RECOVERY OF SPARSE SIGNALS AND PARAMETER PERTURBATIONS FROM PARAMETERIZED SIGNAL MODELS

BY

ERIK C. JOHNSON

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2013

Urbana, Illinois

Adviser:

Professor Douglas L. Jones

ABSTRACT

Estimating unknown signals from parameterized measurement models is a common problem that arises in diverse areas such as statistics, imaging, machine learning, and signal processing. In many of these problems, however, only a limited amount of data is available to recover the unknown signal. Additional constraints are required to successfully recover the unknown signal if there are more unknowns than measurements. Sparsity has proven to be a powerful constraint for signal recovery when the unknown signal has few nonzero elements. If a signal is sparse in a parameterized measurement model, the model parameters must be known to recover the signal. An example of this problem is the recovery of a signal that is a sum of a small number of sinusoids. Reconstruction of this signal requires recovery of both the amplitude of the sinusoids as well as their frequency parameters. Applying traditional sparse reconstruction techniques to such problems requires a dense oversampling of the parameter space.

As an alternative to existing methods, this work proposes an optimization problem to recover sparse signals and sparse parameter perturbations from few measurements given a parameterized model and an initial set of parameter estimates. This problem is then solved by a newly developed Successive Linearized Programming for Sparse Representations algorithm, which is guaranteed to converge to a first-order critical point. For simulated recovery of four sinusoids from 16 noiseless measurements, this method is able to perfectly recover the signal amplitudes and parameters whereas existing approaches have significant error. To demonstrate the potential application of the proposed technique to real-world problems, the novel algorithm is used to find sparse representations of real-world Radio Frequency data. With this dataset, the proposed technique is able to produce sparse recoveries without highly oversampled dictionaries and actually produces sparser solutions than standard sparse recovery techniques with oversampled dictionaries.

To my parents, Brad and Vicki, my family, and my lovely Sarah

ACKNOWLEDGMENTS

I would like to thank my adviser, Douglas Jones, for his patience and guidance. Also, my labmates Cagdas Tuna, Dave Jun, Dave Cohen, and Nam Nguyen have contributed to this thesis through countless discussions. I would also like to thank the generous support of NSF EFRI-BSBA Award No. 0938007 and NSF IGERT Grant No. 0903622T. Professor Steve Franke collected the Radio Frequency data used in this thesis. Without his invaluable contribution, this thesis would not be possible.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS vi

CHAPTER 1 INTRODUCTION 1
1.1 Prior Work on Sparse Recovery with Unknown Signal Dictionaries 3
1.2 Example Recovery of Frequency-Sparse Signals 5
1.3 Proposed Solution 11
1.4 Comparison to Existing Sparse Recovery Formulations 12
1.5 Contributions of This Thesis 14

CHAPTER 2 PROPOSED PARAMETERIZED SPARSE RECOVERY FORMULATION AND ALGORITHM 16
2.1 Sparse Recovery Algorithm 18
2.2 First-Order Necessary Conditions for Optimality 20
2.3 Solution of SLPSR Subproblems 24
2.4 Convergence of the SLPSR Algorithm 30
2.5 Selection of Regularization Parameters 33
2.6 Initialization of SLPSR with BPDN Solution 34
2.7 Relationship to MAP Estimation 34

CHAPTER 3 NUMERICAL EXPERIMENTS 38
3.1 Implementation of Sparse Recovery Methods in MATLAB . . 38
3.2 Recovery of Simulated Frequency-Sparse Signals 42
3.3 Parameter Sensitivity 50
3.4 Example of Super-Resolution 53
3.5 Simulated Source Localization with a Linear Array 55

CHAPTER 4 RECONSTRUCTION RESULTS FOR SPARSE SIGNAL RECOVERY OF RADIO FREQUENCY DATA 60
4.1 Reconstruction of Radio-Frequency Data 61
4.2 Improving Modeling Using Transition Elements 72

CHAPTER 5 DISCUSSION AND CONCLUSIONS 80
5.1 Conclusions and Future Work 82

REFERENCES 85

LIST OF ABBREVIATIONS

AA-P-BPDN	Alternating Algorithm Perturbed Basis Pursuit Denoising
BP	Basis Pursuit
BPDN	Basis Pursuit Denoising
CS	Compressed Sensing
dB	Decibel
DFT	Discrete Fourier Transform
HF	High Frequency
MAP	Maximum *a Posteriori*
PSR	Perturbed Sparse Recovery
RF	Radio Frequency
SLPSR	Successive Linearized Programming for Sparse Recovery
SNR	Signal to Noise Ratio
SOCP	Second-Order Conic Programming
STLS	Sparse Total Least-Squares

CHAPTER 1

INTRODUCTION

Linear inverse problems involve determining unknown variables from a set of linear equations, and have important applications in a wide variety of fields, including medical imaging, geophysics, statistics, and signal processing [1]. The inverse problem formulation is very general and applies to a wide number of measurement models. A case of particular importance is when there are fewer measurements than unknowns (often referred to as the underdetermined case). Such problems are ill-posed [2] in the sense that they do not, in general, have a unique solution. In such cases, additional constraints are required to recover the unknown variables. In standard linear inverse problems, the unknown variables will be represented as a vector $\mathbf{x}$ in $\mathbb{C}^N$, the measurements by a vector $\mathbf{b}$ in $\mathbb{C}^M$, and the linear measurement model as a matrix $\mathbf{A}$ in $\mathbb{C}^{M\times N}$. For the underdetermined case, $\mathbf{A}$ is full row rank and $N \geq M$. Then the inverse problem involves solving $\mathbf{b} = \mathbf{A} * \mathbf{x}$.

In signal processing, many important applications involve signals with parameterized signal models. Some common and well-studied parameterized signals include finite-length signals composed of sums of few sinusoidal components (for example [3]) and source localization using sensor arrays [4]. In the case of sinusoidal signals, the unknown parameters are frequency, amplitude and phase. For source localization, the unknown parameters are the signal energy and direction of arrival. Other examples of parametrized signals arise in problems such as imaging and model fitting.

Given a parameterized signal model, it is often possible to form an inverse problem to recover the unknown signal. For example, given a model $\mathbf{a}(\cdot)$ that maps a parameter $\omega \in \mathbb{R}$ into $\mathbb{C}^M$ and a set of measurements of the unknown signal $\mathbf{b} \in \mathbb{C}^M$, one could discretize the space of possible parameters ω to form a vector $\omega \in \mathbb{R}^N$. Defining $\mathbf{A}(\omega) = \Big[\mathbf{a}(\omega_1), \mathbf{a}(\omega_2), \ldots, \mathbf{a}(\omega_n)\Big]$ as a mapping from $\mathbb{R}^N$ to $\mathbb{C}^{(M\times N)}$, the linear measurement model for a parameterized signal can be written as $\mathbf{b} = \mathbf{A}(\omega) * \mathbf{x}$. For a fixed sampling of the parameter space,

the unknown signal weights $\mathbf{x}$ can be solved as an inverse problem.

As underdetermined (and full row-rank) inverse problems have infinitely many solutions, one must regularize reconstructions using prior assumptions on the known signal $\mathbf{x}$. In recent years, enforcing sparsity in the recovered signal has proven to be a powerful regularization technique. A K-sparse signal $\mathbf{x} \in \mathbb{R}^N$ is one with only K nonzero elements. By enforcing signal sparsity, one can recover, under certain conditions, the unique sparse solution to an undetermined system of linear equations ([5] and [6] among many others). Given a linear measurement model $\mathbf{b} = \mathbf{A} * \mathbf{x}$ and appropriate conditions on $\mathbf{A}$, one can recover a K-sparse $\mathbf{x}$ by solving an ℓ_1 minimization problem known as Basis Pursuit (BP) [7] or using greedy algorithms such as Orthogonal Matching Pursuit [8]. These recovery results form the core of the emerging fields of Compressed Sensing (CS) and sparse recovery.

Compressed Sensing and sparse recovery techniques have been successfully applied to parameterized signal models in a variety of contexts. In prior work [9], the authors applied sparse representation techniques to the problem of source localization with a linear array. By discretizing the direction of arrival θ and performing recovery with Basis Pursuit Denoising (BPDN, a version of the Basis Pursuit problem with inequality constraints), the researchers were able to outperform traditional estimators in terms of resolution and robustness to correlated sources. This idea has been extended to multiple data snapshots and studied in comparison to classic parameter estimation methods [10]. The authors of [11] formulated a wideband array processing formulation parameterized by source location in two-dimensional space. Discretizing this space, the authors demonstrated recovery with simulated data using a block-sparsity penalty. Another interesting application of sparse reconstruction is recovery of narrowband radar signals [12], which are parameterized in range-doppler space by time and frequency shifts. Applying BP, the researchers were able to resolve nearby targets more accurately with fewer measurements than traditional matched filtering.

One of the common problems when applying sparse reconstruction techniques to parameterized signal models is that the dictionary $\mathbf{A}(\omega)$ is a function of the unknown parameters. A major assumption of sparse recovery algorithms is that the signal dictionary $\mathbf{A}$ is exactly known. For sparse signals with unknown parameters, however, the dictionary is not directly given. In this case, the common approach is to sample and discretize the parameter

space to form a signal dictionary. Intuitively, a dense sampling is needed to obtain an accurate reconstruction. This dense discretization, however, can result in poor recovery by BP as the highly correlated elements of the dictionary violate the Restricted Isometry Property [5] ([13] showed this for Discrete Fourier Transforms oversampled in the frequency domain). It is necessary to estimate the unknown parameters to form the signal dictionary.

This thesis considers the joint recovery of both the unknown parameters and the unknown signal weights from a single set of measurements. As many parameterized signal models have few nonzero signal weights compared to the total number of possible parameter values, it is natural to study the application of sparsity constraints to recovering signals with parameterized measurement models. By jointly seeking the unknown signal weights and parameters, one can adapt the dictionary to create sparse solutions. Prior work has investigated CS problems with unknown signal dictionaries, but there are limitations when applying existing techniques to sparse recovery of signals with parameterized measurement models.

1.1 Prior Work on Sparse Recovery with Unknown Signal Dictionaries

Compressive sensing typically requires that the signal dictionary $\mathbf{A}$ is exactly known. In the case of parameterized signal models, the signal dictionary will be a function of the potentially unknown parameters. One common option is to simply discretize the parameter space, as explored, for example, in [9]. As model mismatch may occur if the true signal parameters do not match the discretization, the authors proposed an ad-hoc grid refinement approach to adjust grid elements near detected peaks. Compensating for such model mismatch can be very critical in order to recover the unknown signal. In [14], the authors study the effects of perturbations on the dictionary and found perturbations in the dictionary can result in considerable changes in the recovered signal.

When the signal dictionary $\mathbf{A}$ is completely unknown, the dictionary must be learned from a training dataset. For example, Blind CS [15] provides uniqueness conditions to recover an unknown sparsifying basis given a known sensing model and a set of measurement vectors. Given multiple measure-

ments and a randomized sampling matrix, this techniques can be an effective way to learn a basis for the signal in which it is sparse. In the related field of sparse dictionary learning, many techniques exist to learn an unknown signal dictionary from a set of measurement vectors, for example the K-SVD algorithm [16]. These methods, however, require many measurements and do not exploit any special structure of the unknown dictionary.

Because many applications have at least some prior knowledge on the signal dictionary, further work has sought ways to correct for errors from an initial signal dictionary $\mathbf{A_0}$. Inspired by the method of Total Least-Squares, the approach taken by the Sparse Total Least-Squares (STLS) method ([17], [18]) is to allow for errors in both the measurements and the signal dictionary. The signal dictionary is taken as $\mathbf{A}_0 + \mathbf{dA}$, where $\mathbf{dA}$ is an unknown perturbation of the signal dictionary. A modified BPDN problem can then be solved via an exact branch-and-bound method or an alternating algorithm that approaches a stationary point. This formulation allows for the use of an initial estimate of the signal dictionary and controls the amount of deviation from that dictionary. One of the applications considered by the researchers is direction of arrival estimation, where the initial signal dictionary is formed using a discretization of the direction of arrival parameter space.

In order to enforce knowledge of the parameterized signal model in the sparse recovery problem, it would be desirable to find a structured perturbation from an initial dictionary estimate. Toward this end, further work [19] [20] introduced a Perturbed CS model where the unknown perturbation is a linear function of a set of perturbation parameters. This was again applied to the direction of arrival estimation problem, where the signal dictionary was formed by discretizing the space. The perturbation matrix can be formed using a linearization of the parameterized model with respect to the parameters. The authors then develop recovery guarantees for the perturbed compressed sensing problem, extending the Restricted Isometry Property.

Several other approaches have also been taken toward reconstructing sparse signals with parameterized measurement models. Considering the reconstruction of frequency-sparse signals, recent work [13] has developed iterative thresholding algorithms to recover structured-sparse signals. This method bypasses explicit dictionary formation by leveraging existing parametric estimators for frequency-sparse signals. Although very appealing, this work is only formulated for frequency-sparse signals and the extension to other

applications is not immediately obvious. Taking a Bayesian approach, the authors of [21] develop a Maximum *a Posteriori* (MAP) estimator for estimating parameter perturbations and sparse signals using a sparsity-inducing prior. This results in an iterative recovery algorithm to provide a principled approach to choosing parameters. This Bayesian model is also based on a linear perturbation model.

Current approaches for sparse recovery have several major limitations when applied to parameterized signal models. The common formulation for CS recovery involves a known, finite-dimensional representation of the signal dictionary $\mathbf{A}$. When the unknown signal is a function of a continuous, unknown parameter, it is not immediately obvious how to best form a signal dictionary for sparse recovery. Although the obvious approach is to create a dense discretization of the unknown parameter and seek sparse signal weights, it is not immediately clear if a given discretization will work well or poorly for a given recovery problem.

Current approaches to sparse recovery with parameterized signal models have been somewhat successful but are still limited. Some researchers have developed sparse recovery algorithms for specific parameterized models. Others have attempted to develop solutions that perturb the initial dictionary or parameter estimates. As these approaches rely on specific perturbation models, the error can still depend on the initial sampling. For example, in the case of the linearized perturbation model in [19], the accuracy of the linearized model used depends on the closeness of the true parameter value to the original estimate.

What is clear is that there are several possible approaches to recovering sparse signals with parameterized measurement models. What is less clear is what is the best approach to recovering the unknown signal. The next section presents a simple example to help motivate a potential solution to recover sparse signals with parameterized measurement models.

1.2 Example Recovery of Frequency-Sparse Signals

A simple example of a parameterized signal is a complex sinusoid, which is parameterized by a frequency and a complex amplitude. A frequency-sparse signal can be thought of as the sum of a small number of complex sinusoids.

Given a set of time-domain measurements of such a signal, is it possible to recover the unknown parameters using sparse recovery techniques?

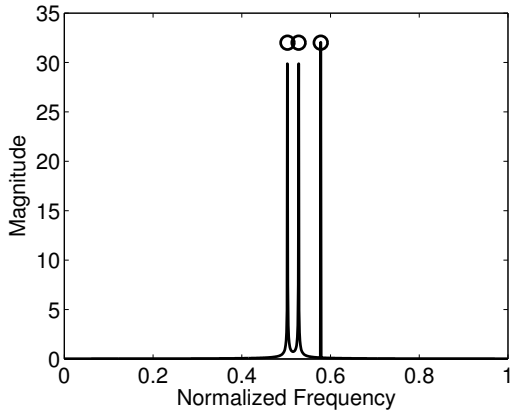
This section first considers a complex-valued, discrete-time signal which is the sum of K sinusoids, $x(n) = \sum_{i=1}^{K}(a_i * \exp(\imath * \omega_i * n))$,where a_i is a complex weight and ω_i is a real-valued frequency parameter. This signal is K-frequency-sparse. Of all possible frequency parameters, only K are used to form the signal. As real-world signals are of finite durations, the time samples are limited to $n = 0, 1, \ldots, M-1$. Due to the well-known time-frequency tradeoff, a finite-length signal will not have a sparse Discrete Fourier Transform (DFT) when the time duration is short. Figure 1.1 demonstrates this well-studied phenomenon for a frequency sparse signal with $K = 3$. When the signal has a long duration the DFT seems to naturally provide a sparse representation of the signal, but at short time durations this is not necessarily true. For this example, it is not even immediately clear from the DFT of the length-32 signal that the signal is composed of three finite-length sinusoidal components. Given this observation, it would seem advantageous to apply sparse recovery techniques to attempt to recover the K unknown frequency and K unknown amplitude parameters.

In order to apply sparse reconstruction techniques to this problem, it is necessary to generate a signal dictionary. A natural way to do so is to define a set of parameters $\omega \in \mathbb{R}^N$ with values bounded between zero and 2π and $N \geq M$. Given a set of noiseless, time-domain measurements $\mathbf{b} \in \mathbb{C}^M$ of the frequency-sparse signal, one can form a sparse recovery problem by defining $\mathbf{A}(\omega) \in \mathbb{C}^{(M \times N)}$ with each column $\mathbf{a}_i$ of $\mathbf{A}$ given by a complex sinusoid of length M and frequency ω_i.

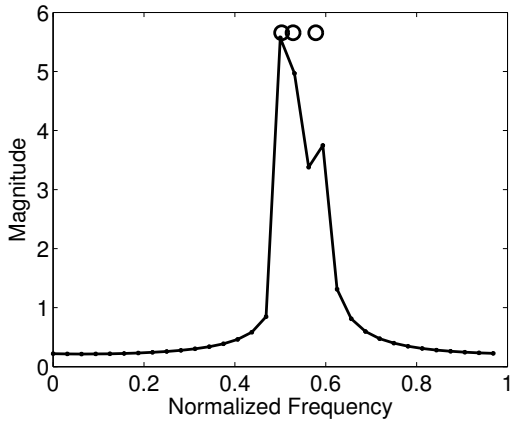
Given a fixed ω, one can then solve $\mathbf{b} = \mathbf{A}(\omega) * \mathbf{x}$ using sparse recovery techniques. The sparse signal $\mathbf{x}$ corresponds to the unknown amplitude parameters and the values of ω at the nonzero values of $\mathbf{x}$ correspond to the unknown frequency parameters. The number of nonzero components in $\mathbf{x}$ corresponds to the recovered model order K.

To quantify the performance of this sparse recovery formulation, consider a length-32 vector of noiseless, complex-valued measurements of a K-frequency-sparse signal with $K = 3$. In this section a sparse $\mathbf{x}$ will be recovered via BP from the model $\mathbf{b} = \mathbf{A}(\omega) * \mathbf{x}$. The final choice remaining is the selection of the discretized set of parameters ω.

An obvious choice is to extend the sampling inherent in the DFT. The



(a) DFT magnitude of length-1024 signal composed of three sinusoids



(b) DFT magnitude of length-32 signal with three sinusoids

Figure 1.1: DFT magnitude of an example signal composed of three sinusoidal components for lengths of 1024 and 32 samples. Circles represent the true magnitude of the sinusoidal components. For the signal of length-1024, the frequency domain representation seems naturally sparse, but the frequency components are no longer clearly separated in a length-32 signal.

DFT can be seen as a frequency-domain sampling of the Discrete-Time Fourier Transform with evenly spaced samples between zero and 2π. For sparse recovery, it is intuitively desirable to have a sample near each unknown frequency, so one would want to oversample compared to the typical DFT frequencies. By defining an oversampling factor k such that $N = k*M$, an evenly spaced sampling is given by $\omega_i = \frac{i*2*\pi}{k*M}$ for $i = 0, 1, \ldots, k*M-1$. Adjusting the oversampling factor allows one to adjust the density of the parameter sampling.

To assess the performance of sparse recovery with evenly sampled parameters, one can compare with the idealized case where the unknown frequencies are in the sampling set. Define ω^* as the sampling where the frequencies of ω closest to the three true frequencies are replaced by the true frequencies. The set of parameters ω^* will be referred to as the oracle parameters. One can then form $\mathbf{A}(\omega)$ and $\mathbf{A}(\omega^*)$ to compare the recoveries.

Figure 1.2 shows an example recovery for a five-times oversampled ω and the corresponding ω^* of an example frequency-sparse signal with $K = 3$. As expected for such a low oversampling, there is significant error in the recovered signal, but it still appears promising. Using ω^*, the recovery is exact. This suggests that highly oversampling the dictionary may not necessarily achieve the most accurate reconstructions.

