better estimates then their non-recursive counterparts and achieves the CRLB.



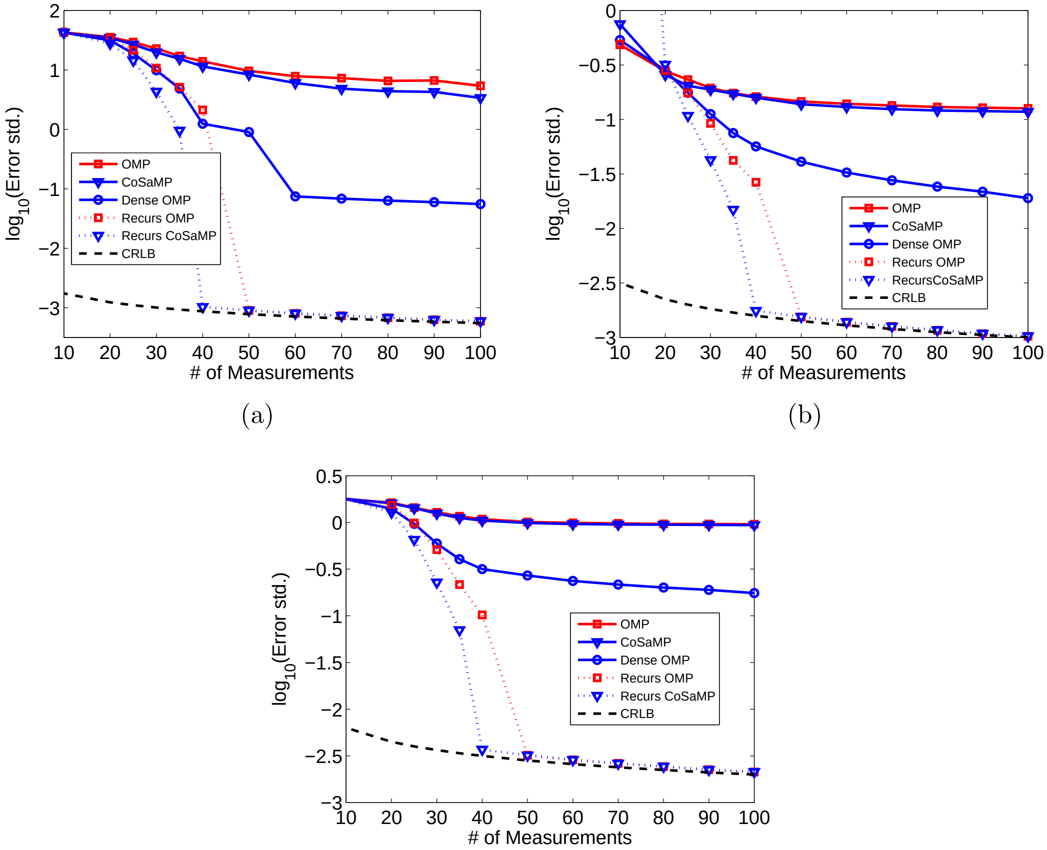
(a)

(b)

(c)

Figure 4.6: Standard deviation of the error in the solution of (a) frequency, (b) magnitude, (c) phase of the components with respect to various levels of sparsity at $M = 100$, SNR $= 40$ dB.

In another set of simulations, number of measurements is varied from 10 to 100 while keeping SNR at 40 dB and sparsity at $K = 5$. Fig. 4.7 provides the standard deviation of the errors with respect to the simulated number of measurements. For the simulated case here, the Nyquist rate sampling would require 400 samples since the bandwidth is chosen as 200Hz and time interval of samples is T = 1 s. As in the noise level analysis, regular solvers and their recursive counterparts have similar break points which is around $10 - 12\%$ of the Nyquist rate samples, i.e. $M \approx 45$. If fewer number of samples are used, both regular solvers and their recursive counterparts results in large error variances. Even though increase in the number of measurements have little impact on the regular solvers due to the off-grid problem, performance of their recursive counterparts achieves the Cramér-Rao lower bound.



(a)

(b)

(c)

Figure 4.7: Standard deviation of the error in the solution of (a) frequency, (b) magnitude, (c) phase of the components with respect to various number of measurements at $K = 5$, SNR = 40 dB.

Important aspect of the proposed recursive framework is its sparse solver oriented design. Discretization interval that allows a solution for K-sparse problems depends heavily on the selected sparse solver. The conditions developed for the solvers are generally pessimistic since they cover the worst case scenarios and give guarantees on recovery of *any* K-sparse signal. However, most of the typical scenarios are well-behaved compared to the worst case. With this optimistic approach, it is possible to relax the guarantees and use denser discretization, allowing more coherence, than the dictated one. Since each solver has a different reaction to the violated recovery guarantees, we conduct the following set of experiments for better characterization of the recovery limits with respect to discretization. Fig. 4.8 shows the reconstruction errors of OMP and CoSaMP for 2 and 3 times denser discretization together with the dictated one in the proposed framework. Having similar support selection metric, CoSaMP is able to operate on the grid designed for OMP. However, since CoSaMP selects a group of support vectors in each iteration, it is very sensitive to higher grid density. On the other hand, since OMP selects a single vector in each iteration, it is robust to denser discretization of the space as long as it is guaranteed that the main problem splits itself all the way down to 1-sparse sub-problems.

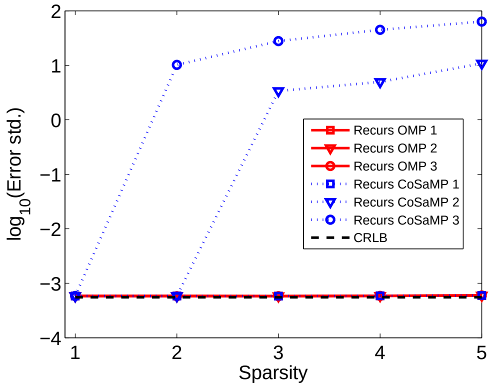


Figure 4.8: Effect of the improper selection of the discretization interval. CoSaMP can operate on the grid designed for OMP, however it is very sensitive to higher grid density.

Finally, complexity analyses provided in (4.32) and (4.42) are validated experimentally. Fig. 4.9 shows the run time of the proposed framework with OMP.

Unoptimized implementation of the algorithms are simulated in MATLAB running on a regular desktop computer. As found analytically, recursion on the unbalanced binary tree has an exponential complexity in terms of the sparsity, whereas balanced binary tree has a complexity almost linear in sparsity.

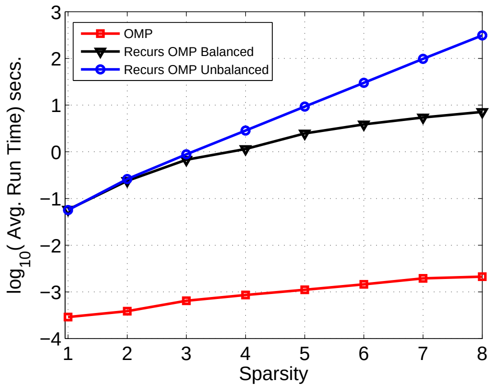


Figure 4.9: Experimental validation and comparison of the complexity analyzes in (4.32) and (4.42). Results are obtained with unoptimized implementations on MATLAB.

When performance-complexity trade-off is considered, CRLB achieving performance of the proposed framework comes with an expense of a higher complexity. Compared to Fig. 4.6 (a), recursive framework decreases the error standard deviation approximately 3 orders of magnitude while requiring 3 orders of magnitude more time. However, gap between the simulated run time is a result of the modular and brute-force implementation of the algorithm. More optimized implementations have the potential to reduce the run time significantly resulting in high performance/complexity gain.

Due to sequential solutions in sub-problems, current form of the algorithm is not suitable for parallelization. However, it is possible to solve each sub-problem independently from each other. Even though this approach is sub-optimal and may take more iterations to converge, independent solutions can run on different processors resulting in savings on the computational time. Although not reported here, preliminary investigations indicate that CRLB is also achievable with a small gap under independent updates.

Chapter 5

Conclusion

Compressive sensing reconstruction techniques suffer from significant performance degradation when there is a mismatch between the actual and the chosen signal dictionaries. Such mismatch generally occurs in practice due to modeling errors, parameter space discritization or simply by sampling jitter. In general, this mismatch problem cannot be solved using denser basis. Also, density of the basis adversely affects the algorithms due to high correlation. In this thesis, novel perturbed greedy reconstruction techniques are proposed for the case of signal reconstruction in the presence of structured and unstructured perturbations in the signal dictionary. Proposed the first technique, named as Parameter Perturbed OMP (PPOMP), adapts the signal dictionary to the actual measurements by performing perturbations of the parameters governing the signal dictionary. To quantify the performance, Kullback-Leibler Divergence is proposed as the error metric for off-grid target reconstruction performance. The performance of the proposed algorithm is investigated on a delay-Doppler radar formulation that implies the off-grid problem. Compared to the tested techniques, proposed method provides significantly higher performance for a wide range of sparsity and SNR levels. Proposed the second technique, named as Perturbed OMP (POMP), performs controlled rotation based perturbations of selected support vectors. Limits on the required perturbation for exact fit to observation signal at any sparsity is found. Also, for a given acceptable residual level, limits on the required perturbation are also provided. A sufficiency condition that assures monotonic decrease on the required perturbations as a function of sparsity is given. The performance of the proposed algorithm is investigated on a time jitter example that causes a unstructured dictionary mismatch. As an extension to SNR, Signal-to-Perturbation (SPR) and Noise-to-Perturbation (NPR) are used for better characterization of the performance limits of POMP. Results show that, in comparison with well known CS reconstruction techniques, POMP provides efficient reconstructions with significantly lower error in a wide range of sparsity levels. Unlike the proposed perturbed techniques, this thesis also introduces a novel recursive framework. The proposed framework partitions the main problem into smaller sub-problems and discretizes the given continuous parameter space adaptively depending on the sparsity of the problem in order to guarantee the reconstruction with the specified sparse solver. The Cramér-Rao lower bound for the random sampling case is investigated in order to find the finest discretization. Theoretical complexity analyzes for different partitioning approaches are provided. The performance of the proposed framework is investigated in sparse spectral estimation problem. Results show that, in comparison with the direct usage, in the proposed framework regular solvers can result significantly lower error variances achieving the CRLB. Due to its modular structure, the proposed framework is customizable and accepts any sparse solver. Also, its parallelizable form allows high performance/complexity gains in multi-processors system.