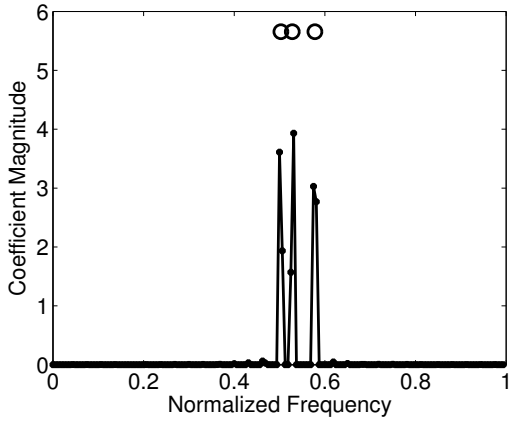
In order to explore the effect of parameter oversampling on this sparse recovery problem, one can examine error in recovered coefficients and number of nonzero components in the sparse recovery as a function of oversampling factor. Ideally, as the oversampling factor increases, coefficient estimation error should approach zero and the number of nonzero components should approach K.

A set of 30 example frequency-sparse signals with $K = 4$ and $M = 32$ were generated with identical amplitudes and random frequency parameters. If the frequency parameters were closer than $2\pi/M$ they were regenerated. These signals were then recovered using BP with evenly spaced parameters and the oracle parameters. Figure 1.3 shows the coefficient recovery error and number of nonzero coefficients as a function of oversampling. The error in the estimated coefficients does not approach zero as oversampling increases. The number of nonzero elements does decrease with oversampling factor, but even at high oversampling factors two elements are required to approximate each unknown. Recovery with the oracle, however, is exact and does not depend drastically on the oversampling factor.

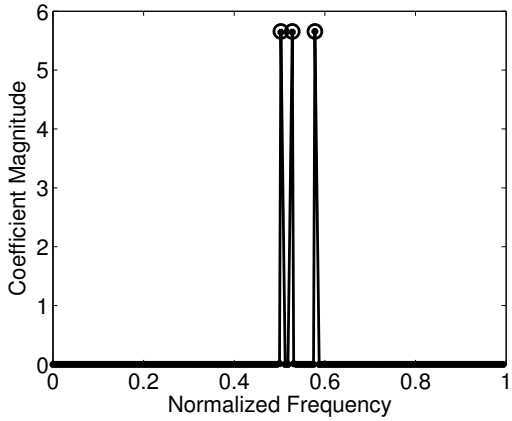
The final quantity shown in Figure 1.3 is the coherence of the matrix $\mathbf{A}(\omega)$ as a function of the oversampling factor. The mutual coherence of a matrix $\mathbf{A}$ is defined as

$$\max_{1 \leq i \leq N, 1 \leq j \leq N, i \neq j} \frac{|\langle \mathbf{a}_i, \mathbf{a}_j \rangle|}{\langle \mathbf{a}_i, \mathbf{a}_i \rangle} \tag{1.1}$$

The mutual coherence rapidly increases as a function of the oversampling factor. This implies that the columns of the dictionary become increasingly



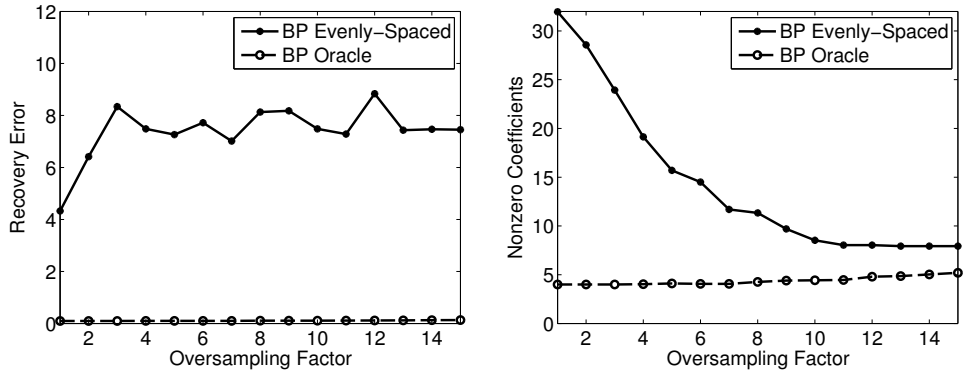
(a) Sparse reconstruction with BP, noiseless measurements, and five-times oversampled frequency parameters.



(b) Sparse reconstruction with BP, noiseless measurements, five-times oversampled frequency parameters. The nearest sampled frequencies are replaced by the true signal frequencies.

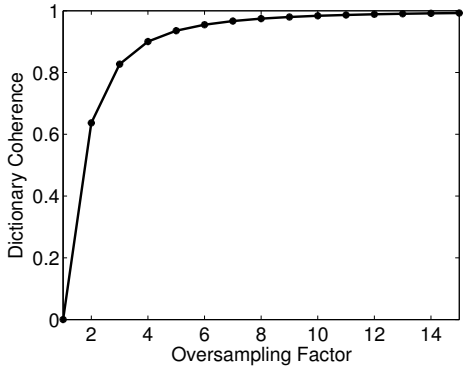
Figure 1.2: Sparse recovery of a length-32 signal composed of three complex sinusoids with noiseless measurements. Recovery using a five-times oversampled DFT frame is reasonable, with recovered parameters near the unknown signals. The coefficient estimates, however, show considerable error. When the true frequency parameters are part of the parameter sampling the recovery is exact.

correlated, which is expected of the oversampled DFT. From a sparse recovery perspective, this is discouraging, as the recovery guarantees developed by the sparse reconstruction community only hold for signal dictionaries with sufficiently low mutual coherence. This implies that although sparse recovery may be possible with highly oversampled dictionaries, it will certainly never



(a) Error in recovered coefficients as a function of oversampling

(b) Number of nonzero coefficients recovered as a function of oversampling



(c) Dictionary coherenece as a function of oversampling

Figure 1.3: Sparse recovery, using BP, of noiseless signals of length-32 composed of four sinusoids averaged over 30 example signals. Coefficient estimation error does not uniformly decrease with oversampling for evenly spaced reconstructions. The number of nonzero components does fall with oversampling, but eventually levels off. For the oracle reconstructions, the true frequencies replace the nearest evenly spaced frequency sample. The oracle reconstruction is able to recover the signal coefficients exactly regardless of oversampling factor. Note that the dictionary coherence, related to the recovery guarantees for sparse signals, increases rapidly with oversampling factor.

be guaranteed.

Simply oversampling the parameter space does not eliminate recovery error, even in this highly simplified case. Introducing additional complications such as noise will only decrease recovery performance. This simple example demonstrates some of the limitations of oversampling the parameter space, but it also demonstrates that sparse recovery can be a powerful tool. The oracle reconstructions are able to recover the unknown parameters with little

error.

This observation leads to a new way to recover sparse signals with parameterized measurement models. The oracle and evenly sampled parameter sets differ only at the K frequencies closest to the true frequencies. At all other locations they are identical. If one could jointly recover a perturbation in the parameters forming the measurement model while recovering a sparse signal, it may be possible to exactly recover the unknown parameters. In fact, since only K parameters should be perturbed, this perturbation term should also be sparse. This observation suggests a possible sparse recovery technique for signals with parameterized measurement models, where one recovers both the sparse signal and a sparse set of parameter perturbations from an initial parameter estimate.

1.3 Proposed Solution

This thesis proposes a new sparse recovery technique for parameterized signal models involving the joint recovery of a sparse signal and a sparse parameter perturbation given a parameterized model, a measurement vector, and an initial set of parameters. The goal is to exploit the parameterized model to recover the unknown signal weights and parameters using sparse recovery techniques.

Given a set of measurements $\mathbf{b} \in \mathbb{C}^M$, a differentiable, parameterized signal model $\mathbf{A}(\omega)$, and an initial set of parameter estimates $\omega \in \mathbb{R}^N$ (with $N \geq M$), this thesis proposes a new Perturbed Sparse Recovery (PSR) problem to solve for the unknown, sparse signal weights $\mathbf{x}$ and sparse parameter perturbations $\mathbf{d}\omega$. This problem can be formulated mathematically as

$$\begin{aligned} \min_{\mathbf{x},\mathbf{d}\omega} \quad & \|\mathbf{x}\|_1 + \lambda * \|\mathbf{d}\omega\|_1 \\ \text{subject to} \quad & \|\mathbf{b} - \mathbf{A}(\omega + \mathbf{d}\omega) * \mathbf{x}\|_2 \leq \epsilon \end{aligned} \tag{1.2}$$

where $\lambda \in \mathbb{R}$ and $\epsilon \in \mathbb{R}$ are parameters to control the size of the parameter perturbation and deviations from the measurements, respectively. The final parameter estimates are taken to be $\omega^* = \omega + \mathbf{d}\omega$. Interestingly, this nonlinear programming problem can be solved as a series of linearized subproblems using convex programming techniques.

This formulation directly uses the parameterized signal model and has the potential to recover both the unknown signal weights and parameters. By allowing for parameter perturbations, this model will hopefully be less sensitive to the initial sampling of the parameter space. This formulation attempts to correct for modeling error present when the parameters are unknown by recovering the parameter perturbations.

This recovery problem is applicable a wide range of signals with parameterized measurement models, provided that the recovered solution should be sparse. The major requirement is that the elements of the parameterized signal model must be differentiable with respect to the unknown parameters. Many problems in imaging, array processing, and estimation fit the proposed model.

1.4 Comparison to Existing Sparse Recovery Formulations

The proposed formulation is related to existing sparse recovery techniques, and can be seen as an important generalization of existing methods. The BP method seeks to find the minimum ℓ_1 norm $\mathbf{x}$ solution to the linear model $\mathbf{b} = \mathbf{A} * \mathbf{x}$. This can be extended to a measurement model with noise, $\mathbf{b} = \mathbf{A} * \mathbf{x} + \mathbf{e}$ for $\mathbf{e} \in \mathbb{R}^m$ and $\|\mathbf{e}\|_2 < \epsilon$. One then solves the Basis Pursuit Denoising (BPDN) problem.

$$\begin{aligned} \min_{\mathbf{x}} \quad & \|\mathbf{x}\|_1 \\ \text{subject to} \quad & \|\mathbf{A} * \mathbf{x} - \mathbf{b}\|_2 \leq \epsilon \end{aligned} \tag{1.3}$$

The proposed recovery problem is similar to the BPDN problem, with the major addition of a parameterized model for the matrix $\mathbf{A}$. If the parameters of the sensing matrix are known in advance, one can simply generate a fixed $\mathbf{A}$ using the parameterized model and solve the appropriate BPDN problem. The PSR formulation recovers both the unknown signal weights $\mathbf{x}$ and the underlying parameters in order to estimate the unknown sparse signal.

The PSR approach differs from previous methods to deal with unknown sensing matrices. Unlike the dictionary learning problem, this problem is formulated for a single measurement vector using a parameterized signal model.

Although the Bayesian approach in [21] was formulated for recovering unknown parameter perturbations, the authors consider a uniform prior on the parameters with a linearized measurement model. The iterative threshold solver proposed in [13] also attempts to recover unknown signal weights and parameters, but takes a very different approach. They formulate a structured-sparse recovery algorithm that iteratively reconstructs the signal using spectral-estimation tools. This approach means that this model is limited only to recovering frequency-sparse signals.

The most closely related methods are the STLS formulation and the Perturbed CS formulation. The STLS formulation aims to recover a sparse signal $\mathbf{x}$ using an ℓ_1 relaxation subject to a perturbed model $(\mathbf{A}+\mathbf{dA})*\mathbf{x} = \mathbf{b}$. The size of the general perturbation term, $\mathbf{dA}$, is controlled by weighting the ℓ_1 norm of the signal $\mathbf{x}$ against the Frobenius norm of the perturbation. This formulation does not explicitly take advantage of any structure or parameterization in the signal dictionary, but can still be used to compensate for model mismatch when there is a parameterized signal model. The proposed PSR problem may be more appropriate when a signal model is known and is relatively easy to compute, as this will enforce more information about the signal.

The work developing Perturbed CS considers a parameterized signal model, but only with a linear perturbation model. This model takes the form $\mathbf{A}(\omega + \mathbf{d}\omega) = \mathbf{A_0} + \mathbf{B} * \Delta$ with $\mathbf{A_0}$ and $\mathbf{B}$ given (Δ is the diagonal matrix with elements given by the perturbations $\mathbf{d}\omega$). In this prior work, the parameterized model is strictly linear in the sense that $\mathbf{a}_i = \mathbf{a}_{i,0} + \mathbf{b}_i * d\omega_i$. The parameters are constrained by the ℓ_∞ norm to control the maximum parameter perturbation. To solve this, an alternating algorithm dubbed the Alternating Algorithm Perturbed Basis Pursuit Denoising (AA-P-BPDN) was developed. The recovery algorithm is only valid for the linear perturbation model.

Of existing techniques, recovery via STLS and Perturbed CS using the AA-P-BPDN technique are most closely related to the proposed PSR problem. All three approaches are based around convex relaxations of the sparse approximation problem using the ℓ_1 norm while allowing for perturbations in the signal dictionary from an initial estimate. To the best of the author's knowledge, this represents the current state of the art in this area. The STLS, AA-P-BPDN, and standard BPDN will form the most relevant comparisons

to the solution proposed in this thesis.

1.5 Contributions of This Thesis

The key contribution of this thesis is the PSR problem formulation and a solution algorithm, Successive Linearized Programming for Sparse Recovery, to find first-order critical points of the PSR problem. This formulation allows for joint recovery of sparse perturbation parameters and sparse signals from parameterized signal models. Due to the generality of the problem formulation, it is possible to apply this technique to a wide variety of problems in array processing, spectral estimation, and imaging.

The formulation for parameter perturbation recovery is more general than the linear structure in [19], and contains the linear structure as a subset. The proposed algorithm is, to the best of our knowledge, novel and is applicable to a far wider range of models than the alternating algorithm in [19]. Unlike the spectral-estimation algorithm in [13], SLPSR is applicable to a wide range of parameterized measurement models.

This new approach is applied to simulations of frequency-sparse signals and narrow-band source localization with linear arrays. The PSR approach is able to recover the unknown coefficient and parameters of frequency-sparse signals almost exactly in the noiseless case, outperforming other state of the art sparse recovery approaches. Interestingly, the PSR approach exhibits the best performance at low oversampling factors, suggesting that this approach does not require highly oversampled signal dictionaries. With additive noise, the PSR approach performs well compared to existing methods.

In order to explore the potential application of sparse recovery techniques to real-world signals with parameterized representations, the proposed technique is applied to the reconstruction of Radio Frequency(RF) data consisting of transmitters broadcasting continuous-wave Morse code. This signal is frequency-sparse over blocks. The proposed sparse recovery technique can be used to form sparse representations of these signals while estimating transmitter frequencies without using highly oversampled signal dictionaries.

The details of the problem formulation and proposed recovery algorithm are developed in Chapter 2. After developing the necessary recovery algorithm, the PSR formulation is applied to simulated signals in Chapter 3.

Finally, the proposed method is applied to the recovery of frequency-sparse RF data in Chapter 4.

CHAPTER 2

PROPOSED PARAMETERIZED SPARSE RECOVERY FORMULATION AND ALGORITHM

In order to recover sparse representations of signals with parameterized measurement models, one must first define a sparse recovery problem. In this chapter, a parameterized signal model is introduced, along with a corresponding sparse recovery problem. The proposed problem, unfortunately, has nonlinear constraint equations that prevent the use of current solution approaches. This chapter develops a novel, iterative algorithm, Successive Linearized Programming for Sparse Reconstruction (SLPSR), to find first-order critical points of the sparse recovery problem for parameterized models. This algorithm linearizes the nonlinear constraint set at the current iterate and solves the linearized subproblem using Second-Order Conic Programming (SOCP) or subgradient descent. This chapter then develops the first-order necessary conditions for this problem to provide a stopping criterion for the algorithm. It is not immediately obvious that this iterative algorithm converges, but this chapter provides a proof of convergence to a first-order critical point. Finally, the proposed problem is formulated as a Maximum *a Posteriori* (MAP) estimation problem with Laplacian priors on the parameter perturbations and signal coefficients. This MAP formulation allows this problem to be interpreted as an estimation problem, and demonstrates how to take a Bayesian approach to parameter tuning and introduction of additional constraints. Although the MAP formulation is not directly exploited in this work, it may provide an important tool for future work on sparse signal and parameter recovery.

To begin, a mathematical model is required for sparse signals with parameterized measurement models. Consider a set of signal coefficients, $\mathbf{y} \in \mathbb{C}^R$, and signal parameters, $\beta \in \mathbb{R}^R$. These parameters and coefficients are related to a set of measurements $\mathbf{b} \in \mathbb{C}^M$. The number of weights and parameters, R, is less than the number of measurements M. A key assumption is that the measurements are related to the unknown weights and parameters by the

model $\mathbf{b} = \sum_{i=1}^{R} \mathbf{a}(\beta_i) * y_i$ where $\mathbf{a}(\beta_i)$ is a function mapping a parameter β to $\mathbb{C}^M$. The function $\mathbf{a}(\bullet)$ essentially generates a vector, or dictionary element, corresponding to a given parameter value.

The goal of signal recovery is to reconstruct the unknown signal coefficients, signal parameters, and model order given only the measurements $\mathbf{b}$. Unfortunately, there is no guarantee of a unique solution to this problem without additional constraints. Typically, there are infinitely many combinations of $\mathbf{y}$, β and R that can describe a typical set of measurements. However, the model order is assumed to be small. It is then reasonable to search for a sparse solution—a solution where R is very small and there are few elements in $\mathbf{y}$ and β.

To formulate a sparse recovery problem, one needs to form a finite-dimensional optimization problem that can be executed on a digital computer. Given a set of initial parameter values, $\omega \in \mathbb{R}^N$, with $N \geq M$, one can form a mapping $\mathbf{A}(\omega)$ from $\mathbb{R}^N$ to $\mathbb{C}^{M \times N}$ where $\mathbf{A}(\omega)$ is given by

$$\mathbf{A}(\omega) = \Big[\mathbf{a}(\omega_1), \mathbf{a}(\omega_2), \cdots, \mathbf{a}(\omega_N)\Big] \tag{2.1}$$

For the proposed recovery algorithm, it is critical that the elements of $\mathbf{a}(\omega_i)$ are differentiable with respect to the parameter ω_i.

The final component to be considered in the sparse signal model is measurement noise. The measurements are assumed to be corrupted by an additive noise term. The unknown weights and parameters $\mathbf{y} \in \mathbb{C}^R$ and $\beta \in \mathbb{R}^R$ are related by the linear model $\mathbf{z} = \sum_{i=1}^{R}(\mathbf{a}(\beta_i) * y_i)$. Then the measurements are given by $\mathbf{b} = \mathbf{z} + \nu$, where ν is an additive noise term with $\|\nu\| < \epsilon$.

To recover the unknown weights $\mathbf{y}$ and the parameters β from the noisy measurements $\mathbf{b}$, this thesis proposes solving a nonlinear programming problem given an initial set of parameter values $\omega \in \mathbb{R}^N$ to recover a set of parameter perturbations $\mathbf{d}\omega \in \mathbb{R}^N$ and weights $\mathbf{x} \in \mathbb{C}^N$. Again, the elements of the model $\mathbf{a}(\omega_i)$ are assumed to be differentiable with respect to ω_i. In order to jointly recover a sparse vector of weights and a sparse vector of parameter perturbations, this work proposes the following problem, referred to

as Perturbed Sparse Recovery.

$$\begin{aligned} \min_{\mathbf{x},\mathbf{d\omega}} \quad & \|\mathbf{x}\|_1 + \lambda * \|\mathbf{d}\omega\|_1 \\ \text{subject to} \quad & \|\mathbf{b} - \mathbf{A}(\omega + \mathbf{d}\omega) * \mathbf{x}\|_2 \leq \epsilon \end{aligned} \tag{2.2}$$

In this formulation, the parameter $\lambda \in \mathbb{R}$ weights the tradeoff between sparsity of the recovered weights and the size of the perturbations. The parameter $\epsilon \in \mathbb{R}$ allows for deviations from the measurements to compensate for noise. The ℓ_1 norm is used to promote sparsity in the final solution as in [7]. Given a solution to the PSR problem, $\mathbf{x}^*$ and $\mathbf{d}\omega^*$, one can take the estimated model order, R^*, to be given by the cardinality of the support of $\mathbf{x}^*$. The solution is then related to the parameterized model by $\mathbf{y}^* = \mathbf{x}^*_{supp(\mathbf{x}^*)}$ and $\beta^* = (\omega^* + \mathbf{d}\omega^*)_{supp(\mathbf{x}^*)}$. Solving the PSR problem can potentially recover the unknown signal parameters and coefficients. However, the PSR problem is nonlinear as the constraint $\|\mathbf{b} - \mathbf{A}(\omega + \mathbf{d}\omega) * \mathbf{x}\|_2 \leq \epsilon$ can have an arbitrary dependence on $\mathbf{d}\omega$. As traditional sparse recovery algorithms are formulated for the linear model $\mathbf{A} * \mathbf{x} = \mathbf{b}$, current approaches are not applicable. In order to solve the PSR problem, a new sparse recovery algorithm is required.

2.1 Sparse Recovery Algorithm

The goal of this section is to develop a recovery algorithm for the sparse vector $\mathbf{x} \in \mathbb{C}^N$ and parameter perturbations $\mathbf{d}\omega \in \mathbb{R}^N$. What is given is a set of initial parameters $\omega \in \mathbb{R}^N$, a model $\mathbf{a}(\omega_i)$ that maps a parameter value to a vector in $\mathbb{C}^N$, and a set of measurements $\mathbf{b} \in \mathbb{C}^M$.

The algorithm developed in this section is capable of finding first-order critical points of the PSR problem by solving linearized subproblems. The algorithm, Successive Linearized Programming for Sparse Recovery (SLPSR), takes an initial, feasible point $(\mathbf{x}_0, \mathbf{d}\omega_0)$ and generates a series of feasible iterates $(\mathbf{x}_k, \mathbf{d}\omega_k)$. The initial feasible solution is found by solving a standard BPDN problem with $\mathbf{d}\omega_0$ set to zero. Iterates are generated as the solutions to linearized versions of the PSR problem. A step $(\mathbf{s}_\mathbf{x}, \mathbf{s}_{\mathbf{d}\omega})$ from the current iterate is found by solving the linearized problem. As the linearization only holds around the point $(\mathbf{x}_k, \mathbf{d}\omega_k)$, the step is restricted to a Trust-Region [22] around the point $(\mathbf{x}_k, \mathbf{d}\omega_k)$. The linearized subproblem, $MinSubproblem$, is

defined as:

$$\begin{aligned} \underset{\mathbf{s}_x, \mathbf{s}_{d\omega}}{\arg\min} \quad & \|\mathbf{x}_k + \mathbf{s}_x\|_1 + \lambda * \|\mathbf{d}\omega_k + \mathbf{s}_{\mathbf{d}\omega}\|_1 \\ \text{subject to} \quad & \|\mathbf{b} - \mathbf{A}(\omega + \mathbf{d}\omega_k) * \mathbf{x}_k + \mathbf{J} * (\begin{bmatrix} \mathbf{s}_\mathbf{x} \\ \mathbf{s}_{\mathbf{d}\omega} \end{bmatrix})\|_2 \leq \epsilon \\ & \|\mathbf{s}_\mathbf{x}\|_2 + \|\mathbf{s}_{\mathbf{d}\omega}\|_2 \leq \Delta_k \end{aligned} \tag{2.3}$$

The parameters λ and ϵ are the same as in the original problem formulation. Here $\mathbf{J}$ is the Jacobian of the function $\mathbf{b} - \mathbf{A}(\omega + \mathbf{d}\omega_k) * \mathbf{x}_k$ at $(\mathbf{x}_k, \mathbf{d}\omega_k)$. The Jacobian exists as the function $\mathbf{a}(\omega_i)$ is assumed to be differentiable with respect to the parameters. The trust-region is enforced by limiting the norm of the step $(\mathbf{s}_\mathbf{x}, \mathbf{s}_{\mathbf{d}\omega})$ with the parameter Δ_k.

It is important to note that the linearized subproblem is a convex objective function subject to a convex constraint. A minimizer therefore exists and can be found using convex programming techniques.

After calculating the step for the linearized subproblem, the new point $(\mathbf{x}_k + \mathbf{s}_x, \mathbf{d}\omega_k + \mathbf{s}_{\mathbf{d}\omega})$ is only a tentative solution as it may not be feasible. A correction step is calculated by fixing $\mathbf{d}\omega + \mathbf{s}_{\mathbf{d}\omega}$ and calculating the minimum change in $\delta_\mathbf{x}$ to produce a feasible solution. The function $FeasibleProjection$ is defined as the minimum norm $\delta_\mathbf{x}$ that satisfies $\|\mathbf{b} - \mathbf{A}(\omega + \mathbf{d}\omega_k + \mathbf{s}_{\mathbf{d}\omega}) * (\mathbf{x}_k + \mathbf{s}_x + \delta_\mathbf{x})\|_2 \leq \epsilon$. Given this new point $(\mathbf{x}_k + \mathbf{s}_x + \delta_\mathbf{x}, \mathbf{d}\omega_k + \mathbf{s}_{\mathbf{d}\omega})$, the new objective function value, $\|\mathbf{x}_k + \mathbf{s}_x + \delta_\mathbf{x}\|_1 + \lambda * \|\mathbf{d}\omega_k + \mathbf{s}_{\mathbf{d}\omega}\|_1$, is calculated. If this function value is an improvement over the previous iterate, the new point is accepted and the trust region radius, Δ_{k+1}, increased. If it is not an improvement, the algorithm remains at $(\mathbf{x}_k, \mathbf{d}\omega_k)$ and decreases the trust region radius Δ_k. This continues until the first-order necessary conditions are satisfied or a maximum number of iterations are reached. The first-order necessary conditions are discussed in Section 2.2.

In order to solve the linearized subproblems, either SOCP or subgradient descent can be applied. Solving for the feasible projection is simply the projection onto a convex set, and can be solved using standard quadratic programming techniques [23]. Solution methods for the subproblems are discussed in Section 2.3. After discussing the solution of the subproblems, the convergence of this algorithm from an initial estimate to a first-order critical point is proven.

Algorithm 1 Successive Linearized Programming for Sparse Recovery (SLPSR)

Require: $0 < \eta < 1$, $0 < \epsilon$, $0 < \lambda$,ω,$\mathbf{x}_0$,$\mathbf{d}\omega_0$

$\mathbf{x} \leftarrow \mathbf{x}_0$
$\mathbf{d}\omega \leftarrow \mathbf{d}\omega_0$
while $\mathbf{x}, \mathbf{d}\omega$ is not First-Order Critical **do**
    $\mathbf{J}_k \leftarrow \mathbf{J}(\mathbf{x}, \mathbf{d}\omega)$
    $\mathbf{A}_k \leftarrow \mathbf{A}(\omega + \mathbf{d}\omega)$
    $\mathbf{s_x}, \mathbf{s_{d\omega}} \leftarrow MinSubproblem(\mathbf{x}, \mathbf{d}\omega, \omega, \lambda, \epsilon, \mathbf{J_k}, \mathbf{A_k})$
    $\delta_\mathbf{x} \leftarrow FeasibleProjection(\mathbf{x}, \mathbf{d}\omega, \omega, \mathbf{s_x}, \mathbf{s_{d\omega}}, \epsilon, \mathbf{b})$
    if $\|\mathbf{x} + \mathbf{s_x} + \delta_\mathbf{x}\|_1 + \lambda * \|\mathbf{d}\omega + \mathbf{s_{d\omega}}\|_1 < \|\mathbf{x}\|_1 + \lambda * \|\mathbf{d}\omega\|_1$ **then**
        $\mathbf{x} \leftarrow \mathbf{x} + \mathbf{s_x} + \delta_\mathbf{x}$
        $\mathbf{d}\omega \leftarrow \mathbf{d}\omega + \mathbf{s_{d\omega}}$
        $\Delta \leftarrow \frac{1}{\eta} * \Delta$
    else
        $\mathbf{x} \leftarrow \mathbf{x}$
        $\mathbf{d}\omega \leftarrow \mathbf{d}\omega$
        $\Delta \leftarrow \eta * \Delta$
    end if
end while

2.2 First-Order Necessary Conditions for Optimality

To provide a stopping criterion and prove convergence of the SLPSR algorithm, one must be able to characterize first-order critical points of the PSR problem. For differentiable problems, first-order necessary conditions for nonlinear programming problems can be found from the well-known Karush-Kuhn-Tucker conditions [23]. The ℓ_1 norm, however, is not differentiable at any point where an element equals zero. It is still possible, however, to characterize first-order critical points of continuous, convex but nonsmooth functions using the subdifferential [24] of the objective function, $\partial f(\mathbf{x}, \mathbf{d}\omega)$.

The subdifferential is an extension of the standard gradient definition. For a convex, real-valued, nonsmooth function, the subdifferential is the set of subgradients. Formally, a subgradient is defined as:

Definition 1. For a continuous function $f(\mathbf{x}) : \mathbb{R}^N \to \mathbb{R}$ at the point $\mathbf{x} \in \mathbb{R}^N$, a vector $\mathbf{d} \in \mathbb{R}^N$ is a subgradient of f at $\mathbf{x}$ if

$$f(\mathbf{y}) \geq f(\mathbf{x}) + \mathbf{d}^T * (\mathbf{y} - \mathbf{x}), \forall \mathbf{y} \in \mathbb{R}^N \tag{2.4}$$

If a function f is differentiable, then the subgradient is unique and equal to

the gradient. If $f(\mathbf{x})$ is convex, then one can define the set of subgradients of f at $\mathbf{x}$ as $\partial f(\mathbf{x})$. For such an f, $\partial f(\mathbf{x})$ is a non-empty, convex, and compact set. Intuitively, this set defines all linear functions that lie below the convex function f. The notion of a subdifferential can then be used to extend the definition of first-order necessary conditions for smooth functions (using the gradient) to nonsmooth functions.

For smooth functions, a necessary condition for a point $\mathbf{x}$ to be a first-order critical is that the gradient of $f(\mathbf{x})$ is zero. The equivalent condition for nonsmooth functions is that the zero element exists in the subdifferential of f at $\mathbf{x}$. The PSR, problem, however, is a constrained optimization problem. The first-order necessary conditions for nonsmooth functions constrained by smooth constraints can be found via the Lagrange multiplier technique [25]. At a feasible point $(\mathbf{x}, \mathbf{d}\omega)$, The first-order critical conditions are $0 \in \partial f(\mathbf{x}, \mathbf{d}\omega) + \mathbf{J}(\mathbf{x}, \mathbf{d}\omega)^T * (\mathbf{b} - \mathbf{A}(\omega + \mathbf{d}\omega) * \mathbf{x}) * \gamma$ for some Lagrange multiplier $\gamma \geq 0$.

Now, one must develop the first-order necessary conditions for the proposed PSR problem. To begin, consider real-valued signals, measurements, and measurement models. A slightly different formulation will be required for complex-valued measurements. Considering the objective function of the PSR formulation, $\|\mathbf{x}\|_1 + \lambda * \|\mathbf{d}\omega\|_1$ with $\mathbf{x} \in \mathbb{R}^N$ and $\mathbf{d}\omega \in \mathbb{R}^N$, the objective function is nondifferentiable for any $x_i = 0$ and any $d\omega_i = 0$.

One can then characterize the subdifferential, $\partial f(\mathbf{x}, \mathbf{d}\omega)$ as

$$\partial f(\mathbf{x}, \mathbf{d}\omega) = \left\{ \begin{array}{c} \begin{bmatrix} \partial \mathbf{x} \\ \partial \mathbf{d}\omega \end{bmatrix} : \\ \partial x_i = \begin{cases} sign(x_i) & \text{if } |x_i| > 0 \\ \Big[-1, 1\Big] & \text{if } |x_i| = 0 \end{cases}, \\ \partial d\omega_i = \begin{cases} \lambda * sign(d\omega_i) & \text{if } |d\omega_i| > 0 \\ \Big[-\lambda, \lambda\Big] & \text{if } |d\omega_i| = 0 \end{cases} \end{array} \right\} \tag{2.5}$$

For a complex-valued $\mathbf{x}$,$\mathbf{b}$ and $\mathbf{A}(\omega)$ (but with real-valued ω), the story is somewhat different. In general, complex derivatives are quite different than gradients of real-valued functions. However, because the PSR problem only involves the optimization of a real-valued function (the ℓ_1 norm) with respect to the complex variable $\mathbf{x}$, one can exploit the well-known $\mathbb{CR}$ calculus [26] by

writing the objective function in terms of the real and complex parts of the complex variables. This effectively transforms the optimization problem into the optimization of a real-valued function with $2 * N$ real-valued variables. The measurements and the constraints must be rewritten

$$\mathbf{b_c} = \begin{bmatrix} \text{real}(\mathbf{b}) \\ \text{imag}(\mathbf{b}) \end{bmatrix} \tag{2.6}$$

$$\mathbf{A}_c(\omega) = \begin{bmatrix} \text{real}(\mathbf{A}(\omega)), -\text{imag}(\mathbf{A}(\omega)) \\ \text{imag}(\mathbf{A}(\omega)), \text{real}(\mathbf{A}(\omega)) \end{bmatrix} \tag{2.7}$$

If $\mathbf{x}$ is similarly rewritten

$$\mathbf{x_c} = \begin{bmatrix} \text{real}(\mathbf{x}) \\ \text{imag}(\mathbf{x}) \end{bmatrix} \tag{2.8}$$

then the constraint of the PSR problem for complex-valued vector $\mathbf{x}$ can be written as $\|\mathbf{b_c} - \mathbf{A}_c(\omega) * \mathbf{x}_c\|_2 \leq \epsilon$. This constraint is now also a real-valued inequality that is a function of the real-valued parameters ω, real($\mathbf{x}$), and imag($\mathbf{x}$). $\mathbf{J_c}$ is defined as the Jacobian of this function with respect to ω, real($\mathbf{x}$), and imag($\mathbf{x}$).

Now it is possible to characterize the subdifferential of the PSR problem in terms of the real and imaginary components, $\mathbf{x} = \mathbf{x}_r + \imath * \mathbf{x}_i$. The real-valued function $\|\mathbf{x}\|_1 + \lambda * \|\mathbf{d}\omega\|_1 = \sum_{j=1}^{N} \sqrt{x_{r,j}^2 + x_{i,j}^2} + \lambda * \|\mathbf{d}\omega\|_1$ has a subdifferential with respect to the real and imaginary components of $\mathbf{x}$ and the real-valued parameter perturbations $\mathbf{d}\omega$.

$$\partial f(\mathbf{x}_c, \mathbf{d}\omega) = \left\{ \begin{bmatrix} \partial \mathbf{x_r} \\ \partial \mathbf{x_i} \\ \partial \mathbf{d}\omega \end{bmatrix} : \begin{array}{c} \partial x_{r,j} = \begin{cases} \frac{x_{r,j}}{|x_j|} & \text{if } |x_j| > 0 \\ \left[-\sqrt{1 - \partial x_{i,j}^2}, \sqrt{1 - \partial x_{i,j}^2}\right] & \text{if } |x_j| = 0 \end{cases}, \\ \partial x_{i,j} = \begin{cases} \frac{x_{i,j}}{|x_j|} & \text{if } |x_j| > 0 \\ \left[-\sqrt{1 - \partial x_{r,j}^2}, \sqrt{1 - \partial x_{r,j}^2}\right] & \text{if } |x_j| = 0 \end{cases}, \\ \partial d\omega_j = \begin{cases} \lambda * sign(d\omega_j) & \text{if } |d\omega_j| > 0 \\ \left[-\lambda, \lambda\right] & \text{if } |d\omega_j| = 0 \end{cases} \end{array} \right\} \tag{2.9}$$

For real-valued $\mathbf{x}$, the first-order necessary conditions for the PSR problem for a feasible point $(\mathbf{x}, \mathbf{d}\omega)$ are given by $0 \in \partial f(\mathbf{x}, \mathbf{d}\omega) + \mathbf{J}(\mathbf{x}, \mathbf{d}\omega)^T * (\mathbf{b} - \mathbf{A}(\omega + \mathbf{d}\omega) * \mathbf{x}) * \gamma$ for some $\gamma \geq 0$. Similarly, for complex-valued $\mathbf{x}$, the conditions are given by $0 \in \partial f(\mathbf{x}_c, \mathbf{d}\omega) + \mathbf{J}_c(\mathbf{x}_c, \mathbf{d}\omega)^T * (\mathbf{b}_c - \mathbf{A}_c(\omega + \mathbf{d}\omega) * \mathbf{x}_c) * \gamma$ for some $\gamma \geq 0$.

It is important to note that these are equivalent to the first-order necessary conditions for the linearized subproblem with $\mathbf{s} = 0$. The stopping criterion for the SLPSR algorithm provided by these first-order necessary conditions corresponds to a first-order critical point of the linearized subproblem.

The simple form of the subdifferential for the PSR problem is fortuitous, as it is possible to actually directly check the first-order necessary conditions. This involves solving a constrained least-squares problem.

To formulate this least-squares problem for a real-valued $\mathbf{x}$, one notes that the subdifferential is fixed for nonzero elements and is allowed to vary between negative one and one for zero-valued elements. Define $I_{\mathbf{x}}$ as the set of nonzero elements of $\mathbf{x}$ and $I_{\mathbf{d}\omega}$ as the set of nonzero elements of $\mathbf{d}\omega$. Similarly, $I_{\mathbf{x}}^c$ and $I_{\mathbf{d}\omega}^c$ denote the corresponding sets of zero-valued elements. Define a vector $g \in \mathbb{R}^{N+N}$, where $g_{I_{\mathbf{x}}} = sign(\mathbf{x}_{I_{\mathbf{x}}})$, $g_{I_{\mathbf{d}\omega}} = sign(\mathbf{d}\omega_{I_{\mathbf{d}\omega}})$, and the other elements of g are set to zero. Next, one must define a matrix $\Phi \in \mathbb{R}^{(N+N)\times(|I_{\mathbf{x}}{}^c|+|I_{\mathbf{d}\omega}{}^c|+1)}$. There is one column in Φ for each zero-valued elements of $\mathbf{x}$ and $\mathbf{d}\omega$. For each zero-valued element in $\mathbf{x}$, Φ is equal to one at the corresponding column and row index. Similarly, for each zero-valued element in $\mathbf{d}\omega$, Φ is equal to λ at the corresponding column and row index. The elements of Φ are zero everywhere else, except the final column. This column corresponds to the Lagrange multiplier, and is equal to $\mathbf{J}(\mathbf{x}, \mathbf{d}\omega)^T * (\mathbf{b} - \mathbf{A}(\omega + \mathbf{d}\omega) * \mathbf{x})$.

The formulation of $\mathbf{g}$ and Φ allows for the recovery of $\mathbf{y} \in \partial f(\mathbf{x}, \mathbf{d}\omega) + \mathbf{J}(\mathbf{x}, \mathbf{d}\omega)^T * (\mathbf{b} - \mathbf{A}(\omega + \mathbf{d}\omega) * \mathbf{x}) * \gamma$ such that $\mathbf{y}$ is the minimum norm vector

in the set. This is done by solving

$$\begin{aligned} \mathbf{u}^* = \quad & \underset{\mathbf{u}\in\mathbb{R}^{(|I^c_{\mathbf{x}}|+\left|I^c_{\mathbf{d}\omega}\right|+1)}}{\arg\min} && \|\mathbf{g}+\Phi*\mathbf{u}\|_2^2 \\ \text{subject to} \quad & && -1 \le u_1 \le 1 \\ & && -1 \le u_2 \le 1 \\ & && \dots \\ & && -1 \le u_{|I^c_{\mathbf{x}}|+\left|I^c_{\mathbf{d}\omega}\right|} \le 1 \\ & && 0 \le u_{|I^c_{\mathbf{x}}|+\left|I^c_{\mathbf{d}\omega}\right|+1} \end{aligned} \tag{2.10}$$

and then taking $\mathbf{y}^* = \mathbf{g}+\Phi*\mathbf{u}^*$. If $\|\mathbf{y}^*\|_2^2 = 0$ then the current point is a first-order critical point of the PSR problem. This problem can be solved using standard techniques for constrained least-squares problems. The problem can be extended to the case of complex $\mathbf{x}$ using the definitions of $\mathbf{x}_c$, $\mathbf{b}_c$, $\mathbf{A}_c$ and $\mathbf{J}_c$ and solved using Second-Order Conic Programming.

The first-order necessary conditions characterize the solutions to the PSR problem and provide a stopping criterion for the SLPSR algorithm. Critically, the subgradient for this problem takes a simple form and it is possible to test for first-order critical points using convex programming methods.

2.3 Solution of SLPSR Subproblems

In order to actually recover sparse signals and parameter perturbations using the SLPSR subproblem, it is necessary to have reasonably efficient solution techniques to the linearized subproblem $MinSubproblem$ and the feasible projection step $FeasibleProjection$. For both real-valued and complex-valued $\mathbf{x}$, there are at least two techniques to solve $MinSubproblem$. One potential method is formulation as Second-Order Conic Programming (SOCP) and another is subgradient descent. There are several solvers available for solving SOCP (for example SeDuMi [27]), allowing for easy implementation. SOCP allows for the formulation and solution of convex, quadratically constrained optimization problems. SOCP will find the unique minimizer for $MinSubproblem$. Subgradient descent, analogous to gradient descent for nonsmooth optimization, provides an alternative that is simpler to imple-

ment and has a low computational complexity per iteration. Because of the varying tradeoffs of both approaches, both are developed for the linearized subproblem. Finally, the feasible projection step can be cast as a constrained least-squares problem and solved with a combination of a projection step and a line-search.