Bibliography

[1] D. Donoho, “Compressed sensing,” *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289 – 1306, 2006.

[2] E. Candes, J. Romberg, and T. Tao, “Robust uncertanity principles: Exact signal reconstruction from highly incomplete frequency information,” *IEEE Transactions on Information Theory*, vol. 52, pp. 489–509, 2006.

[3] M. D. ad M. Davenport, D. Takhar, J. Laska, T. Sun, K. Kelly, and R. Baraniuk, “Single-pixel imaging via compressive sampling,” *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 83–91, 2008.

[4] M. Lustig, D. Donoho, and J. Pauly, “Sparse mri: The application of compressed sensing for rapid MR imaging,” *Magnetic Resonance in Medicine*, vol. 58, pp. 1182–1195, Dec. 2007.

[5] R. Baraniuk and P. Steeghs, “Compressive radar imaging,” in *IEEE Radar Conf.*, pp. 128–133, 2007.

[6] V. Patel, G. Easley, D. H. Jr., and R. Chellappa, “Compressed synthetic aperture radar,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 2, pp. 244–254, 2010.

[7] W. Bajwa, J. Haupt, A. Sayeed, and R. Nowak, “Compressive wireless sensing,” in *Int. Conf. on Information Processing in Sensor Networks (IPSN)*, pp. 134–142, April 2006.

[8] E. Candes, J. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Comm. on Pure and Applied Math.*, vol. 59, no. 8, pp. 1207–1223, 2006.

[9] J. Haupt and R. Nowak, "Signal reconstruction from noisy random projections," *IEEE Transactions on Information Theory*, vol. 52, no. 9, pp. 4036–4048, 2006.

[10] D. Donoho, M. Elad, and V. Temlyakov, "Stable recovery of sparse overcomplete representations in the presence of noise," *IEEE Transactions on Information Theory*, vol. 52, no. 1, pp. 6–18, 2006.

[11] S. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Transactions on Signal Processing*, vol. 41, pp. 3397–3415, 1993.

[12] J. Tropp and A. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Transactions on Information Theory*, vol. 53, no. 12, pp. 4655–4666, Dec. 2007.

[13] D. Needell and J. Tropp, "Cosamp: Iterative signal recovery from incomplete and inaccurate samples," *Applied and Computational Harmonic Analysis*, vol. 26, no. 3, pp. 301 – 321, 2009.

[14] T. Blumensath and M. E. Davies, "Iterative hard thresholding for compressed sensing," *Applied and Computational Harmonic Analysis*, vol. 27, no. 3, pp. 265–274, 2009.

[15] D. L. Donoho, A. Maleki, and A. Montanari, "Message passing algorithms for compressed sensing," *Proc. Nat. Acad. Sci.*, vol. 106, no. 45, pp. 18914–18919, Nov. 2009.

[16] R. G. Baraniuk, V. Cevher, M. F. Duarte, and C. Hedge, "Model-based compressive sensing," *IEEE Transactions on Information Theory*, vol. 56, no. 4, pp. 1982–2001, April 2010.

[17] S. Som and P. Schniter, "Compressive imaging using approximate message passing and a markov-tree prior," *IEEE Transactions on Signal Processing*, vol. 60, no. 7, pp. 3439–3448, July 2010.

[18] V. Cevher, M. Duarte, and R. Baraniuk, "Distributed target localization via spatial sparsity," in *European Signal Processing Conf. (EUSIPCO)*, pp. 134 – 142, August 2008.

[19] J. H. Ender, "On compressive sensing applied to radar," *Signal Processing*, vol. 90, no. 5, pp. 1402–1414, 2010.

[20] A. C. Gurbuz, J. H. McClellan, and W. R. Scott Jr., "A compressive sensing data acquisition and imaging method for stepped frequency gprs," *IEEE Transactions on Signal Processing*, vol. 57, no. 7, pp. 2640–2650, 2009.

[21] A. C. Gurbuz, V. Cevher, and J. H. McClellan, "A compressive beamformer," in *ICASSP 2008*, (Las Vegas, Nevada), 2008.

[22] Y. Yu, A. P. Petropulu, and H. V. Poor, "Mimo radar using compressive sampling," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 1, pp. 146–163, 2010.

[23] N. Aggarwal and W. C. Karl, "Line detection in images through Regularized Hough Transform," *IEEE Transactions on Image Processing*, vol. 15, pp. 582–590, 2006.

[24] M. A. Tuncer and A. C. Gurbuz, "Analysis of unknown velocity and target off the grid problems in compressive sensing based subsurface imaging," in *ICASSP 2011*, (Prague, Czech Republic), pp. 2880–2883, 2011.

[25] Y. Chi, L. Scharf, A. Pezeshki, and R. Calderbank, "The sensitivity to basis mismatch of compressed sensing in spectrum analysis and beamforming," in *Sixth Workshop on Defense Applications of Signal Processing (DASP)*, 2009.

[26] R. Baraniuk, M. Davenport, R. Devore, and M. Wakin, "A simple proof of the restricted isometry property for random matrices," *Constr. Approx*, 2007.

[27] M. Herman and T. Strohmer, "General deviants: An analysis of perturbations in compressed sensing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 2, pp. 342–349, 2010.

[28] Y. Chi, L. Scharf, A. Pezeshki, and R. Calderbank, “Sensitivity of basis mismatch to compressed sensing,” *IEEE Transactions on Signal Processing*, vol. 59, pp. 2182–2195, 2011.

[29] D. H. Chae, P. Sadeghi, and R. A. Kennedy, “Effects of basis-mismatch in compressive sampling of continuous sinusoidal signals,” in *International Conference on Future Computer and Communication (ICFCC)*, pp. 739–743, 2010.

[30] G. Peyre, “Best basis compressed sensing,” *IEEE Transaction on Signal Processing*, vol. 58, no. 5, pp. 2613–2622, 2010.

[31] C. Ekanadham, D. Tranchina, and E. Simoncelli, “Recovery of sparse translation-invariant signals with continuous basis pursuit,” *IEEE Transactions on Signal Processing*, vol. 59, no. 10, pp. 4735–4744, 2011.

[32] Z. Yang, C. Zhang, and L. Xie, “Robustly stable signal recovery in compressed sensing with structured matrix perturbation,” *IEEE Transactions on Signal Processing*, vol. 60, pp. 4658–4671, sept. 2012.

[33] H. Zhu, G. Leus, and G. Giannakis, “Sparsity-cognizant total least-squares for perturbed compressive sampling,” *IEEE Transaction on Signal Processing*, vol. 59, no. 5, pp. 2002–2016, 2011.

[34] M. Elad, *Sparse and Redundant Representations.* Springer, 2010.

[35] S. Foucart and H. Rauhut, *A Mathematical Introduction to Compressive Sensing.* Springer New York, 2013.

[36] http://dsp.rice.edu/cs, “Compressive sensing resources,” 19/06/2014.

[37] S. Chen, D. Donoho, and M. Saunders, “Atomic decomposition by basis pursuit,” *SIAM Journal on Scientific Computing*, vol. 20, no. 1, pp. 33–61, 1998.

[38] D. P. Bertsekas, *Nonlinear Programming.* Athena Scientific, 1999.

[39] S. Boyd and L. Vandenberghe, *Convex Optimization.* Cambridge University Press, 2004.

[40] A. Bandeira, E. Dobriban, D. Mixon, and W. Sawin, “Certifying the restricted isometry property is hard,” *IEEE Transactions on Information Theory*, vol. 59, pp. 3448–3450, June 2013.

[41] M. Davenport and M. Wakin, “Analysis of orthogonal matching pursuit using the restricted isometry property,” *IEEE Transactions on Information Theory*, vol. 56, no. 9, pp. 4395–4401, 2010.

[42] J. Wang and B. Shim, “On the recovery limit of sparse signals using orthogonal matching pursuit,” *IEEE Transactions on Signal Processing*, vol. 60, pp. 4973–4976, Sept 2012.

[43] J. Tropp, “Greed is good: Algorithmic results for sparse approximation,” *IEEE Transactions on Information Theory*, vol. 50, no. 10, pp. 2231–2242, Oct. 2004.

[44] O. Teke, A. C. Gurbuz, and O. Arikan, “A robust compressive sensing based technique for reconstruction of sparse radar scenes,” *Digital Signal Processing*, vol. 27, pp. 23–32, 2014.

[45] O. Teke, O. Arikan, and A. C. Gurbuz, “Sparse delay-doppler image reconstruction under off-grid problem,” in *Sensor Array and Multichannel Signal Processing Workshop (SAM), 2014 IEEE 8th*, June 2014.

[46] M. Skolnik, *Introduction to Radar Systems.* McGraw Hill, 1980.

[47] M. Richards, *Fundementals of Radar Signal Processing.* McGraw Hill, 2005.

[48] J. Li and P. Stoica, *MIMO Radar Signal Processing.* Wiley, 2008.

[49] M. Herman and T. Strohmer, “High-resolution radar via compressed sensing,” *IEEE Transactions on Signal Processing*, vol. 57, no. 6, pp. 2275–2284, 2009.

[50] D. Donoho, “Superresolution via sparsity constraints,” *SIAM J. Math. Analysis*, vol. 23, no. 5, pp. 1309–1331, 1992.

[51] O. Teke, A. C. Gurbuz, and O. Arikan, “A new omp technique for sparse recovery,” in *20th Signal Processing and Communications Applications Conference (SIU)*, pp. 1–4, April 2012.

[52] O. Teke, A. Gurbuz, and O. Arikan, “Perturbed orthogonal matching pursuit,” *IEEE Transactions on Signal Processing*, vol. 61, no. 24, pp. 6220–6231, 2013.

[53] M. Pilanci, O. Arikan, and M. Pinar, “Structured least squares problems and robust estimators,” *IEEE Transactions on Signal Processing*, vol. 58, no. 5, pp. 2453–2465, May 2010.