2.3.1 Formulation as Second-Order Conic Program

The standard form [28] for SOCP is a linear objective function subject to conic and linear constraints.

$$\begin{aligned} \min_{\mathbf{x}} \quad & \mathbf{c}^T * \mathbf{x} \\ \text{subject to} \quad & \|\mathbf{Q}_i * \mathbf{x} + \mathbf{r}_i\|_2 \leq \mathbf{f}_i^T * \mathbf{x} + \mathbf{p}_i \\ & \mathbf{A} * \mathbf{x} = \mathbf{b} \end{aligned} \tag{2.11}$$

For $i = 1 \ldots m$, $\mathbf{Q}_i$, $\mathbf{r}_i$, $\mathbf{f}_i$, and $\mathbf{p}_i$ define the second-order cone constraints. This problem consists of a linear objective function subject to linear and convex constraints. Thus it has a unique minimizer that can be found efficiently through interior-point approaches.

To formulate *MinSubproblem* using SOCP for real-valued $\mathbf{x}$ and $\mathbf{b}$, one can introduce two auxiliary variables for each element of $\mathbf{x}$ representing the positive and negative component, $\mathbf{x}_+$ and $\mathbf{x}_-$. The variables $\mathbf{x}_+$ and $\mathbf{x}_-$ are constrained to have positive elements, and $\mathbf{x} = \mathbf{x}_+ - \mathbf{x}_-$. Similar auxiliary variables are defined for the real-valued parameter perturbations, $\mathbf{d}\omega_+$ and $\mathbf{d}\omega_-$. Given an initial, feasible point $(\mathbf{x}_k, \mathbf{d}\omega_k)$, the linearized subproblem

MinSubproblem can be rewritten as

$$\begin{aligned}
\underset{\mathbf{x}_+,\mathbf{x}_-,\mathbf{d}\omega_+,\mathbf{d}\omega_-}{\arg\min} \quad & \mathbf{1}_N^T * \mathbf{x}_+ + \mathbf{1}_N^T * \mathbf{x}_- + \lambda * \mathbf{1}_N^T * \mathbf{d}\omega_+ + \lambda * \mathbf{1}_N^T * \mathbf{d}\omega_- \qquad (2.12) \\
\text{subject to} \quad & \|\mathbf{d} + [\mathbf{J}, -\mathbf{J}] * (\begin{bmatrix} \mathbf{x}_+ \\ \mathbf{d}\omega_+ \\ \mathbf{x}_- \\ \mathbf{d}\omega_- \end{bmatrix})\|_2 \leq \epsilon \\
& \| \begin{bmatrix} (\mathbf{x}_+ - \mathbf{x}_-) - \mathbf{x}_k \\ (\mathbf{d}\omega_+ - \mathbf{d}\omega_-) - \mathbf{d}\omega_k \end{bmatrix} \|_2 \leq \Delta_k \\
& \mathbf{x}_+ \geq 0, \mathbf{x}_- \geq 0, \mathbf{d}\omega_+ \geq 0, \mathbf{d}\omega_- \geq 0
\end{aligned}$$

where each vector is constrained to have non-negative elements, $\mathbf{1}_N$ is a vector of length N composed of all ones, and the vector $\mathbf{d} \in \mathbb{R}^N$ is given by

$$\mathbf{d} = \mathbf{b} - \mathbf{A}(\omega + \mathbf{d}\omega_k) * \mathbf{x}_k + \mathbf{J} * (\begin{bmatrix} \mathbf{x}_k \\ \mathbf{d}\omega_k \end{bmatrix}) \qquad (2.13)$$

At the optimal point, this problem is the same as *MinSubproblem*, but is formulated using SOCP without linear equality constraints. This problem can then be implemented as a convex programming problem and solved via efficient interior point methods.

For complex-valued $\mathbf{x}$, a slightly different formulation is required to convert *MinSubproblem*. This formulation exploits the $\mathbb{CR}$ calculus representation developed in Section 2.2. The optimization is now carried out with respect to $\mathbf{x}_r$,$\mathbf{x}_i$,$\mathbf{d}\omega_+$,$\mathbf{d}\omega_-$, and a new set of parameters $\mathbf{t} \in \mathbb{R}^N$ is used to bound the elements of $\mathbf{x}$. Given an initial, feasible point $\mathbf{x}_k \in \mathbb{C}^N$, $\mathbf{d}\omega_k \in \mathbb{R}^N$, the

linearized subproblem $MinSubproblem$ can be rewritten as

$$\begin{aligned}
\underset{\mathbf{x}_r,\mathbf{x}_i,\mathbf{d}\omega_+,\mathbf{d}\omega_-,\mathbf{t}}{\arg\min} \quad & \mathbf{1}_N^T * \mathbf{t} + \lambda * \mathbf{1}_N^T * \mathbf{d}\omega_+ + \lambda * \mathbf{1}_N^T * \mathbf{d}\omega_- \qquad (2.14)\\
\text{subject to} \quad & \|\mathbf{d} + [\mathbf{J}_c] * (\begin{bmatrix} \mathbf{x}_r \\ \mathbf{x}_i \\ (\mathbf{d}\omega_+ - \mathbf{d}\omega_-) \end{bmatrix})\|_2 \leq \epsilon \\
& \| \begin{bmatrix} \mathbf{x}_r - \text{real}(\mathbf{x}_k) \\ \mathbf{x}_i - \text{imag}(\mathbf{x}_k) \\ (\mathbf{d}\omega_+ - \mathbf{d}\omega_-) - \mathbf{d}\omega \end{bmatrix} \|_2 \leq \Delta_k \\
& \mathbf{d}\omega_+ \geq 0, \mathbf{d}\omega_- \geq 0 \\
& \sqrt{x_{r,1}^2 + x_{i,1}^2} \leq t_1 \\
& \sqrt{x_{r,2}^2 + x_{i,2}^2} \leq t_2 \\
& \dots \\
& \sqrt{x_{r,N}^2 + x_{i,N}^2} \leq t_N
\end{aligned}$$

where the variables $\mathbf{d}\omega_-$ and $\mathbf{d}\omega_+$ are constrained to have non-negative elements, $\mathbf{1}_N$ is a vector of length N composed of all ones, and the vector $\mathbf{d} \in \mathbb{R}^N$ is given by

$$\mathbf{d} = \mathbf{b}_c - \mathbf{A}_c(\omega + \mathbf{d}\omega_k) * \mathbf{x}_{c,k} + \mathbf{J}_c * (\begin{bmatrix} \mathbf{x}_{c,k} \\ \mathbf{d}\omega_k \end{bmatrix}) \qquad (2.15)$$

In this formulation, the conic constraints $\sqrt{x_{r,j}^2 + x_{i,j}^2} \leq t_j$ enforce the ℓ_1 norm for complex-valued x. The SOCP formulation allows for efficient solutions of the subproblems for either real-valued or complex-valued $\mathbf{x}$.

2.3.2 Solution of Linearized Subproblems with Subgradient Descent

An alternative approach to solving the linearized subproblem is subgradient descent, which is analogous to gradient descent for smooth problems. Although SOCP formulations are capable of directly solving the linearized subproblem with a convex programming approach, it is often desirable to have a solution method that can be simply and directly implemented with

reasonable computational complexity. The notion of subgradients introduced in Section 2.2 provides the basis for a low-complexity, iterative solution.

Using the form of the subgradients introduced in Section 2.2, it seems reasonable to attempt a gradient descent approach. The key observation is that the solution to the problem

$$\min_{\mathbf{g}\in\partial f(\mathbf{x},\mathbf{d}\omega)} \|\mathbf{g}\|_2^2 \tag{2.16}$$

gives a descent direction that can be defined as $\frac{-\mathbf{g}}{\|\mathbf{g}\|_2}$ [24]. For the unconstrained case, a descent direction can be found using the definitions for real-valued or complex-valued subgradients as defined in Section 2.2. Finding $\mathbf{g}$ simply involves setting the elements corresponding to zero values of $\mathbf{x}$ and $\mathbf{d}\omega$ to zero. Each step of the gradient descent must be projected into the feasible set of the linearized constraint (similar to the method described in Section 2.3.3), and the total step must be constrained to Δ_k. The most important parameter for the subgradient descent approach is the step size μ. One reasonable approach is the Armijo-Wolfe rule [29] which balances the size of the step taken with the improvement in the objective function. Algorithm 2 provides pseudocode for subgradient descent with the Armijo rule. Various approaches for correcting or averaging descent directions over iterations can be employed [30]. Due to the reasonably simple form of the subdifferential and the descent direction, subgradient descent can be a reasonable approach for solving *MinSubproblem*.

2.3.3 Solution of the Feasible Projection Step

The remaining subproblem to be solved is the projection of the candidate step into the feasible subspace with $d\omega$ fixed. The function *FeasibleProjection* was defined as the minimum norm $\delta_{\mathbf{x}}$ that satisfies $\|\mathbf{b} - \mathbf{A}(\omega + \mathbf{d}\omega_k + \mathbf{s}_{\mathbf{d}\omega}) * (\mathbf{x}_k + \mathbf{s}_x + \delta_{\mathbf{x}})\|_2 \leq \epsilon$. This is a simple minimum-norm problem subject to a convex constraint. This can be accomplished using a standard projection operator and a bisection search.

First, take $\mathbf{A}_k = \mathbf{A}(\omega + \mathbf{d}\omega_k + \mathbf{s}_{\mathbf{d}\omega})$, and define the minimum-norm solution as $\mathbf{x}_{mn} = \mathbf{A}_k^\dagger * \mathbf{b}$, where $\mathbf{A}_k^\dagger$ is the Moore-Penrose pseudoinverse [31]. One can then define a shifted version of this problem, using the minimum norm solution. The projector onto the nullspace of $\mathbf{A}_k$ is also required, and can

Algorithm 2 MinSubproblem Solved with Subgradient Descent

Require: $\mathbf{x}, \mathbf{d}\omega, \omega, \lambda, \Delta, n, 0 < c < 1, 0 < \alpha < 1$

$k \leftarrow 1$
$\mathbf{s_x} \leftarrow 0$
$\mathbf{s_{d\omega}} \leftarrow 0$
$n_s \leftarrow \| \begin{bmatrix} \mathbf{s_x} \\ \mathbf{s_{d\omega}} \end{bmatrix} \|_2$
while $k < n$ and $n_s < \Delta$ **do**
    $\mathbf{g} \leftarrow \arg\min_{\mathbf{g} \in \partial f(\mathbf{x}+\mathbf{s_x}, \mathbf{d}\omega+\mathbf{s_{d\omega}})} \|\mathbf{g}\|_2$
    $\mathbf{d} \leftarrow -\mathbf{g}/\|\mathbf{g}\|_2$
    $\mu \leftarrow 1$
    $f_0 \leftarrow \|\mathbf{x} + \mathbf{s_x}\|_1 + \lambda * \|\mathbf{d}\omega + \mathbf{s_{d\omega}}\|_1$
    $\begin{bmatrix} \mathbf{t_x} \\ \mathbf{t_{d\omega}} \end{bmatrix} \leftarrow \begin{bmatrix} \mathbf{s_x} \\ \mathbf{s_{d\omega}} \end{bmatrix} + \mu * \mathbf{d}$
    $f \leftarrow \|\mathbf{x} + \mathbf{t_x}\|_1 + \lambda * \|\mathbf{d}\omega + \mathbf{t_{d\omega}}\|_1$
    while $f > f_0 + c * \mu * \mathbf{d}^T * \mathbf{g}$ **do**
        $\mu \leftarrow \alpha * \mu$
        $\begin{bmatrix} \mathbf{t_x} \\ \mathbf{t_{d\omega}} \end{bmatrix} \leftarrow \begin{bmatrix} \mathbf{s_x} \\ \mathbf{s_{d\omega}} \end{bmatrix} + \mu * \mathbf{d}$
        $f \leftarrow \|\mathbf{x} + \mathbf{t_x}\|_1 + \lambda * \|\mathbf{d}\omega + \mathbf{t_{d\omega}}\|_1$
    end while
    $\begin{bmatrix} \mathbf{s_x} \\ \mathbf{s_{d\omega}} \end{bmatrix} \leftarrow \begin{bmatrix} \mathbf{t_x} \\ \mathbf{t_{d\omega}} \end{bmatrix}$
    $n_s \leftarrow \| \begin{bmatrix} \mathbf{s_x} \\ \mathbf{s_{d\omega}} \end{bmatrix} \|_2$
    $k \leftarrow k + 1$
end while
return $\mathbf{s_x}, \mathbf{s_{d\omega}}$

be written using $\mathbf{A}_k^\dagger$, $\mathbf{P}_{N(\mathbf{A})} = (\mathbf{I} - \mathbf{A}_k^\dagger * \mathbf{A}_k)$. Taking $\delta_\mathbf{0} = \mathbf{x}_k + \mathbf{s}_x - \mathbf{x}_{mn}$ and $\delta_\mathbf{mn} = \mathbf{P}_{N(\mathbf{A})} * \delta_0$, the problem can now be viewed as a search for the first feasible point on the line connecting $\delta_\mathbf{0}$ and $\delta_\mathbf{mn}$. The problem is a one-dimensional bisection search over $\gamma \in \Big[0, 1\Big]$ for the zero of the function $\|\mathbf{A}_k * (\gamma * \delta_0 + (1 - \gamma) * \delta_{mn})\|_2 = \epsilon$. Due to the simple form of the feasible projection approach, this will generally be much faster than applying general quadratic programming solvers. Algorithm 3 shows the pseudocode for this approach.

Due to the special structure of the problems in the SLPSR algorithm, it is possible to formulate solvers with reasonably simple form. Next it is necessary to discuss the convergence of this algorithms to a first-order critical point.

Algorithm 3 Feasible Projection

Require: $\mathbf{x}, \mathbf{d}\omega, \omega, \mathbf{s_x}, \mathbf{s_{d\omega}}, \epsilon, \mathbf{b}, \mathbf{A}(\cdot), n_s$
$\mathbf{A_s} \leftarrow \mathbf{A}(\omega + \mathbf{d}\omega + \mathbf{s_{d\omega}})$
$\mathbf{x_{mn}} \leftarrow \mathbf{A_s}^\dagger * \mathbf{b}$
$\mathbf{P} \leftarrow (\mathbf{I} - \mathbf{A_s}^\dagger * \mathbf{A_s})$
$\delta_\mathbf{0} \leftarrow \mathbf{x} + \mathbf{s_x} - \mathbf{x_{mn}}$
$\delta_\mathbf{mn} \leftarrow \mathbf{P} * \delta_\mathbf{0}$
$\gamma \leftarrow 0.5$
$k \leftarrow 1$
$\delta_\mathbf{x} \leftarrow \delta_\mathbf{mn}$
while $k < n_s$ **do**
    if $(\|\mathbf{A_s} * (\gamma * \delta_\mathbf{0} + (1 - \gamma) * \delta_\mathbf{mn})\|_2 \leq \epsilon)$ **then**
        $\delta_\mathbf{x} \leftarrow \gamma * \delta_\mathbf{0} + (1 - \gamma) * \delta_\mathbf{mn}$
        $\gamma \leftarrow \gamma * 2$
    else
        $\gamma \leftarrow \gamma * 0.5$
    end if
    $k \leftarrow k + 1$
end while
return $(\delta_\mathbf{x} + \mathbf{x_{mn}}) - (\mathbf{x} + \mathbf{s_x})$

2.4 Convergence of the SLPSR Algorithm

Because of the nonlinear form of the constraint function, it is not immediately obvious that the SLPSR algorithm converges to a first-order critical point

of the PSR problem. This section examines the convergence of the SLPSR algorithm to a first-order critical point, as defined in Section 2.2. In the formulated PSR problem, the objective function is convex over $\mathbf{x}$ and $\mathbf{d}\omega$, but the constraint function has a nonlinear dependence on the variables. Because of this nonlinear dependence, it is difficult to say anything general about the convergence of the algorithm to a global minimizer. It will be shown, however, that the SLPSR algorithm will continue to reduce the objective function $\|\mathbf{x}\|_1 + \lambda * \|\mathbf{d}\omega\|_1$ as long as the current iterate is not at a first-order critical point. The proof assumes that the mapping $\mathbf{A}(\omega)$ is twice differentiable with respect to ω and that the norm of the Hessian of $\mathbf{A}(\omega)$ is finite with respect to all ω. This assumption is necessary to linearize the constraint and to bound the error incurred by linearizing the constraint. The proof also requires that the matrix $\mathbf{A}(\omega)$ have bounded spectral norm for all ω. This is required to bound the projection into the feasible set. Given these assumptions, one can show the following convergence result.

Theorem 1. *Given a feasible point* $(\mathbf{x} \in \mathbb{R}^N, \mathbf{d}\omega \in \mathbb{R}^N)$*, of the PSR problem which is not a first-order critical point,* $\exists$ *a* $\Delta > 0$ *such that the feasible projection of the linearized subproblem minimizer reduces the objective function of the PSR problem,* $\|\mathbf{x}\|_1 + \lambda * \|\mathbf{d}\omega\|_1$.

Proof. Given that the elements of $\mathbf{a}_i(\omega + \mathbf{d}\omega)$ are twice differentiable for all $\mathbf{d}\omega$, then the approximation error of the constraint function for a perturbation $\mathbf{s} = \begin{bmatrix} \mathbf{s_x} \\ \mathbf{s_{d\omega}} \end{bmatrix}$ from the point $(\mathbf{x}, \mathbf{d}\omega)$ can be bounded by the integral mean value theorem. Defining $\mathbf{r} = \mathbf{A}(\omega+\mathbf{d}\omega+\mathbf{s_{d\omega}})*(\mathbf{x}+\mathbf{s_x})-(\mathbf{A}(\omega+\mathbf{d}\omega)*\mathbf{x}+\mathbf{J}(\mathbf{x},\mathbf{d}\omega)*\mathbf{s})$ gives $\|\mathbf{r}\|_2 \leq c*\|\mathbf{s}\|_2^2$. This allows the formulation of a bound on the correction step $\delta_\mathbf{x}$ required to maintain feasibility.

There are two possible cases. The first case is when the constraint $\|\mathbf{b} - \mathbf{A}(\omega + \mathbf{d}\omega) * \mathbf{x}\|_2 \leq \epsilon$ is not tight (does not hold with equality). In this case, one can simply choose a Δ small enough that all points $\mathbf{x}+\mathbf{s_x}, \mathbf{d}\omega+\mathbf{s_{d\omega}}$ with $\|\mathbf{s_x}\|_2 + \|\mathbf{s_{d\omega}}\|_2 \leq \Delta$ are feasible. Then, if the current iterate is not a first-order critical point, the solution to the linearized subproblem will reduce the objective function. The candidate point $\mathbf{x} + \mathbf{s_x}, d\omega + \mathbf{s_{d\omega}}$ is already feasible and $\delta_\mathbf{x} = 0$. Then, by definition, the linear subproblem has reduced the function value and maintained feasibility.

The more complicated case occurs when the constraint is tight, as the

solution may no longer be feasible even for small values of Δ. A feasible solution is found by projecting the point into the feasible set with $\mathbf{d}\omega + \mathbf{s}_{\mathbf{d}\omega}$ fixed. From the residual error $\mathbf{r}$, one can bound the necessary correction step as $\delta_{\mathbf{x}} = \mathbf{A}^{\dagger}(\omega + \mathbf{d}\omega + \mathbf{s}_{\mathbf{d}\omega}) * \mathbf{r}$. Since the problem assumes $\mathbf{A}(\omega)$ has finite matrix norm for all ω, $\|\delta_{\mathbf{x}}\|_2 \leq d * \|\mathbf{s}\|_2^2$ for some d that does not depend on the size of the perturbation.

By considering the improvement in the function value for a simple subgradient descent strategy, we show next that if Δ is small enough the improvement in the model will be better than the worst-case correction step. This result will show that there exists a $\Delta > 0$ such that the subgradient approach will reduce the objective function unless the current point is a first-order critical point. The true minimizer will do at least as well as the subgradient approach. If $(\mathbf{x}, \mathbf{d}\omega)$ is not a first-order critical point, then there exists a descent direction $\hat{\mathbf{g}} = \frac{-\mathbf{g}}{\|\mathbf{g}\|_2}$ with $\mathbf{g}$ given by $\mathbf{g} = \arg\min_{\mathbf{g} \in \mathbf{P}_{N(\mathbf{J})} * \partial f(\mathbf{x}, \mathbf{d}\omega)} \|\mathbf{g}\|_2$.