[54] Z. Yang, L. Xie, and C. Zhang, “Off-grid direction of arrival estimation using sparse bayesian inference,” *IEEE Transactions on Signal Processing*, vol. 61, no. 1, pp. 38–43, 2013.

[55] O. Teke, O. Arikan, and A. C. Gurbuz, “A recursive approach to reconstruction of sparse signals,” in *Signal Processing and Communications Applications Conference (SIU), 2014 22nd*, pp. 1142–1145, April 2014.

[56] O. Teke, A. Gurbuz, and O. Arikan, “Recursive implementation of compressive sensing techniques for improved performance,” *Submitted to IEEE Transactions on Signal Processing*, 2014.

[57] A. Dempster, N. Laird, and D. Rubin, “Maximum likelihood from incomplete data via the em algorithm,” *Journal of the Royal Statistical Society Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.

[58] A. Gurbuz, M. Pilanci, and O. Arikan, “Expectation maximization based matching pursuit,” in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pp. 3313–3316, March 2012.

[59] D. Rife and R. Boorstyn, “Single tone parameter estimation from discrete-time observations,” *IEEE Transactions on Information Theory*, vol. 20, no. 5, pp. 591–598, 1974.

[60] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data.* Prentice-Hall, Inc., 1988.

Appendix A

Proofs

A.1 Lipschitz Continuity of the Delay-Doppler Objective Function

After the linearization of (3.13a), normalized cost function becomes $J(\boldsymbol{u}) = \|\boldsymbol{r}_l - \boldsymbol{B}_l\boldsymbol{u}\|_2^2/\|\boldsymbol{y}\|_2^2$ and its gradient is $\nabla J(\boldsymbol{u}) = -(2/\|\boldsymbol{y}\|_2^2)\boldsymbol{B}_l^H(\boldsymbol{r}_l - \boldsymbol{B}_l\boldsymbol{u})$. Therefore,

$$\begin{aligned}\frac{\|\nabla J(\boldsymbol{u}_1) - \nabla J(\boldsymbol{u}_2)\|_2}{\|\boldsymbol{u}_1 - \boldsymbol{u}_2\|_2} &\leq \frac{2}{\|\boldsymbol{y}\|_2^2}\|\boldsymbol{B}_l^H\boldsymbol{B}_l\|_2 \\ &\leq \frac{2}{\|\boldsymbol{y}\|_2^2}\|\boldsymbol{B}_l^H\boldsymbol{B}_l\|_* = \frac{2}{\|\boldsymbol{y}\|_2^2}\, tr(\boldsymbol{B}_l^H\boldsymbol{B}_l). \qquad \text{(A.1)}\end{aligned}$$

Furthermore, we can expand the trace as:

$$\begin{aligned}tr(\boldsymbol{B}_l^H\boldsymbol{B}_l) &= \sum_{i=1}^{k}\left\|\Delta_\tau\alpha_{i,l}\frac{\partial\boldsymbol{a}}{\partial\tau_{i,l}}\right\|_2^2 + \sum_{i=1}^{k}\left\|\Delta_\nu\alpha_{i,l}\frac{\partial\boldsymbol{a}}{\partial\nu_{i,l}}\right\|_2^2 \\ &= \sum_{i=1}^{k}|\alpha_{i,l}|^2\Big(\Delta_\tau^2\Big\|\frac{\partial\boldsymbol{a}}{\partial\tau_{i,l}}\Big\|_2^2 + \Delta_\nu^2\Big\|\frac{\partial\boldsymbol{a}}{\partial\nu_{i,l}}\Big\|_2^2\Big). \qquad \text{(A.2)}\end{aligned}$$

Depending on the definition of the basis function in (3.21), the i^{th} index of $\boldsymbol{a}(\tau,\nu)$ is $a(t_i;\tau,\nu) = s(t_i - \tau)\, e^{j2\pi\nu t_i}$ where t_i is the i^{th} time sample and $s(t) = e^{jf(t)}$ is the radar waveform. Hence, $a(t_i;\tau,\nu) = e^{j(2\pi\nu t_i + f(t_i-\tau))}$ and norm of the partial derivative of $a(t_i;\tau,\nu)$ with respect to τ can be written as:

$$\Big|\frac{\partial a(t_i;\tau,\nu)}{\partial\tau}\Big| = |-jf'(t_i-\tau)\,e^{j(2\pi\nu t_i+f(t_i-\tau))}| = |f'(t_i-\tau)|. \tag{A.3}$$

For linear chirp signals, $f(t)=\pi\beta(t-T_p/2)^2$, and $f'(t)=2\pi\beta(t-T_p/2)$. Since pulse duration is $0\le t\le T_p$, we have $|f'(t)|\le\pi\beta T_p$. Therefore, norm of the partial derivative can be bounded as:

$$\Big|\frac{\partial a(t_i;\tau,\nu)}{\partial\tau}\Big| = |f'(t_i-\tau)| \le \pi\beta T_p. \tag{A.4}$$