One must find the optimal gradient step size $t \in \mathbb{R}$ from the point $(\mathbf{x}, \mathbf{d}\omega)$ for the descent direction. Define the PSR objective function as a weighted ℓ_1 norm, $\| \begin{bmatrix} \mathbf{x} \\ \mathbf{d}\omega \end{bmatrix} \|_{w,1} = \sum_i |x_i| + \lambda * \sum_j |d\omega_j|$. The step size is then given by

$$\begin{aligned} \underset{t \in \mathbb{R}}{\arg\min} \quad & \| \begin{bmatrix} \mathbf{x} \\ \mathbf{d}\omega \end{bmatrix} + t * \hat{\mathbf{g}} \|_{w,1} \\ \text{subject to} \quad & t \geq 0 \\ & \|t * \hat{\mathbf{g}}\|_2 \leq \Delta \end{aligned} \tag{2.17}$$

Now it must be shown that for the optimal t, the feasible projection of the gradient step is an improvement over the original point. This requires the following to be true

$$\begin{aligned} \| \begin{bmatrix} \mathbf{x} \\ \mathbf{d}\omega \end{bmatrix} + t * \hat{\mathbf{g}} + \begin{bmatrix} \delta_{\mathbf{x}} \\ 0 \end{bmatrix} \|_{w,1} \leq \\ \| \begin{bmatrix} \mathbf{x} \\ \mathbf{d}\omega \end{bmatrix} + t * \hat{\mathbf{g}} \|_{w,1} + d * \|\mathbf{s}\|_2^2 \leq \\ \| \begin{bmatrix} \mathbf{x} \\ \mathbf{d}\omega \end{bmatrix} \|_{w,1} \end{aligned} \tag{2.18}$$

The second term applies the triangle inequality and the Norm Equivalence

of norms on finite dimensional spaces ($C * \|\mathbf{x}\|_\alpha \leq \|\mathbf{x}\|_\beta \leq D * \|\mathbf{x}\|_\alpha$ for some constant C, D for all $\mathbf{x}$). The question is: does such a step $\mathbf{s}$ exist? For a small enough Δ, the constraint on the linearized subproblem will be tight, $\|\mathbf{s}\|_2 = \Delta$. Rewriting Equation 2.18, this then gives the requirement that

$$d * \Delta^2 \leq \| \begin{bmatrix} \mathbf{x} \\ \mathbf{d}\omega \end{bmatrix} \|_{w,1} - \| \begin{bmatrix} \mathbf{x} \\ \mathbf{d}\omega \end{bmatrix} + \Delta * \hat{\mathbf{g}} \|_{w,1} \tag{2.19}$$

As the weighted ℓ_1 norm is piecewise linear, the objective-function improvement becomes proportional to Δ, so $\| \begin{bmatrix} \mathbf{x} \\ \mathbf{d}\omega \end{bmatrix} \|_{w,1} - \| \begin{bmatrix} \mathbf{x} \\ \mathbf{d}\omega \end{bmatrix} + \Delta * \hat{\mathbf{g}} \|_{w,1} = \Delta * \mathbf{e}^T * \hat{\mathbf{g}} = c * \Delta$ for some unknown slope parameter $\mathbf{e}$. The term on the left-hand side of Equation 2.19 is proportional to Δ^2 because the worst case correction step is bounded by the square of the norm of $\mathbf{s}$. One then chooses a $\Delta > 0$ such that the inequality holds, or that the step remains feasible before projection. This will then reduce the objective function of the PSR problem.

As either case 1 or case 2 holds, it is possible to pick a $\Delta > 0$ such that the objective function is improved unless the current iterate is a first-order critical point. □

Given this result, the SLPSR algorithm will reduce the parameter Δ until a feasible iterate is found that reduces the objective function. This will continue until the first-order necessary conditions are satisfied, ensuring convergence to a first-order critical point.

This analysis can be extended to complex-valued $\mathbf{x}$ using the $\mathbb{CR}$ calculus as in Section 2.2. The construction of this proof gives no indication of convergence rate, but the uniform improvement of each iterate is a meaningful result for the algorithm. This implies that the SLPSR algorithm produces a series of iterates that are all strictly feasible and shows monotonic improvement in the objective function until a first-order critical point is reached.

2.5 Selection of Regularization Parameters

The SLPSR algorithm requires the selection of two parameters, λ and ϵ. The basic intuition is that λ weights the changes in parameters and changes in

the signal weights. Therefore, it is critical to choose $0 < \lambda \leq \frac{\Delta_{\mathbf{x}}}{\Delta_{\mathbf{d}\omega}}$ where $\Delta_{\mathbf{x}}$ is an approximation of the change in the ℓ_1 norm of $\mathbf{x}$ possible by perturbing the initial parameters. Similarly, $\Delta_{\mathbf{d}\omega}$ is an approximation of the increase in the ℓ_1 norm of $\mathbf{d}\omega$.

The inequality constraint parameter ϵ should be chosen with respect to the norm of the additive noise term ν. This can be seen by manipulating the assumed measurement model, $\mathbf{A}(\omega + \mathbf{d}\omega) * \mathbf{x} + \nu = \mathbf{b}$. Then $\|\mathbf{A}(\omega + \mathbf{d}\omega) * \mathbf{x} - \mathbf{b} + \nu\|_2 = 0$, so it is desirable to formulate the inequality constraint to be $\|\mathbf{A}(\omega + \mathbf{d}\omega) * \mathbf{x} - \mathbf{b}\|_2 \leq c * \|\nu\|_2$ for some multiplier c near 1. In practice, one wants to choose an ϵ slightly larger than the expected norm of the noise term. For example, if the noise term is a vector in $\mathbb{R}^N$ with elements identically, independently distributed as mean-zero Gaussian random variables with variance σ^2, then $E(\|\nu\|_2^2) = E(\nu^T * \nu) = \sum_{i=1}^{N} E(\nu_i * \nu_i) = N * \sigma^2$.

2.6 Initialization of SLPSR with BPDN Solution

An important corollary of the convergence result and the initialization of the SLPSR algorithm is that the SLPSR algorithm is guaranteed not to increase the ℓ_1 norm of the solution $\mathbf{x}$ over the BPDN solution for a given ω and ϵ. This can be seen from the fact that the SLPSR technique is initialized with the solution $\mathbf{x}$ of the BPDN problem and zero as the parameter perturbation. As the objective function, $\|\mathbf{x}\|_1 + \lambda * \|\mathbf{d}\omega\|_1$, will be reduced every time a new iterate is accepted it is clear that the ℓ_1 norm of $\mathbf{x}$ will be decreased if a new iterate is accepted. Therefore, the solution found by SLPSR will never have a higher ℓ_1 norm than the BPDN solution.

2.7 Relationship to MAP Estimation

Although the PSR problem has been formulated as a optimization problem, it is not immediately clear how to interpret this new problem. A potential interpretation of the PSR problem is as a MAP estimation problem with sparsity-inducing priors, similar to the methods developed in [21] and [32]. This formulation requires probabilistic models for all the relevant parameters, but provides benefits in terms of interpretation, extension, and parameter

tuning for the PSR problem.

The measurement model used to develop the PSR problem was given as $\mathbf{b} = \mathbf{A}(\omega + \mathbf{d}\omega) * \mathbf{x} + \nu$ where the measurements are $\mathbf{b} \in \mathbb{C}^M$, the unknown signal is $\mathbf{x} \in \mathbb{C}^N$, the initial parameter set is $\omega \in \mathbb{R}^N$, the unknown parameter perturbations are $\mathbf{d}\omega \in \mathbb{R}^N$, the measurement model $\mathbf{A}(\omega)$ is a mapping from $\mathbb{R}^N$ to $\mathbb{C}^{M \times N}$, and the additive noise term is $\nu \in \mathbb{C}^N$. The goal is to develop a Bayesian estimator for the unknown signal and parameter perturbation that promotes sparsity in the recovered solution.

When there is no particular known structure to the noise term ν, a Gaussian noise assumption is appealing due to its simplicity and its applicability to many cases. For complex-valued ν, this model assumes that the noise is distributed as a multivariate, mean-zero, white, circularly symmetric complex Gaussian, $\nu \sim \mathcal{CN}(\mathbf{0}, \sigma^2 * \mathbf{I})$. The probability density function of ν is then given by

$$\mathcal{CN}(\nu|\mathbf{0}, \sigma^2 * \mathbf{I}) = \frac{1}{\pi^M * |\sigma^2 * \mathbf{I}|} * \exp\{\nu^H \mathbf{\Sigma}^{-1} \nu\} \tag{2.20}$$

Here σ, the noise variance, is assumed to be known. Because any linear transformation of complex, Gaussian random variables is also complex and Gaussian, one can form the conditional distribution the measurements $\mathbf{b}$ as $P(\mathbf{b}|\mathbf{x}, \sigma, \mathbf{d}\omega) \sim \mathcal{CN}(\mathbf{A}(\omega + \mathbf{d}\omega) * \mathbf{x}, \sigma^2 * \mathbf{I})$.

Next, one must introduce a prior on $\mathbf{x} \in \mathbb{C}^N$ aimed at enforcing sparsity. One possible choice is the Laplacian distribution, which is sharply peaked at the mean with fairly heavy tails. For mean-zero variables, the Laplacian tends to favor zero, but allows for large elements with fairly high probability. As will be seen, the log-likelihood of this distribution also has a strong connection to ℓ_1 objective functions. Assuming independence between the elements of $\mathbf{x}$, the prior distribution on x_i is $\frac{1}{z}\exp(-\lambda_{\mathbf{x}} * |x_i|)$ where z is a normalization factor and $\lambda_{\mathbf{x}}$ is a known parameter that is constant for all x_i. Using the independence assumption,

$$\begin{aligned} P(\mathbf{x}|\lambda_{\mathbf{x}}) &= \frac{1}{z} * \exp(-\lambda_{\mathbf{x}} * |x_1|) * \exp(-\lambda_{\mathbf{x}} * |x_2|) * \dots \\ &\quad * \exp(-\lambda_{\mathbf{x}} * |x_N|) \\ &= \frac{1}{z} * \exp(-\lambda_{\mathbf{x}} * \|\mathbf{x}\|_1) \end{aligned} \tag{2.21}$$

where z is again a normalization constant. This prior will serve to promote a signal $\mathbf{x}$ with many zero elements.

For the parameter perturbations $\mathbf{d}\omega \in \mathbb{R}^N$, it is not immediately obvious what an appropriate prior would be. A key observation is that the parameter perturbations from the initial set of parameters ω should also be sparse. The perturbations should not change parameters corresponding to zero elements of $\mathbf{x}$, specifically $\text{supp}(\mathbf{d}\omega) \subseteq \text{supp}(\mathbf{x})$. A sparsity-promoting prior similar to the prior applied to $\mathbf{x}$ would be appropriate. Assuming the elements of $\mathbf{d}\omega$ are independent, each element is taken to be Laplacian distributed with probability density function given by $\frac{1}{z} * \exp\left(-\lambda_{\mathbf{d}\omega} * |d\omega_i|\right)$ where z is a normalization factor and $\lambda_{\mathbf{d}\omega}$ is a known parameter. Because the elements of $\mathbf{d}\omega$ are assumed independent, the distribution over $\mathbf{d}\omega$ can be written as

$$\begin{aligned} P(\mathbf{d}\omega|\lambda_{\mathbf{d}\omega}) &= \frac{1}{z} * \exp(-\lambda_{\mathbf{d}\omega} * |d\omega_1|) * \exp(-\lambda_{\mathbf{d}\omega} * |d\omega_2|) * \dots \\ &\quad * \exp(-\lambda_{\mathbf{d}\omega} * |d\omega_N|) \\ &= \frac{1}{z} * \exp(-\lambda_{\mathbf{d}\omega} * \|\mathbf{d}\omega\|_1) \end{aligned} \tag{2.22}$$

where z is again a normalization constant. Similarly to the prior on $\mathbf{x}$, the prior on $\mathbf{d}\omega$ will serve to promote a signal $\mathbf{d}\omega$ with many zero elements.

Assuming that the signal amplitudes $\mathbf{x}$ and parameter perturbations $\mathbf{d}\omega$ are independent, one can form the posterior to estimate the unknowns $\mathbf{x}$ and $\mathbf{d}\omega$. The posterior is given by

$$\begin{aligned} P(\mathbf{x}, \mathbf{d}\omega|\mathbf{b}, \lambda_{d\omega}, \lambda_x, \sigma) &= \frac{P(\mathbf{x}, \mathbf{d}\omega, \mathbf{b}|\lambda_{\mathbf{d}\omega}, \lambda_{\mathbf{x}}, \sigma)}{P(\mathbf{b}|\lambda_{d\omega}, \lambda_x, \sigma)} \\ &\propto P(\mathbf{b}|\mathbf{x}, \mathbf{d}\omega, \lambda_{\mathbf{d}\omega}, \lambda_{\mathbf{x}}, \sigma) * P(\mathbf{x}|\lambda_{\mathbf{x}}) * P(\mathbf{d}\omega|\lambda_{\mathbf{d}\omega}) \end{aligned} \tag{2.23}$$

Taking the negative of the log-likelihood of this posterior and minimizing over $\mathbf{x}$ and $\mathbf{d}\omega$ yields the following optimization problem (ignoring constant terms)

$$\min_{\mathbf{x}, \mathbf{d}\omega} \frac{1}{\sigma^2} \|\mathbf{b} - \mathbf{A}(\omega + \mathbf{d}\omega) * \mathbf{x}\|_2^2 + \lambda_{\mathbf{x}} * \|\mathbf{x}\|_1 + \lambda_{\mathbf{d}\omega} * \|\mathbf{d}\omega\|_1 \tag{2.24}$$

Interestingly, this formulation is identical to a dual formulation of the PSR problem, where the constraint $\|\mathbf{b} - \mathbf{A}(\omega + \mathbf{d}\omega) * \mathbf{x}\|_2^2 \leq \epsilon^2$ is integrated into the objective function, balanced by a constant multiplier.

Ultimately, the PSR formulation can be seen as a constrained dual of the MAP estimation formulation with sparsity-inducing Laplacian priors. However, the constant multiplier relating the two problems will be unknown beforehand. This formulation still has very interesting implications for the PSR problem. First, this interpretation helps motivate the PSR formulation as a form of MAP estimation. This also helps to relate the PSR problem to other potential parameter estimation techniques.

Importantly, the MAP interpretation allows for an alternative method of parameter selection. Consider the dual form of the PSR objective function, $\tau_1 * \|\mathbf{b}-\mathbf{A}(\omega+\mathbf{d}\omega)*\mathbf{x}\|_2^2+\tau_2*\|\mathbf{d}\omega\|_1+\|\mathbf{x}\|_1$. The MAP formulation can guide the selection of τ_1, τ_2, with the MAP formulation corresponding to $\tau_1 = \frac{1}{\lambda_{\mathbf{x}}*\sigma^2}$ and $\tau_2 = \frac{\lambda_{d\omega}}{\lambda_{\mathbf{x}}}$.

The MAP interpretation also allows for the introduction of different distributions for the noise model or signal priors. For example, suppose for real-valued measurements, signals, and noise, the noise is no longer white Gaussian. The noise correlations can then be captured in the correlation matrix $\mathbf{R}$, which will not be identity in general. This formulation can be translated into a weighted ℓ_2 norm, $\|\mathbf{b}-\mathbf{A}(\omega+\mathbf{d}\omega)*\mathbf{x}\|_{\mathbf{W},2}^2$, where $\mathbf{W} = \mathbf{R}^{-1}$. Another interesting extension possible from the MAP estimation interpretation is the use of hyperparameters to determine values for σ, $\lambda_{\mathbf{x}}$, and $\lambda_{\mathbf{d}\omega}$ when they are unknown. Appropriate conjugate priors can be chosen to make the problem tractable.

This chapter has developed the PSR problem and introduced the SLPSR algorithm to solve it. The SLPSR algorithm is guaranteed to find a first-order critical point of the PSR problem. By formulating the PSR problem, it is possible to recover both a sparse signal and a sparse set of parameter perturbations for recovery with a parameterized model. Chapter 3 presents a series of numerical experiments exploring the performance and implementation of the SLPSR algorithm for recovery of sparse signals using parameterized measurement models.

CHAPTER 3

NUMERICAL EXPERIMENTS

After developing the SLPSR algorithm and studying the PSR problem, it is imperative to validate this new technique with realistic simulations. This chapter presents a series of numerical experiments conducted in the MATLAB programming environment. Two common parameterized models are simulated: frequency-sparse signals and far-field source localization on a linear sensor array. This chapter presents a quantitative comparison of sparse recovery with standard BPDN, STLS, Perturbed CS via the AA-P-BPDN algorithm, and the proposed SLPSR algorithm. The goal of this chapter is to establish that sparse recovery is possible in these parameterized dictionaries and explore the various tradeoffs involved.

3.1 Implementation of Sparse Recovery Methods in MATLAB

In order to test the properties of these sparse recovery algorithms, it is first necessary to have practical implementations of these techniques. All of the methods studied in this chapter involve solving a convex objective function that can be seen as a convex relaxation of non-convex functions which enforce sparsity, such as the ℓ_0 pseudo-norm (the count of the number of nonzero terms in the recovered solution).

These methods seek a sparse solution to the underdetermined system of equations $\mathbf{b} = \mathbf{A}(\omega) * \mathbf{x}$. In this formulation, $\mathbf{b} \in \mathbb{C}^M$ is a measurement vector, $\mathbf{A}(\omega)$ is the parameterized measurement model mapping a set of parameters $\omega \in \mathbb{R}^N$ to a matrix in $\mathbb{C}^{(M \times N)}$, and $\mathbf{x} \in \mathbb{C}^N$ is a vector of unknown signal weights. The BPDN problem is implemented as a convex programming problem and the other formulations are solved iteratively by a series of convex programming problems.

The standard basis pursuit problem can be adapted to parameterized models given a fixed set of parameters ω. The BPDN problem is then given by

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{C}^N} \quad & \|\mathbf{x}\|_1 \\ \text{subject to} \quad & \|\mathbf{A}(\omega) * \mathbf{x} - \mathbf{b}\|_2 \leq \epsilon \end{aligned} \tag{3.1}$$

Here the signal dictionary $\mathbf{A}(\omega)$ is fixed. The parameter ϵ controls the deviation from the measurements and is used to correct for additive noise. In the noiseless case, the inequality constraint can be rewritten with equality. This convex programming problem for a complex-valued, unknown $\mathbf{x}$ is solved by reformulating the problem using SOCP and solving with the SeDuMi package for MATLAB [27]. This takes the form

$$\begin{aligned} \min_{\mathbf{t}\in\mathbb{R}^N} \quad & \mathbf{1}_N^T * \mathbf{t} \\ \text{subject to} \quad & \|\mathbf{A_c}(\omega) * \mathbf{x_c} - \mathbf{b_c}\|_2 \leq \epsilon \\ & \sqrt{x_{r,1}^2 + x_{i,1}^2} \leq t_1 \\ & \sqrt{x_{r,2}^2 + x_{i,2}^2} \leq t_2 \\ & \dots \\ & \sqrt{x_{r,N}^2 + x_{i,N}^2} \leq t_N \\ & t_i \geq 0, \forall i \end{aligned} \tag{3.2}$$

The vector $\mathbf{1}_N$ is of length N with every element equal to one.

For BPDN, the parameter ϵ must be chosen to compensate for additive noise to the measurements. Using simulated data, this parameter is selected by initializing ϵ to the norm of the known noise term and searching around this value to determine the parameter value that gives the best recovery performance.

Recovery using STLS was implemented using the alternating algorithm developed in [18], adapted for complex signal recovery. The STLS formulation can be written as

$$\min_{\mathbf{x}\in\mathbb{C}^N, \mathbf{dA}\in\mathbb{C}^{(M\times N)}} \|\mathbf{x}\|_1 + \lambda_1 * \|\mathbf{dA}\|_F^2 + \lambda_2 * \|(\mathbf{A}(\omega) + \mathbf{dA}) * \mathbf{x} - \mathbf{b}\|_2^2 \tag{3.3}$$

Here the signal dictionary is fixed given an initial sampling, and the problem

recovers a general perturbation $\mathbf{dA}$ and sparse signal $\mathbf{x}$. A stationary point of this problem can be found by first fixing $\mathbf{dA}$ and solving

$$\min_{\mathbf{x}\in\mathbb{C}^N} \|\mathbf{x}\|_1 + \lambda_2 * \|(\mathbf{A}(\omega) + \mathbf{dA}) * \mathbf{x} - \mathbf{b}\|_2^2 \tag{3.4}$$

which can be solved in MATLAB using the YALL1 package for complex-valued variables [33]. Then fixing $\mathbf{x}$, one can solve for $\mathbf{dA}$ using the following quadratic program.

$$\min_{\mathbf{dA}\in\mathbb{C}^{(M\times N)}} \lambda_1 * \|\mathbf{dA}\|_F^2 + \lambda_2 * \|(\mathbf{A}(\omega) + \mathbf{dA}) * \mathbf{x} - \mathbf{b}\|_2^2 \tag{3.5}$$

This approach will reduce the objective function with each iteration, and is a form of block coordinate descent. This iterative approach guarantees the convergence of this problem to a stationary point. For this work, recovery is run for a maximum of 100 iterations or until the objective function has not changed for 10 iterations. The STLS technique is initialized with the BPDN solution.

For STLS, it is necessary to select the parameters λ_1 and λ_2. The parameter λ_1 trades off between the norm of $\mathbf{x}$ and the perturbation norm. This parameter must be chosen small enough such that the change in the BPDN solution to the ideal weights, Δ_x is larger than the squared Frobenius norm of the difference between the original signal dictionary and the dictionary generated with the true signal parameters $\|\mathbf{dA}^*\|_F^2$. Searching for λ_1 around $\frac{\Delta_x}{\|\mathbf{dA}^*\|_F^2}$ should provide a reasonable regularization parameter. The parameter λ_2 can be initialized using the optimal dual variable found by the solution of the BPDN problem via SeDuMi. This dual variable corresponds to the optimal Lagrange multiplier of the BPDN problem used to initialize the STLS recovery. This should provide a good initial estimate of the regularization parameter.

The Perturbed CS model can be solved using the AA-P-BPDN algorithm.

For a parameterized model, this recovery problem can be written as

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{C}^N, \mathbf{d}\omega\in\mathbb{R}^N} \quad & \|\mathbf{x}\|_1 \\ \text{subject to} \quad & \|(\mathbf{A}(\omega) + \mathbf{B} * \Delta) * \mathbf{x} - \mathbf{b}\|_2 \leq \epsilon \\ & \|\mathbf{d}\omega\|_\infty \leq r \\ & \Delta = \text{diag}(\mathbf{d}\omega) \end{aligned} \tag{3.6}$$

Here the matrix $\mathbf{B}$ is a linear approximation to the change in the model $\mathbf{A}(\omega)$ with respect to ω. Choosing $\mathbf{B}$ will be discussed for each model. This problem is initialized with the BPDN solution then updated by first fixing $\mathbf{d}\omega$ and solving

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{C}^N} \quad & \|\mathbf{x}\|_1 \\ \text{subject to} \quad & \|(\mathbf{A}(\omega) + \mathbf{B} * \Delta) * \mathbf{x} - \mathbf{b}\|_2 \leq \epsilon \end{aligned} \tag{3.7}$$

Next, one fixes $\mathbf{x}$ and solves the following constrained least-squares problem

$$\min_{\mathbf{d}\omega\in\Big[-r,\, r\Big]^N} \|(\mathbf{A}(\omega) + \mathbf{B} * \Delta) * \mathbf{x} - \mathbf{b}\|_2 \tag{3.8}$$

This is again a form of block coordinate descent and will converge to a first-order critical point of the Perturbed CS model.

For the AA-P-BPDN algorithm, the parameters ϵ and r must be chosen. The parameter ϵ is selected as in the BPDN model. The parameter r controls the maximum possible parameter perturbation. Given an evenly spaced initial parameter sampling, r should intuitively be chosen to be less than half the sampling spacing to preserve uniqueness. The AA-P-BPDN algorithm was run for 100 iterations or until the solution had not improved the objective function significantly for 10 iterations.

Finally, the SLPSR technique is applied to the simulated data. The SLPSR algorithm is implemented as discussed in Chapter 2 using SOCP to solve the linearized subproblems. The algorithm was run for 100 iterations or until the first-order necessary conditions were satisfied.

3.2 Recovery of Simulated Frequency-Sparse Signals

In order to demonstrate the joint recovery of sparse vectors and parameter perturbations, this section considers the reconstruction of frequency-sparse signals from M complex-valued, time-domain measurements. The unknown parameters to be estimated are the digital frequencies and the unknown weights are the complex-valued amplitudes corresponding to those frequencies. The digital frequency is bound between zero and 2π for this recovery.

For this work, the simulated K-frequency-sparse signals will take the form $b(t) = \sum_{k=1}^{K} \beta_i * \exp(\imath * \omega_i * t)$ for $t = 0, 1, \ldots, M-1$ to form a vector $\mathbf{b} \in \mathbb{C}^M$. The goal of sparse recovery in this application is to provide an accurate reconstruction of this parameterized signal using very few recovery elements.

To simulate additive noise, a circularly symmetric, white, complex-valued Gaussian noise term was generated in MATLAB and added to the simulated measurements. The noise amplitude was adjusted to achieve the desired Signal-to-Noise Ratio (SNR).

An important question is how to sample the parameter space. One appealing approach is to evenly sample the digital frequency space, similar to the DFT. In this section the parameters will evenly sample the space, with $\omega_i = 2*\pi*i/(k*M), i = 0, 1, 2, \ldots, k*M-1$. Here k is the oversampling factor. Then the set of parameter samples will be $\omega \in \mathbb{R}^{(k*M)}$. Increasing k will increase the density of the sampling. When $k = 1$ the sampled frequencies correspond to the DFT frequencies.

For frequency-sparse signals, the parameterized model is given by

$$\mathbf{A}(\omega) = \Big[\mathbf{a}(\omega_1), \mathbf{a}(\omega_2), \cdots, \mathbf{a}(\omega_n)\Big] \tag{3.9}$$

where the elements of $\mathbf{a}(\cdot)$ are given by $a_j(\omega_i) = \frac{1}{\sqrt{M}} * \exp(\imath * \omega_i * j),\ j = 1, 2, \ldots, M$. Then $\mathbf{A}(\omega)$ maps the vector of frequencies ω^N to a matrix $\mathbf{A} \in \mathbb{C}^{M \times N}$ where each column is a complex sinusoid of frequency ω_i and of length M.

Given an initial set of parameters ω, the goal is to recover a set of sparse weights $\mathbf{x} \in \mathbb{C}^N$ and parameter perturbations $\mathbf{d}\omega \in \mathbb{R}^N$. The elements of $\mathbf{x}$ correspond to the complex amplitudes of each frequency component in $\mathbf{A}(\omega)$.

The final model element to consider is the Jacobian of the signal model.

For the SLPSR algorithm, this should be defined with respect to the real and imaginary components of $\mathbf{x}$ and the real-valued parameters, $x_j = x_{r,j} + \imath * x_{i,j}$. First, define the function $\partial\mathbf{a}(\omega_i)$,

$$\partial\mathbf{a}(\omega_i) = \begin{bmatrix} 0 * \imath * \frac{1}{\sqrt{M}} * e^{\imath * \omega_i * 0} \\ 1 * \imath * \frac{1}{\sqrt{M}} * e^{\imath * \omega_i * 1} \\ \cdots \\ (M-1) * \imath * \frac{1}{\sqrt{M}} * e^{\imath * \omega_i * (M-1)} \end{bmatrix} \tag{3.10}$$

Then the Jacobian of the model $\mathbf{A}(\omega + \mathbf{d}\omega) * \mathbf{x}$ at the point $\mathbf{x}, \mathbf{d}\omega$ with respect to $\mathbf{x_r}, \mathbf{x_i}$, and $\mathbf{d}\omega$ is given by

$$\begin{aligned} \mathbf{J}_c(\mathbf{x_r}, \mathbf{x_i}, \mathbf{d}\omega) = \Big[& \text{real}(\mathbf{A}(\omega + \mathbf{d}\omega)), -\text{imag}(\mathbf{A}(\omega + \mathbf{d}\omega)), \\ & \text{imag}(\mathbf{A}(\omega + \mathbf{d}\omega)), \text{real}(\mathbf{A}(\omega + \mathbf{d}\omega)), \\ & \text{real}(\partial\mathbf{a}(\omega_1) * (x_{r,1} + \imath * x_{i,1})), \text{real}(\partial\mathbf{a}(\omega_2) * (x_{r,2} + \imath * x_{i,2})), \ldots \\ & \text{imag}(\partial\mathbf{a}(\omega_1) * (x_{r,1} + \imath * x_{i,1})), \text{imag}(\partial\mathbf{a}(\omega_2) * (x_{r,2} + \imath * x_{i,2})), \ldots \\ & \ldots, \text{real}(\partial\mathbf{a}(\omega_n) * (x_{r,n} + \imath * x_{i,n})) \\ & \ldots, \text{imag}(\partial\mathbf{a}(\omega_n) * (x_{r,n} + \imath * x_{i,n})) \Big] \end{aligned} \tag{3.11}$$

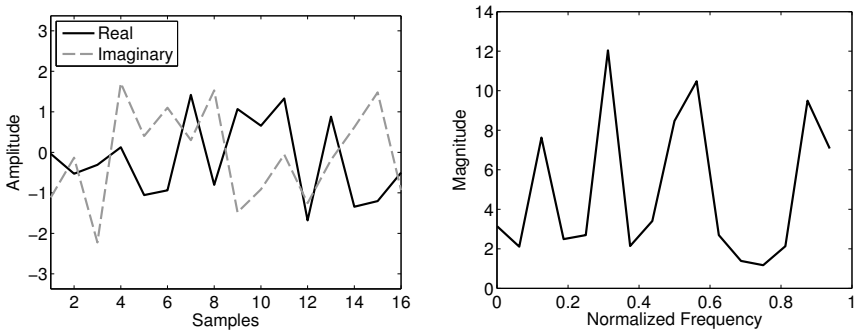
This defines the change in the real and imaginary parts of the model equations with respect to the variables $\mathbf{x}, \mathbf{d}\omega$.

Developing the Jacobian also allows one to define an appropriate matrix $\mathbf{B}$ for the linearized model used in the AA-P-BPDN algorithm. One can define the matrix $\mathbf{B}$ as

$$\mathbf{B} = \Big[\partial\mathbf{a}(\omega_1), \partial\mathbf{a}(\omega_2), \cdots, \partial\mathbf{a}(\omega_n)\Big] \tag{3.12}$$

To test recovery, four complex sinusoids were generated with frequencies ω^* uniformly randomly distributed between zero and 2π. If generated frequencies were closer than $\frac{2\pi}{M}$, ω^* was regenerated. The real and imaginary parts of the weights were drawn independently from uniform random distributions on $[1, -1]$. Thirty examples of such signals were generated to form a testing dataset. Circularly symmetric, white, complex-valued Gaussian noise was also simulated and added to the signal to generate noisy versions with 20 dB and 10 dB SNR. Figure 3.1 shows an example of this signal in the time-domain and the frequency domain.

Parameters were initialized as described in Section 3.1 using the known properties of the simulated data where necessary. For the BDPN, AA-P-



(a) Real and imaginary parts of simulated frequency-sparse data

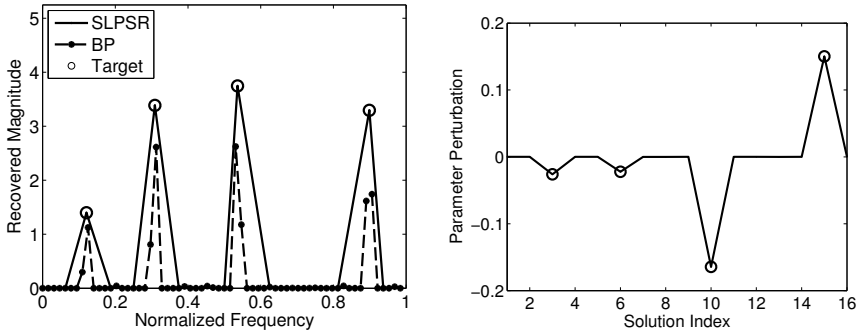
(b) DFT Magnitude of simulated frequency-sparse data

Figure 3.1: Example of simulated data containing four complex sinusoids without noise. The simulated data are 16 samples long. Figure 3.1(a) shows the complex, time-domain signal. Figure 3.1(b) shows the DFT magnitude of this data. Note that although there are four peaks in the DFT, the signal is not truly sparse in the DFT domain due to the effects of the finite-length window.

BPDN, and SLPSR methods, the parameter ϵ was swept from the initial estimate to four times the initial estimate. For the SLPSR recovery, the parameter λ was swept from one times the initial estimate to 0.1 times the initial estimate. The parameter r, for recovery using AA-P-BPDN, was varied to be 0.1 to 0.5 times the sampling spacing. For recovery with STLS, the parameters λ_1 and λ_2 were varied from 0.1 to 10 times the initial estimate.

These algorithms were all applied to the simulated frequency-sparse dataset to test sparse recovery approaches for this parameterized model. Figure 3.2 compares a sample recovery using BP with an eight-times oversampled dictionary and the proposed SLPSR recovery with an oversampling factor of one. The true amplitudes, parameters, and parameter perturbations are marked with circles. The BPDN recovery is reasonable, with several nonzero elements near the true parameter value used to recover the signal. The BPDN recovery, however, does use more than four coefficients to describe the signal. The SLPSR recovery, promisingly, consists of only four nonzero elements that exactly match the amplitude and parameter values of the simulated data. The SLPSR parameter perturbations also recover the unknown frequency parameter values.

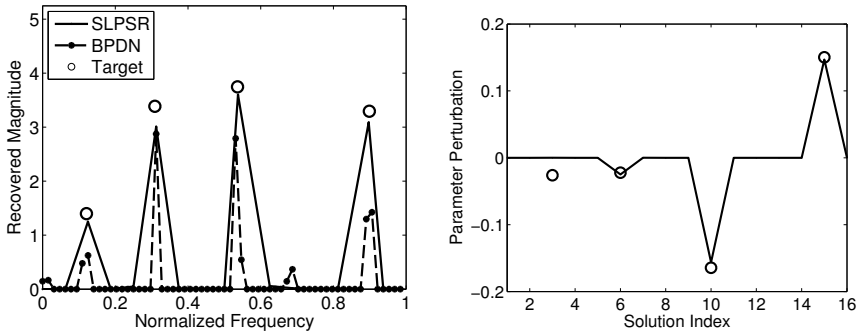
As noise is added to the simulated data, it is no longer reasonable to expect perfect recovery of the unknown signal weights and parameters using sparse



(a) Recovered coefficient magnitude, no noise

(b) Recovered parameter perturbations, no noise

Figure 3.2: Example recovery of simulated data containing four complex sinusoids with a noiseless measurement of sixteen samples. The SLPSR reconstruction with no oversampling and the BP reconstruction with eight-times oversampling are compared. The SLPSR method is able to exactly recover the unknown coefficients and parameter perturbations. The SLPSR solution uses only four nonzero elements.



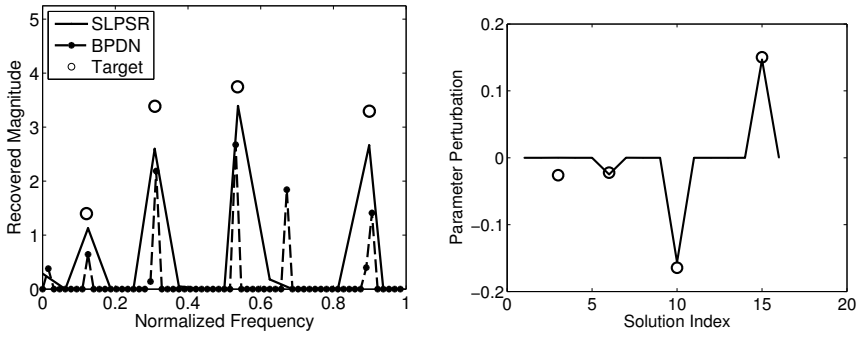
(a) Recovered coefficient magnitude, 20 dB SNR

(b) Recovered parameter perturbations, 20 dB SNR

Figure 3.3: Example recovery of simulated data containing four complex sinusoids with a SNR of 20 dB from 16 measurements. The SLPSR reconstruction with no oversampling and the BPDN solution with eight-times oversampling are compared. The same parameter ϵ is used for both reconstructions. The SLPSR solution uses fewer elements, has better parameter estimates, and more accurately recovers the unknown coefficients.

recovery techniques. Figures 3.3 and 3.4 show the recovery at 20 dB and 10 dB of the same simulated data in Figure 3.2. Again, the BPDN recovery is shown for an eight-times oversampled dictionary, and the SLPSR algorithm for a dictionary with an oversampling factor of one. The parameter ϵ

is the same for both reconstructions. The BPDN recovery recovers nonzero coefficients near the true parameter values, but shows odd artifacts in the recovery as noise is added to the signal. The SLPSR performance is degraded as well, but still seems to recover parameters and weights close to the true values. The signal coefficients are no longer exactly recovered, and the smallest parameter perturbations are not recovered. Still, the SLPSR technique seems to produce sparse parameter estimates without a highly oversampled dictionary.



(a) Recovered coefficient magnitude, 10 db SNR

(b) Recovered parameter perturbations, 10 dB SNR

Figure 3.4: Example recovery of simulated data containing four complex sinusoids with a SNR of 10 dB from 16 measurements. The SLPSR reconstruction with no oversampling and the BPDN reconstruction with four-times oversampling are compared. The same parameter ϵ is used for both reconstructions. Although the SLPSR recovery performance has degraded from the noiseless case, it still recovers fairly sparse solutions. The BPDN recovery generates a large artifact that could be mistaken for an additional frequency component.

Although example recoveries provide some interesting insight into sparse recovery for frequency-sparse signals, a quantitative comparison between BPDN, SLPSR, AA-P-BPDN, and STLS is required. Using the simulated frequency-sparse dataset, reconstructions using each technique were obtained for oversampling factors ranging from one to 16. The parameters for each reconstruction varied as previously described. Note that for the STLS technique, recovery with 16 times oversampling was not numerically stable using the parameter values tested. This oversampling factor is omitted from the STLS analysis.

For the reconstructions, three metrics were calculated to assess the quality of the sparse reconstructions. First, the coefficient recovery error captures

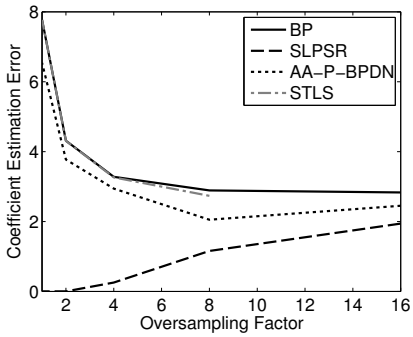
the accuracy of the true coefficient recovery by taking the absolute difference between the true coefficient, rounded to the nearest parameter value in the recovery, and the recovered coefficient at that parameter value. This is summed over all four frequency coefficients. Next, the error in the parameter recovery finds the indexes with coefficients larger than a small threshold (0.01 in this chapter) and computes the smallest absolute error between the recovered parameter values and the true parameter values. Note that because the STLS approach uses an unstructured perturbation, it is omitted from the parameter recovery analysis. Further assumptions would be required to estimate parameter values from the STLS solution. This metric is again summed over all four frequency components in the simulated signal. Finally, the number of nonzero elements is the number of recovered coefficients above the small threshold of 0.01. This metric assesses the sparsity of the recovered signal.

Over all tested parameter values, the values minimizing the coefficient recovery error were used to compute the three metrics. The results were averaged for all tested methods over the thirty simulated examples.

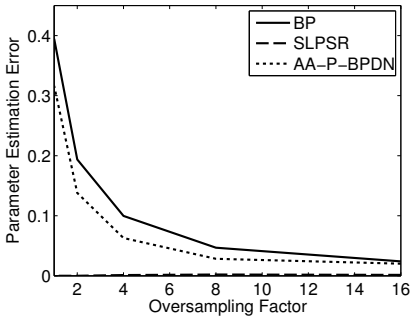
Figure 3.5 shows these metrics as a function of oversampling factor for the noiseless case. Generally, the recovery performance of AA-P-BPDN, BPDN, and STLS improves with the oversampling factor, but these methods are unable to exactly recover the frequency-sparse signal. Interestingly, the only method capable of near-exact recovery of the unknown coefficients and parameters of the frequency-sparse signal is the proposed SLPSR method. The best recovery using the SLPSR method is actually for low oversampling factors. The SLPSR technique does not require a highly oversampled signal dictionary for sparse recovery and actually shows better signal recovery with low oversampling factors.