Using (A.4), $\|\partial\boldsymbol{a}/\partial\tau\|_2^2$ can be bounded as:

$$\left\|\frac{\partial\boldsymbol{a}}{\partial\tau}\right\|_2^2 = \sum_{i=1}^{M}\Big|\frac{\partial a(t_i;\tau,\nu)}{\partial\tau}\Big|^2 \le M\pi^2\beta^2T_p^2 = M\pi^2\Delta_\tau^{-2} \tag{A.5}$$

where the last part follows with the selection of $\beta=2B/T_p$, and $\Delta_\tau=1/(2B)$ is the Rayleigh resolution spacing. Similarly, norm of the partial derivative of $a(t_i;\tau,\nu)$ with respect to ν can be written as:

$$\Big|\frac{\partial a(t_i;\tau,\nu)}{\partial\nu}\Big| = |-j2\pi t_i\,e^{j(2\pi\nu t_i+f(t_i-\tau))}| = |2\pi t_i|. \tag{A.6}$$

Since time samples are taken from $[0,\ N_pT_{PRI}]$ range, we have $|2\pi t_i|\le 2\pi N_pT_{PRI}$. Norm of the partial derivative can be bounded as:

$$\Big|\frac{\partial a(t_i;\tau,\nu)}{\partial\nu}\Big| = |2\pi t_i| \le 2\pi N_pT_{PRI}. \tag{A.7}$$

Using (A.7), $\|\partial\boldsymbol{a}/\partial\nu\|_2^2$ can be bounded as:

$$\left\|\frac{\partial\boldsymbol{a}}{\partial\nu}\right\|_2^2 = \sum_{i=1}^{M}\Big|\frac{\partial a(t_i;\tau,\nu)}{\partial\nu}\Big|^2 \le M4\,\pi^2\,N_p^2\,T_{PRI}^2 = M4\pi^2\Delta_\nu^{-2} \tag{A.8}$$

where $\Delta_\nu = 1/(N_p T_{PRI})$ is the Rayleigh resolution spacing. Combining (A.2), (A.5) and (A.8), $tr(\boldsymbol{B}_l^H \boldsymbol{B}_l)$ can be upper bounded as:

$$tr(\boldsymbol{B}_l^H \boldsymbol{B}_l) \leq \sum_{i=1}^{k} |\alpha_{1,l}|^2 (M\pi^2 + 4M\pi^2) = 5\pi^2 M \|\boldsymbol{\alpha}_l\|_2^2. \tag{A.9}$$

Notice that $\boldsymbol{\alpha}_l$ is the coefficients of the projection of $\boldsymbol{y}$ onto the estimated parameters. Since $\|\boldsymbol{a}(\tau, \nu)\|_2^2 = M$, we have $M\|\boldsymbol{\alpha}_l\|_2^2 \approx \|\boldsymbol{y}\|_2^2$. Using (A.9), (A.1) can be upper bounded as:

$$\frac{\|\nabla J(\boldsymbol{u}_1) - \nabla J(\boldsymbol{u}_2)\|_2}{\|\boldsymbol{u}_1 - \boldsymbol{u}_2\|_2} \leq 10\pi^2 = L. \tag{A.10}$$

A.2 Proof of Theorem 4

Let $\boldsymbol{S}_k = [\boldsymbol{s}_1,\, \boldsymbol{s}_2,\, \ldots, \boldsymbol{s}_k]$ be the matrix whose columns correspond to the current estimate for the support of $\boldsymbol{x}$. Since $\boldsymbol{S}_k$ is full rank, we can define $\boldsymbol{M}_k = (\boldsymbol{S}_k^T \boldsymbol{S}_k)^{-1}$ and $\boldsymbol{S}_k^\dagger = \boldsymbol{M}_k \boldsymbol{S}_k^T$. The following equations provide recursive relations for $\boldsymbol{M}_{k+1}$ and $\boldsymbol{S}_{k+1}^\dagger$:

$$\boldsymbol{M}_{k+1} = \begin{pmatrix} \boldsymbol{M}_k + d(\boldsymbol{S}_k^\dagger \boldsymbol{s}_{k+1})(\boldsymbol{S}_k^\dagger \boldsymbol{s}_{k+1})^T & -d(\boldsymbol{S}_k^\dagger \boldsymbol{s}_{k+1}) \\ -d(\boldsymbol{S}_k^\dagger \boldsymbol{s}_{k+1})^T & d \end{pmatrix}, \tag{A.11}$$

$$\boldsymbol{S}_{k+1}^\dagger = \boldsymbol{M}_{k+1} \boldsymbol{S}_{k+1}^T = \begin{pmatrix} \boldsymbol{S}_k^\dagger - d\, \boldsymbol{S}_k^\dagger\, \boldsymbol{s}_{k+1}\, \boldsymbol{s}_{k+1,r}^T \\ d\, \boldsymbol{s}_{k+1,r}^T \end{pmatrix}, \tag{A.12}$$

where $d = 1/\|\boldsymbol{s}_{k+1,r}\|_2^2$ and $\boldsymbol{s}_{k+1,r} = \boldsymbol{s}_{k+1} - \boldsymbol{S}_k \boldsymbol{S}_k^\dagger \boldsymbol{s}_{k+1}$. Since the ℓ_1 norm of a partitioned vector is the sum of the ℓ_1 norms of each partition:

$$\|\boldsymbol{S}_{k+1}^\dagger \boldsymbol{y}\|_1 = \|\boldsymbol{S}_k^\dagger \boldsymbol{y} - d\, \boldsymbol{S}_k^\dagger\, \boldsymbol{s}_{k+1}\, \boldsymbol{s}_{k+1,r}^T \boldsymbol{y}\|_1 + \|d\, \boldsymbol{s}_{k+1,r}^T \boldsymbol{y}\|_1, \tag{A.13}$$

which can be written as:

$$\|\boldsymbol{S}_{k+1}^\dagger \boldsymbol{y}\|_1 = \|\boldsymbol{S}_k^\dagger (\boldsymbol{y} - \boldsymbol{u})\|_1 + |\lambda|, \tag{A.14}$$

where $\lambda = d\, \boldsymbol{s}_{k+1,r}^T \boldsymbol{y}$ and $\boldsymbol{u} = \lambda\, \boldsymbol{s}_{k+1}$.

To have $\phi_{k+1}^* \leq \phi_k^*$, we need the following inequality to hold true:

$$\frac{\|\boldsymbol{\alpha}^{(k)}\|_1}{\|\boldsymbol{\alpha}^{(k+1)}\|_1} \leq \frac{\|\boldsymbol{y}_\perp^{(k)}\|_2}{\|\boldsymbol{y}_\perp^{(k+1)}\|_2} = \beta. \tag{A.15}$$

Using the triangle inequality on Eq. A.14, the following bound can be obtained,

$$|\lambda| - \|\boldsymbol{S}_k^\dagger \boldsymbol{u}\|_1 \leq \|\boldsymbol{S}_{k+1}^\dagger \boldsymbol{y}\|_1 - \|\boldsymbol{S}_k^\dagger \boldsymbol{y}\|_1. \tag{A.16}$$

Since $\|\boldsymbol{S}_k^\dagger \boldsymbol{u}\|_1 = \|\boldsymbol{S}_k^\dagger \lambda\, \boldsymbol{s}_{k+1}\|_1 = |\lambda|\, \|\boldsymbol{S}_k^\dagger \boldsymbol{s}_{k+1}\|_1$, Eq. A.16 becomes:

$$|\lambda|\, (1 - \|\boldsymbol{S}_k^\dagger \boldsymbol{s}_{k+1}\|_1) \leq \|\boldsymbol{S}_{k+1}^\dagger \boldsymbol{y}\|_1 - \|\boldsymbol{S}_k^\dagger \boldsymbol{y}\|_1. \tag{A.17}$$

By adding $(\beta - 1)\|\boldsymbol{S}_{k+1}^\dagger \boldsymbol{y}\|_1$ to both sides of Eq. A.17, we obtain:

$$(\beta - 1)\|\boldsymbol{S}_{k+1}^\dagger \boldsymbol{y}\|_1 + |\lambda|\, (1 - \|\boldsymbol{S}_k^\dagger \boldsymbol{s}_{k+1}\|_1) \leq \beta\|\boldsymbol{S}_{k+1}^\dagger \boldsymbol{y}\|_1 - \|\boldsymbol{S}_k^\dagger \boldsymbol{y}\|_1. \tag{A.18}$$

The desired condition on angle decrease requires $\beta\|\boldsymbol{S}_{k+1}^\dagger \boldsymbol{y}\|_1 \geq \|\boldsymbol{S}_k^\dagger \boldsymbol{y}\|_1$, which is always achieved if:

$$0 \leq (\beta - 1)\|\boldsymbol{S}_{k+1}^\dagger \boldsymbol{y}\|_1 + |\lambda|\, (1 - \|\boldsymbol{S}_k^\dagger \boldsymbol{s}_{k+1}\|_1), \tag{A.19}$$

or equivalently:

$$\|\boldsymbol{S}_k^\dagger \boldsymbol{s}_{k+1}\|_1 \leq 1 + \frac{(\beta - 1)\|\boldsymbol{\alpha}^{(k+1)}\|_1}{|\lambda|}. \tag{A.20}$$

Therefore, by using Eq. A.15, Eq. A.20 and definition of λ, the non-negative constant γ in the statement of Theorem 4 can be obtained as:

$$\gamma = \frac{\left(\dfrac{\left\|\boldsymbol{y}_\perp^{(k)}\right\|_2}{\left\|\boldsymbol{y}_\perp^{(k+1)}\right\|_2} - 1 \right) \left\|\boldsymbol{\alpha}^{(k+1)}\right\|_1 \left\|\boldsymbol{s}_{k+1} - \boldsymbol{S}_k \boldsymbol{S}_k{}^\dagger \boldsymbol{s}_{k+1}\right\|_2^2}{\left|\boldsymbol{s}_{k+1}^T\, \boldsymbol{y}_\perp^{(k)}\right|}, \text{ q.e.d.} \tag{A.21}$$