Recovery of noiseless signals, however, is not of as much practical interest as recovery of signals with noise. The four sparse recovery techniques were applied to frequency-sparse signals with additive noise. The results are shown in Figures 3.6 and 3.7 for 20 dB and 10 dB SNR, respectively.

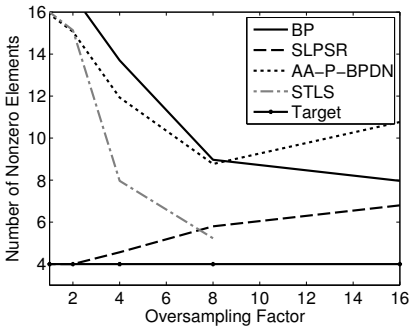
In the presence of additive noise, perfect recovery is no longer possible. The STLS technique is capable of recovering the coefficients more accurately than with the standard BPDN reconstruction, while using fewer elements on average. The STLS reconstruction generally improves with oversampling. The STLS technique does not, however, provide improved parameter esti-



(a) Error in recovered coefficients, no noise



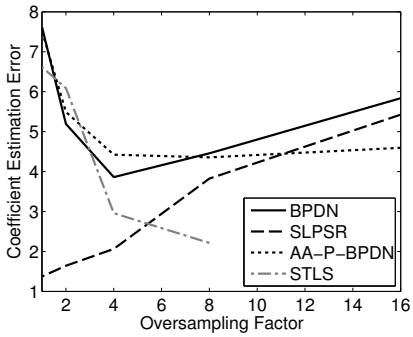
(b) Error in recovered parameter values, no noise



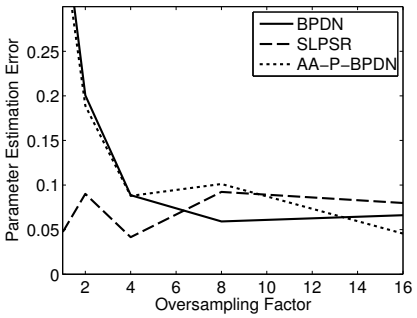
(c) Number of nonzero coefficients in recovery, no noise

Figure 3.5: Recovery of four sinusoids from 16 noiseless measurements, averaged over 30 trials. Figure 3.5(a) shows the average error in coefficient recovery for BP, SLPSR, STLS and AA-P-BPDN as a function of the oversampling factor. Figure 3.5(b) shows the average error in parameter recovery. Figure 3.5(c) shows the average number of nonzero coefficients recovered. The SLPSR approach is able to recover the signal with minimal error, even at coarse initial samplings. The other techniques are not able to exactly recover the signal even in the noiseless case.

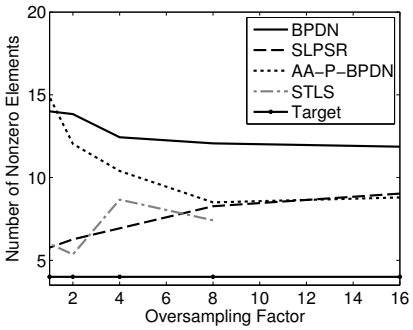
mates due to the unstructured parameter perturbation. The AA-P-BPDN algorithm generally improves on the recovery performance of BPDN, and outperforms the proposed SLPSR technique at high oversampling factors. This suggests that for highly oversampled dictionaries, the linear perturbation model used in the AA-P-BPDN algorithm may be more appropriate for reconstruction. Interestingly, the SLPSR solution again demonstrates improved recovery at lower oversampling factors. This technique produces solutions that are quite sparse, while still providing parameter recovery errors on par with the other algorithms tested. Performance of the SLPSR algorithm generally decreases (errors increase) with the oversampling factor. For



(a) Error in recovered coefficients, 20 dB SNR



(b) Error in recovered parameter values, 20 dB SNR

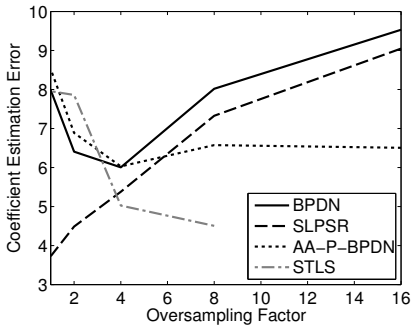


(c) Number of nonzero coefficients in recovery, 20 dB SNR

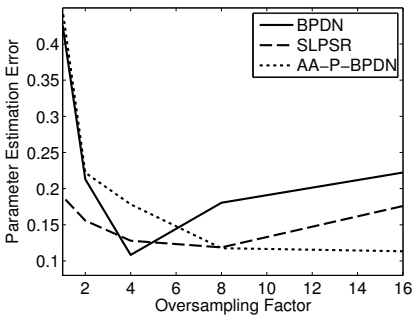
Figure 3.6: Recovery of four sinusoids from 16 measurements with 20 dB SNR, averaged over 30 trials. Figure 3.5(a) shows the average error in coefficient recovery for BPDN, SLPSR, STLS and AA-P-BPDN as a function of the oversampling factor. Figure 3.5(b) shows the average error in parameter recovery. Figure 3.5(c) shows the average number of nonzero coefficients recovered. The SLPSR approach with no oversampling provides the best performance of the available methods, estimating the parameters and the coefficients well with few nonzero coefficients.

frequency-sparse signals, the proposed SLPSR algorithm can perform sparse recovery without the need for a highly oversampled signal dictionary. In fact, the most accurate coefficient recovery of all tested methods was the SLPSR technique with no oversampling. This solution was consistently sparser than the AA-P-BPDN or BPDN solutions at any oversampling factor.

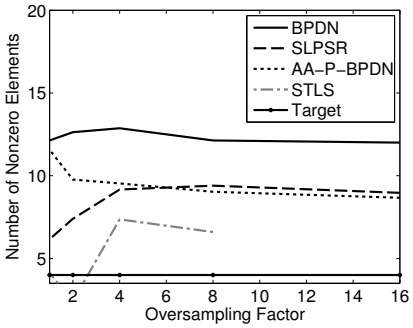
The proposed SLPSR algorithm is capable of recovering simulated, frequency-sparse signals without the need for highly oversampled signal dictionaries. The following sections present further numerical examples of SLPSR recovery of frequency-sparse signals to examine the sensitivity to parameters and super-resolution potential of this technique.



(a) Error in recovered coefficients, 10 dB SNR



(b) Error in recovered parameter values, 10 dB SNR



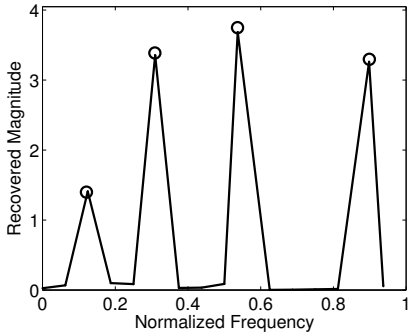
(c) Number of nonzero coefficients in recovery, 10 dB SNR

Figure 3.7: Recovery of four sinusoids from 16 measurements with 10 dB SNR, averaged over 30 trials. Figure 3.5(a) shows the average error in coefficient recovery as a function of the oversampling factor. Figure 3.5(b) shows the average error in parameter recovery. Figure 3.5(c) shows the average number of nonzero coefficients recovered.

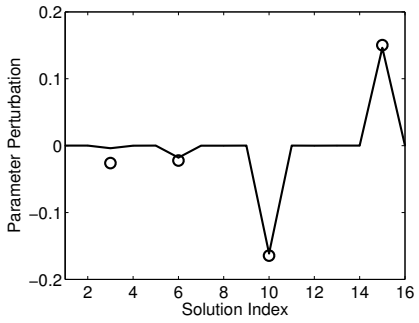
3.3 Parameter Sensitivity

For the proposed SLPSR recovery algorithm, there are two key parameters to select: λ and ϵ. This section contains some simple numerical examples of the SLPSR sensitivity to these two parameters for frequency-sparse signals.

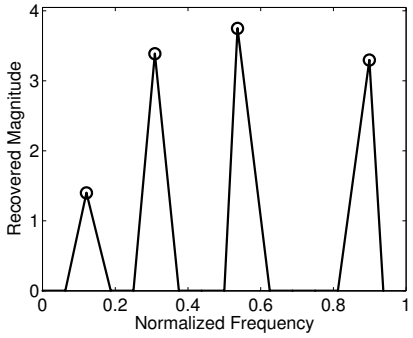
The parameter λ is used to weigh between the norm of the parameter perturbations and the norm of the sparse signal $\mathbf{x}$. Section 2.5 lays out the basic intuition for selecting this parameter; this section investigates the effect of varying this parameter on the recovery of frequency-sparse signals. Figure 3.8 shows the results of varying the parameter λ from the suggested bound to 0.1 times the suggested bound. Exactly at the calculated bound recovery is affected, but recovery is robust to the changes in parameter value below this guideline.



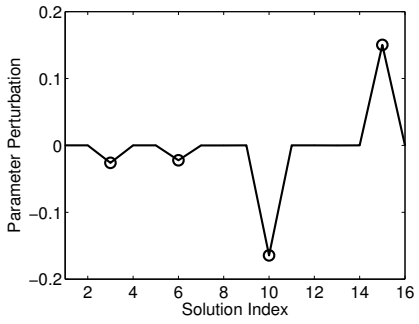
(a) Recovered coefficient magnitude, one times the calculated bound for λ



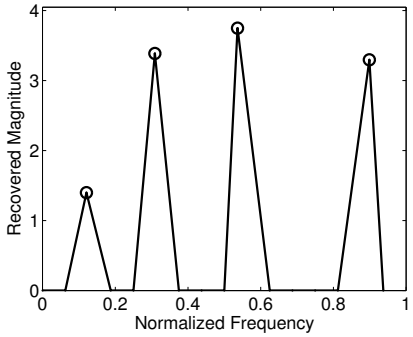
(b) Recovered parameter perturbations, one times the calculated bound for λ



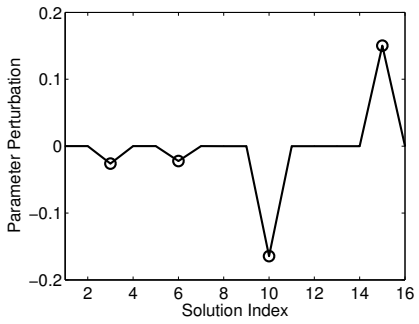
(c) Recovered coefficient magnitude, 0.5 times the calculated bound for λ



(d) Recovered parameter perturbations, 0.5 times the calculated bound for λ

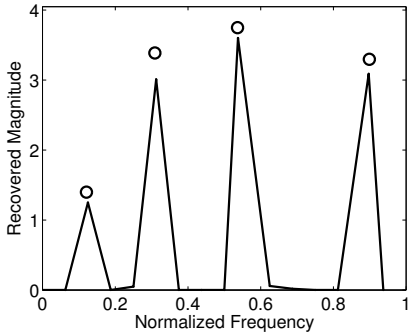


(e) Recovered coefficient magnitude, 0.1 times the calculated bound for λ

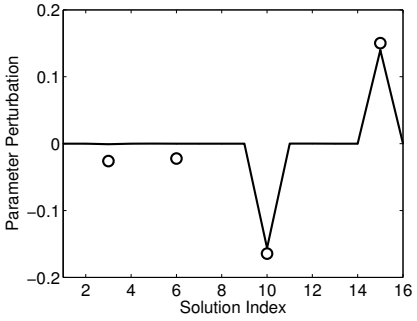


(f) Recovered parameter perturbations, 0.1 times the calculated bound for λ

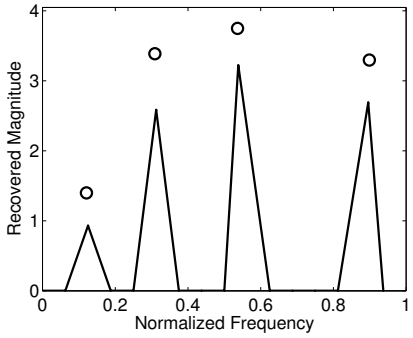
Figure 3.8: SLPSR recovery of four sinusoids from 16 noiseless measurements for different values of the regularization parameter λ. The λ value is taken as one, 0.5, and 0.1 times the bound calculated for this recovery. There are errors in recovery exactly at the calculated bound, but the recovery is relatively robust for a wide range of λ values.



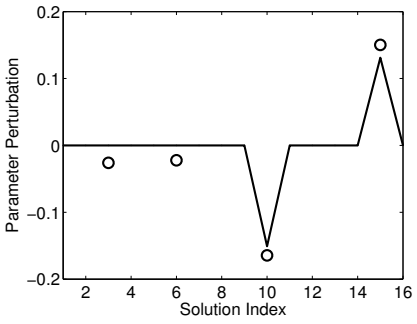
(a) Recovered coefficient magnitude, 20 dB SNR, one times the calculated bound for ϵ



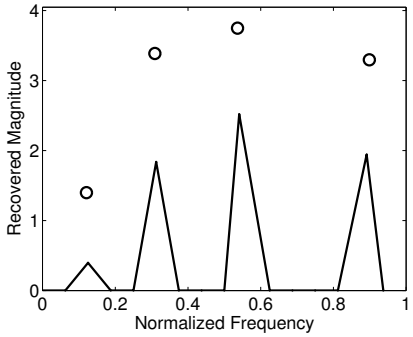
(b) Recovered parameter perturbations, 20 dB SNR, one times the calculated bound for ϵ



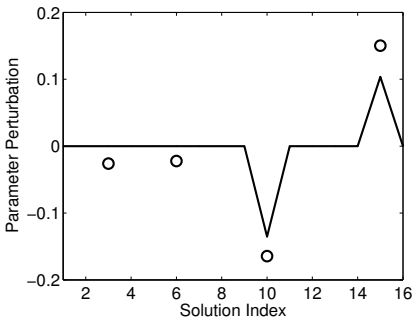
(c) Recovered coefficient magnitude, 20 dB SNR, two times the calculated bound for ϵ



(d) Recovered parameter perturbations, 20 dB SNR, two times the calculated bound for ϵ



(e) Recovered coefficient magnitude, 20 dB SNR, four times the calculated bound for ϵ



(f) Recovered parameter perturbations, 20 dB SNR, four times the calculated bound for ϵ

Figure 3.9: SLPSR recovery of four complex-valued sinusoids from 16 noiseless measurements for different values of the regularization parameter ϵ. The ϵ value is taken as one, two, and four times the bound calculated for this recovery. Adjusting this parameter can significantly impact the recovered coefficients.

For this algorithm, the other major parameter to select is ϵ, which allows for deviations from the measurements to correct for noise. Intuitively, ϵ should be chosen to be slightly larger than the norm of any additive noise term. Figure 3.9 shows the effect of varying this parameter from one times the norm of the additive noise term to four times the additive noise term, at 20 dB SNR. The recovery is more sensitive to the selection of this parameter than λ. Selecting an ϵ that is too large results in degraded recovery, especially of the signal coefficients.

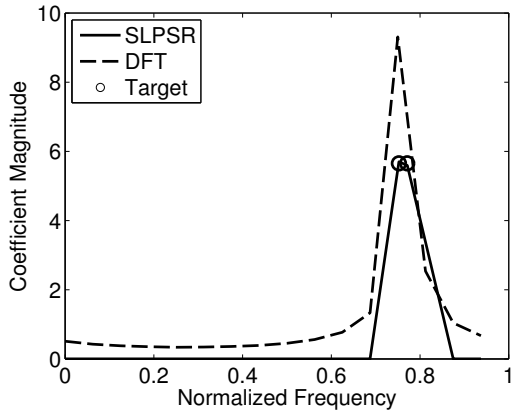
For sparse recovery with the SLPSR algorithm, careful selection of the two key parameters are required. The recovery, however, seems to be less heavily affected by the parameter λ as long as it is below the selection guidelines developed in Section 2.5. The selection of the parameter ϵ requires more careful thought, but this is also required for the BPDN, AA-P-BPDN, and STLS (through the dual variable) algorithms.

3.4 Example of Super-Resolution

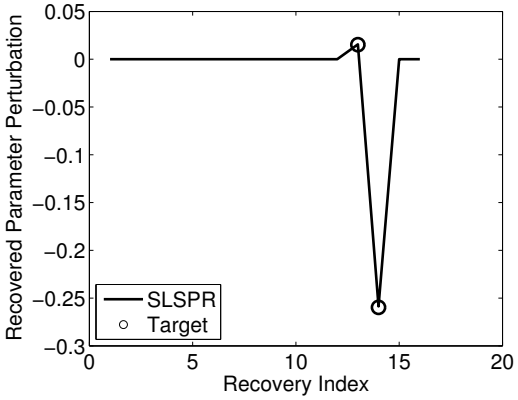
This work does not present a formal resolution analysis of the SLPSR algorithm for frequency-sparse signals. The estimation of the unknown frequency parameters, for instance, will be bounded by the statistical properties of this problem. As the algorithm is only convergent to a first-order necessary point, the achievable resolution of the SLPSR algorithm is also a function of the initial sampling of the parameter space. Given the performance of the SLPSR algorithm in the noiseless case, however, this algorithm could potentially exceed the classic resolution of the DFT. Regardless of zero-padding, the DFT is typically defined to have a resolution limit of $\frac{2*\pi}{M}$.

The SLPSR algorithm is able to resolve frequency components spaced more closely than $\frac{2*\pi}{M}$. Figure 3.10 shows the recovery of a two-frequency sparse signal at 20 dB SNR. The two frequency components are separated by $0.3 * \frac{2*\pi}{M}$. This example demonstrates that the SLPSR algorithm can recover the coefficients and parameter values of frequency-sparse signals with frequency components separated by less than the DFT resolution limit.

Using sparse recovery with BPDN and an oversampled signal dictionary, it is also possible to resolve closely spaced frequency components. Figure 3.11 demonstrates BPDN recovery for two- and eight-times oversampled dic-



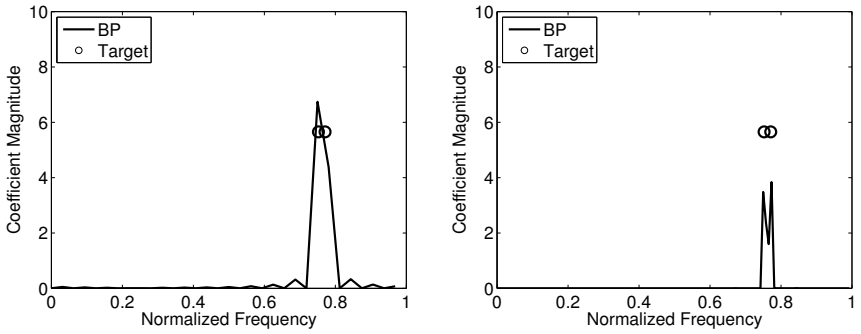
(a) Recovered coefficient magnitude using SLPSR, 20 dB SNR



(b) Recovered parameter perturbations using SLPSR, 20 dB SNR

Figure 3.10: Example recovery demonstrating the potential super-resolution capability of SLPSR. In this example, the length-16 signal is composed of two complex sinusoids with additive Gaussian noise at 20dB SNR. The spacing between the sinusoid frequencies is only $\frac{2*pi*0.3}{16}$, less than the Rayleigh resolution limit for this problem. In this example the DFT is unable to resolve the two frequencies, but SLPSR can recover the coefficients and parameter perturbations almost exactly.

tionaries. With the eight-times oversampled dictionary, there are two peaks corresponding to each transmitter. Clearly the resolution ability of BPDN depends on the sampling of the signal dictionary. However, it is important to note that simply increasing the dictionary oversampling will not necessarily increase the resolution. The fundamental resolution limit of the signal will depend on the frequency spacing, SNR, and relative amplitudes of the signals. Increasing oversampling factor does not automatically increase res-



(a) Recovered coefficient magnitude using BPDN, two-times oversampled, 20 dB SNR

(b) Recovered coefficient magnitude using BPDN, eight-times oversampled, 20 dB SNR

Figure 3.11: Example recovery demonstrating the potential super-resolution capability of standard BPDN recovery with evenly-spaced frequency sampling. In this example, the length-16 signal is composed of two complex sinusoids with additive Gaussian noise at 20dB SNR. The spacing between the sinusoid frequencies is only $\frac{2*pi*0.3}{16}$, less than the Rayleigh resolution limit for this problem. At an oversampling factor of eight, BPDN is capable of resolving two peaks, but does make significant errors estimating the signal amplitude coefficients. At an oversampling factor of two, BPDN is not able to resolve the peaks. Note that the resolution of recovery via BPDN is dependent on the oversampling factor.

olution ability, which is still governed by these fundamental limits.

After examining the recovery of frequency-sparse signals in detail, the final section of this chapter examines sparse recovery with a different parameterized model.