A.3 Derivation of CRLB for Single Frequency Estimation

For sampling times $\boldsymbol{t} \in \Re^M$, observations of a single frequency tone in noise can be written as:

$$\begin{aligned} y_i &= \alpha\, e^{j\phi}\, e^{j\,\omega\, t_i} + n_i, \\ &= \alpha\ \cos(\omega\, t_i + \phi) + w_i + j\big(\alpha\ \sin(\omega\, t_i + \phi) + u_i\big) \end{aligned} \tag{A.22}$$

where w_i and u_i are i.i.d. Gaussian noise with zero mean and variance σ^2. Since the noise is circularly symmetric, real and imaginary parts of the observations are independent. Therefore we can write the combined observation vector as:

$$\boldsymbol{y} = \underbrace{\begin{bmatrix} \alpha\ \cos(\omega\, t_1 + \phi) \\ \vdots \\ \alpha\ \cos(\omega\, t_M + \phi) \\ \alpha\ \sin(\omega\, t_1 + \phi) \\ \vdots \\ \alpha\ \sin(\omega\, t_M + \phi) \end{bmatrix}}_{\boldsymbol{\mu}(\alpha,\phi,\omega)} + \underbrace{\begin{bmatrix} w_1 \\ \vdots \\ w_M \\ u_1 \\ \vdots \\ u_M \end{bmatrix}}_{\boldsymbol{z}} = \boldsymbol{\mu}(\alpha,\phi,\omega) + \boldsymbol{z}, \tag{A.23}$$

where each z_i is an i.i.d. Gaussian noise with zero mean and variance σ^2. Therefore, $\boldsymbol{y}$ is an i.i.d. Gaussian with mean $\boldsymbol{\mu}(\alpha,\phi,\omega)$ and variance σ^2. Since covariance matrix is independent from the parameters and diagonal with constant elements, the Fisher information matrix is written as:

$$\boldsymbol{I} = \frac{1}{\sigma^2}\begin{bmatrix} \frac{\partial\boldsymbol{\mu}^T}{\partial\alpha}\frac{\partial\boldsymbol{\mu}}{\partial\alpha} & \frac{\partial\boldsymbol{\mu}^T}{\partial\alpha}\frac{\partial\boldsymbol{\mu}}{\partial\phi} & \frac{\partial\boldsymbol{\mu}^T}{\partial\alpha}\frac{\partial\boldsymbol{\mu}}{\partial\omega} \\ \frac{\partial\boldsymbol{\mu}^T}{\partial\phi}\frac{\partial\boldsymbol{\mu}}{\partial\alpha} & \frac{\partial\boldsymbol{\mu}^T}{\partial\phi}\frac{\partial\boldsymbol{\mu}}{\partial\phi} & \frac{\partial\boldsymbol{\mu}^T}{\partial\phi}\frac{\partial\boldsymbol{\mu}}{\partial\omega} \\ \frac{\partial\boldsymbol{\mu}^T}{\partial\omega}\frac{\partial\boldsymbol{\mu}}{\partial\alpha} & \frac{\partial\boldsymbol{\mu}^T}{\partial\omega}\frac{\partial\boldsymbol{\mu}}{\partial\phi} & \frac{\partial\boldsymbol{\mu}^T}{\partial\omega}\frac{\partial\boldsymbol{\mu}}{\partial\omega} \end{bmatrix}. \tag{A.24}$$

Explicitly computing each element:

$$\frac{\partial\boldsymbol{\mu}^T}{\partial\alpha}\frac{\partial\boldsymbol{\mu}}{\partial\alpha} = M, \quad \frac{\partial\boldsymbol{\mu}^T}{\partial\alpha}\frac{\partial\boldsymbol{\mu}}{\partial\phi} = \frac{\partial\boldsymbol{\mu}^T}{\partial\alpha}\frac{\partial\boldsymbol{\mu}}{\partial\omega} = 0.$$

$$\frac{\partial \boldsymbol{\mu}^T}{\partial \phi}\frac{\partial \boldsymbol{\mu}}{\partial \phi} = \alpha^2 M,\ \frac{\partial \boldsymbol{\mu}^T}{\partial \phi}\frac{\partial \boldsymbol{\mu}}{\partial \omega} = \alpha^2 \sum_{i=1}^{M} t_i,\ \frac{\partial \boldsymbol{\mu}^T}{\partial \omega}\frac{\partial \boldsymbol{\mu}}{\partial \omega} = \alpha^2 \sum_{i=1}^{M} t_i^2$$

Assuming sampling times are taken from an i.i.d. random variables:

$$E[\boldsymbol{I}] = \frac{M\alpha^2}{\sigma^2}\begin{bmatrix} 1/\alpha^2 & 0 & 0 \\ 0 & 1 & E[t_i] \\ 0 & E[t_i] & E[t_i^2] \end{bmatrix}.$$

Then the inverse of the expected Fisher information matrix, hence the corresponding CRLB are as follows:

$$\big(E[\boldsymbol{I}]\big)^{-1} = \frac{(\sigma/\alpha)^2}{M}\begin{bmatrix} \alpha^2 & 0 & 0 \\ 0 & E[t_i^2]/var(t_i) & -E[t_i]/var(t_i) \\ 0 & -E[t_i]/var(t_i) & 1/var(t_i) \end{bmatrix}.$$