3.5 Simulated Source Localization with a Linear Array

Another common and important parameterized model arises in array processing, where signals arrive on an array of sensors and are spatially sampled. Here the goal is to recover the spatial location of signal sources from the signal arriving on the array. This work considers the simple case of simulated, narrow-band, far-field, complex-valued signals arriving on a linear array with equally spaced elements.

In this simplified case, the goal is to recover the angular direction of arrival θ of each source from the measurements on a linear array with equally spaced

sensors. Simulated complex sinusoidal sources of the same frequency were placed at a radial distance r and angle θ_i from the array center generating a time-domain signal $b_i * \exp(\imath * \omega * t)$. The wave propagates spatially with a speed c. As signals from different locations will arrive at the sensors at different times, it is possible, in principle, to determine the origin of the sources.

The simulated array consisted of M sensors in a linear pattern. The spacing between sensors was d. Then, depending on the origin of the source, the array measurements will have a different response. With this simple array geometry and far-field sources ($r >> M * d$), it is only possible to estimate the angle of arrival θ_i of each signal arriving at the array. For this geometry, the array response of a far-field signal from a direction θ_i is given by

$$\mathbf{a}(\theta_i) = \begin{bmatrix} 1 \\ \exp(-\imath * \frac{2\pi f d}{c} * \cos(\theta_i)) \\ \exp(-\imath * \frac{2\pi f d}{c} * \cos(\theta_i) * 2) \\ \vdots \\ \exp(-\imath * \frac{2\pi f d}{c} * \cos(\theta_i) * (M-1)) \end{bmatrix} \tag{3.13}$$

where c is the speed of the wave in meters per second, f is the frequency of the signal in hertz, and d is the array spacing in meters. Given a sampling of the direction of arrival from zero to π, the parameterized model $\mathbf{A}(\theta)$ can be defined with each column giving the response from a different θ_i.

For the AA-P-BPDN algorithm and the proposed SLPSR solution, it is necessary to define the function $\partial\mathbf{a}(\theta_i)$,

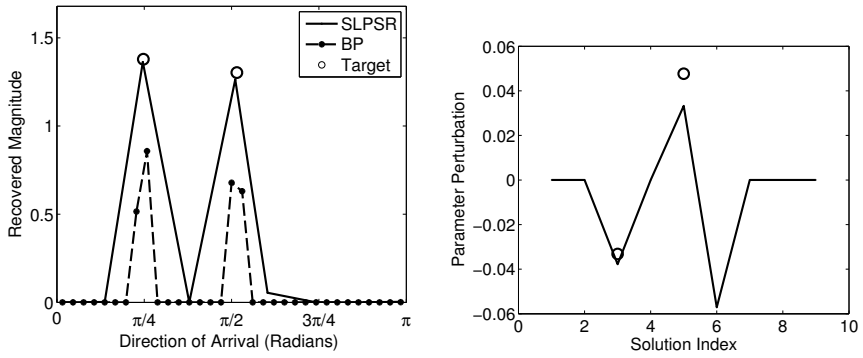
$$\partial\mathbf{a}(\theta_i) = \begin{bmatrix} 0 \\ \frac{2\pi\imath f d}{c} * \sin(\theta_i) * \exp(-\frac{2\pi\imath f d}{c} * \cos(\theta_i)) \\ \vdots \\ \frac{2\pi\imath f d(M-1)}{c} * \sin(\theta_i) * \exp(-\frac{2\pi\imath f d(M-1)}{c} * \cos(\theta_i)) \end{bmatrix} \tag{3.14}$$

This definition can be used to form the Jacobian and the matrix $\mathbf{B}$ required for AA-P-BPDN.

The goal then is to recover the complex-valued amplitudes and unknown directions of arrival of the incoming narrow-band, far-field sources. In this example, f was chosen small enough to allow for unambiguous source lo-

calization. Reconstructions were done using a single data snapshot, or one measurement from each sensor.

Thirty example snapshots were simulated of two signals incident on a linear array of eight elements. The source frequency was the same for both sources and across all simulations. The sources were placed at random angles generated uniformly between zero and π at a large radial distance. The real and imaginary parts of the signal amplitudes were drawn independently from a uniform random distribution over $\Big[-1, 1\Big]$. The time differences between the source locations and each array element were calculated and used to generate the signal at the array. To generate noisy signals, white, circularly symmetric, complex-valued Gaussian noise was added to the signal.



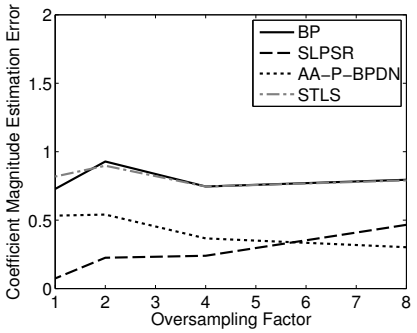
(a) Recovered coefficient magnitude, no noise

(b) Recovered direction of arrival parameter perturbations, no noise

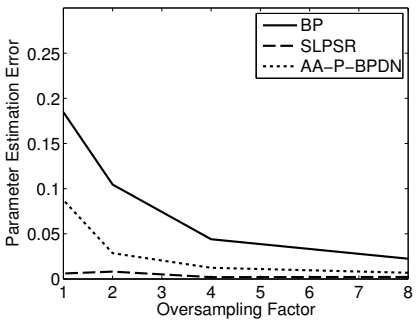
Figure 3.12: Example recovery of simulated source localization of two narrow-band, far-field sources with eight sensors. The SLPSR reconstruction with no oversampling and the BP reconstruction with four-times oversampling are compared. In this model, the SLPSR recovery is not exact, but does appear to improve parameter and coefficient estimates compared to the BP recovery.

Figure 3.12 shows the simulated recovery of an example, noiseless signal using BPDN and a four-times oversampled dictionary and the SLPSR algorithm with an oversampling factor of one. In this case, the SLPSR recovery is not exact, but does seem to improve sparse recovery when compared with the BPDN solution.

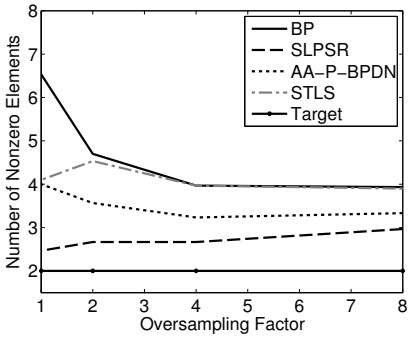
As with the frequency-sparse signals, the AA-P-BPDN, BPDN, STLS, and SLPSR algorithm were applied to the recovery of the simulated direction of arrival data. The recovery coefficient error, parameter recovery error, and



(a) Error in recovered coefficients, no noise



(b) Error in recovered parameter values, no noise

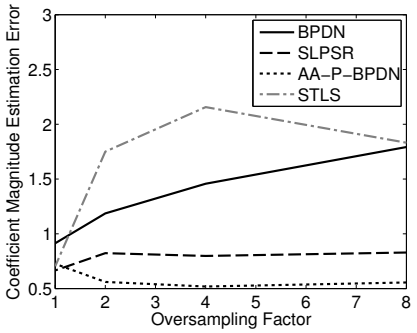


(c) Number of nonzero coefficients in recovery, no noise

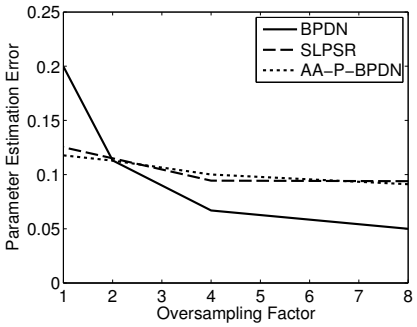
Figure 3.13: Recovery results of simulated source localization of two narrow-band, far-field sources with eight sensors. Figure 3.13(a) shows the average error in coefficient recovery for BP, SLPSR, STLS and AA-P-BPDN as a function of the oversampling factor. Figure 3.13(b) shows the average error in parameter recovery. Figure 3.13(c) shows the average number of nonzero coefficients recovered. Even in the noiseless case, none of the sparse-recovery algorithms are able to perfectly recover the signal. The SLPSR recovery with no oversampling, however, achieves the best recovery of all methods tested.

coefficient recovery error were calculated as for the frequency-sparse signals. Figure 3.13 shows the results of this for the noiseless case. For this model, the recovery is not quite error-free, even in the noiseless case. The SLPSR algorithm however, achieves the lowest error in the recovered coefficients and parameters while generating sparse solutions.

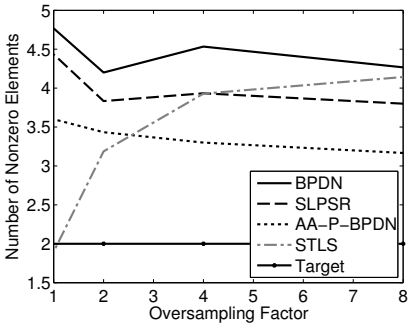
However, with this parameterized model, the recovery techniques do not perform as well with additive noise. Figure 3.14 shows the results of the recovery with 20 dB SNR. Here the sparse recovery techniques do not perform as well, in particular the SLPSR recovery. The AA-P-BPDN and SLPSR



(a) Error in recovered coefficients, 20 dB



(b) Error in recovered parameter values, 20 dB



(c) Number of nonzero coefficients in recovery, 20 dB

Figure 3.14: Example recovery of simulated source localization of two narrow-band, far-field sources with eight sensors in an equally spaced, linear array. Figure 3.14(a) shows the average error in coefficient recovery for BPDN, SLPSR, STLS and AA-P-BPDN as a function of the oversampling factor. Figure 3.14(b) shows the average error in parameter recovery. Figure 3.14(c) shows the average number of nonzero coefficients recovered.

techniques both improve coefficient estimates and generate sparser solutions than BPDN, but the linear perturbation model of AA-P-BPDN outperforms the SLPSR technique.

This example demonstrates that it is necessary to carefully consider the parameterized model when developing sparse recovery techniques.

Now that the proposed algorithm has been explored with simulated data, Chapter 4 demonstrates the sparse recovery of real-world RF data using the BPDN and SLPSR algorithms.

CHAPTER 4

RECONSTRUCTION RESULTS FOR SPARSE SIGNAL RECOVERY OF RADIO FREQUENCY DATA

The ultimate goal of developing sparse recovery techniques is to apply them to real-world signals. Many real-world signals exhibit considerable structure and are sparse in a particular signal dictionary. Using Radio-Frequency data, this chapter considers the recovery of real-world frequency-sparse signals. The goal is to recover sparse representations of these signals that accurately estimate the amplitudes and frequencies present in the signal. In this chapter, the techniques developed for simulated frequency-sparse signal recovery in Section 3.2 are applied to real-world data. The proposed SLPSR technique is used to recover sparse signals and parameter perturbations, and the results are compared against BPDN reconstructions.

An example of a real-world, frequency-sparse signal can be found in the High Frequency (HF) spectrum. This segment of the electromagnetic spectrum corresponds to frequencies between 3 megahertz and 30 megahertz. In parts of the HF spectrum, there are many amateur radio transmitters with well-defined center frequencies and minimal modulation around those center frequencies. Many of these transmitters are transmitting a form of continuous-wave Morse code, with long and short duration signals. These signals appear to be frequency-sparse over short segments, with the additional complication of transition events when the transmitters turn on or off.

This chapter applies the sparse recovery algorithm developed in this work to RF data, specifically the HF spectrum. The data was collected by Professor Steve Franke of the University of Illinois. Received RF signals were down-converted from a center frequency of 7.1 megahertz to complex baseband and sampled at 250 kilohertz. This dataset contains many different transmitters, and segments of the data appear to be frequency-sparse.

In order to create a dataset suitable for testing sparse recovery with a parameterized measurement model, the complex baseband signal with a 250

kilohertz bandwidth was bandpass filtered to a 16 kilohertz bandwidth, which was then down-sampled by a factor of eight. This dataset represents a narrow segment of the HF spectrum with several well-defined transmitters, which appears to be frequency-sparse.

4.1 Reconstruction of Radio-Frequency Data

To explore the potential of the proposed SLPSR algorithm for joint recovery of sparse signal weights and sparse parameter perturbations on real-world data, the BPDN algorithm and SLPSR algorithm were both applied to a segment of the RF data. Figure 4.1 shows the magnitude spectrogram corresponding to this segment of data. This spectrogram was generated with length-64 Hann windows with a 50% overlap. As can be seen in the figure, there appear to be transmitters in different bands broadcasting in short bursts. There is no particularly obvious modulation. Due to these observations, it is reasonable to assume these are radio transmitters broadcasting continuous-wave Morse code. Judging by the spectrogram, it appears that this data may be well modeled as a frequency-sparse signal.

The goal of applying sparse recovery techniques to this dataset is to represent the data as a combination of only a few complex-valued sinusoids. The formulation of Section 3.2 is applied to this data to recover a set of sparse signal weights $\mathbf{x}$ and, in the case of the SLPSR algorithm, a set of sparse parameter perturbations $\mathbf{d}\omega$.

The initial parameter sampling was chosen with an evenly spaced grid, the same sampling used when recovering the simulated data. The parameters ω_i correspond to $i * 2 * \pi/(k * M)$ where k is the oversampling factor and $i = 0, 1, \ldots, k * M - 1$. When k is equal to one, the parameters are not oversampled and correspond to the DFT frequencies. For both the SLPSR and BPDN algorithms, the parameter ϵ was varied between ten and twenty percent of the total power in each reconstructed data frame. For the SLPSR technique, the initial parameter λ was chosen using the guidelines in Section 2.5. The change possible in the ℓ_1 norm of $\mathbf{x}$, $\Delta_{\mathbf{x}}$, was taken to be 20% of the ℓ_1 norm of the BPDN solution. The estimated parameter perturbation, $\Delta_{\mathbf{d}\omega}$ was calculated by allowing for up to four elements to move up to one half of the spacing between parameter values. The parameter λ was then tuned

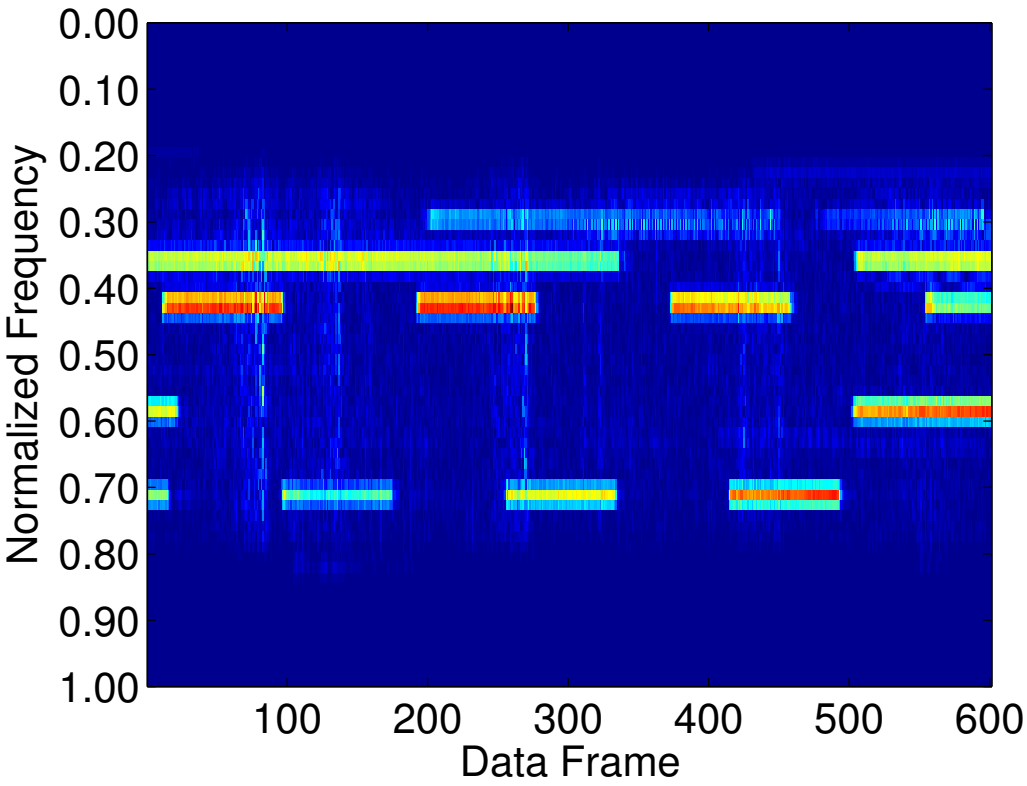
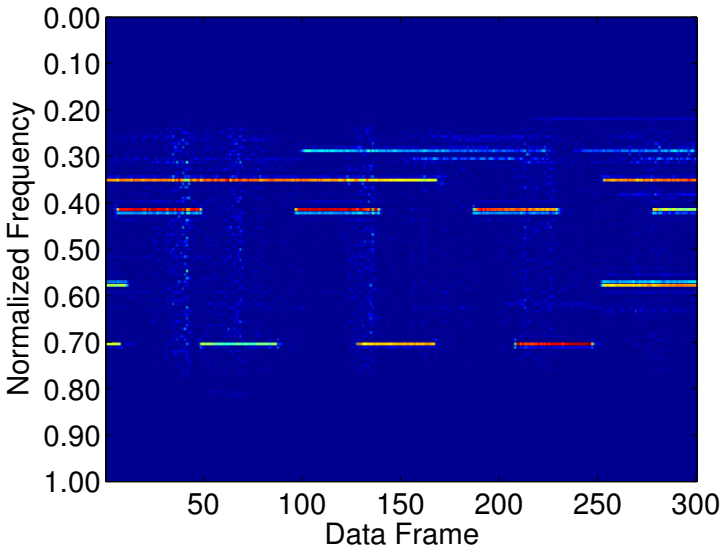


Figure 4.1: Magnitude Spectrogram of example section of Radio-Frequency spectral data. The spectrogram if formed with length-64 Hann windows with a 50% overlap. The data appear to be composed of complex sinusoids of finite durations. It is quite likely that this dataset captures transmitters in different bands broadcasting a form of continuous-wave Morse code. As the transmitters appear to be quite discrete and do not show obvious modulation, it is likely they can be well modeled as a complex, frequency-sparse signal.

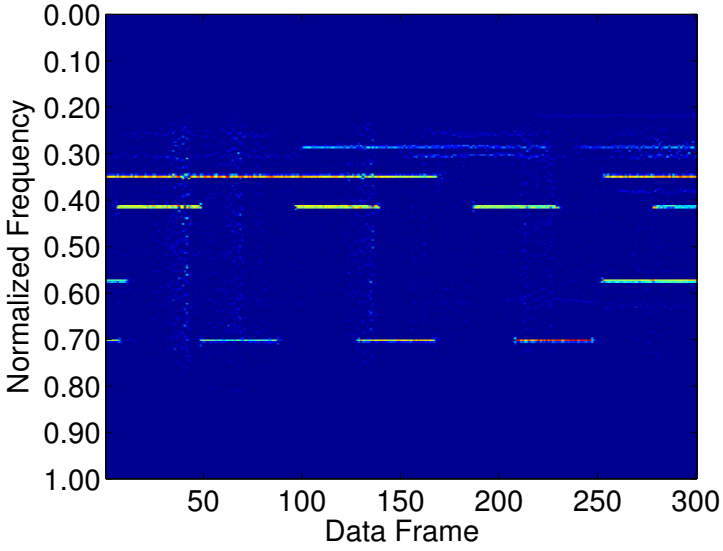
around this initial estimate.

Figure 4.2 shows two sample reconstructions using the BPDN technique with ϵ set to 10% of the total norm of the data. The reconstructions use length-64 windows of data with no overlap as measurement vectors. Results are shown with a two- and four-times oversampled dictionary. The results are displayed by taking a dense grid of frequencies and plotting the recovered sparse magnitudes $\mathbf{x}$ at the corresponding frequency parameter value.

The sparse solutions recovered by BPDN seem qualitatively reasonable. The recovered coefficients are clustered around the transmitters that are visible in the spectrogram. This gives an estimate of the measured data as a



(a) Coefficient Magnitude of sparse recovery using BPDN, using a two-times oversampled dictionary



(b) Coefficient Magnitude of sparse recovery using BPDN, using a four-times oversampled dictionary

Figure 4.2: Example sparse recovery of RF data with BPDN. Reconstructions are done with length-64, non-overlapping segments of data. This corresponds exactly to the time domain data used to generate the spectrogram in Figure 4.1. The BPDN recovery seems qualitatively reasonable. Studying the reconstructions in detail, however, it appears that the BPDN solution uses several nearby frequency elements to describe each transmitter, and the recovery quality varies with the initial sampling.

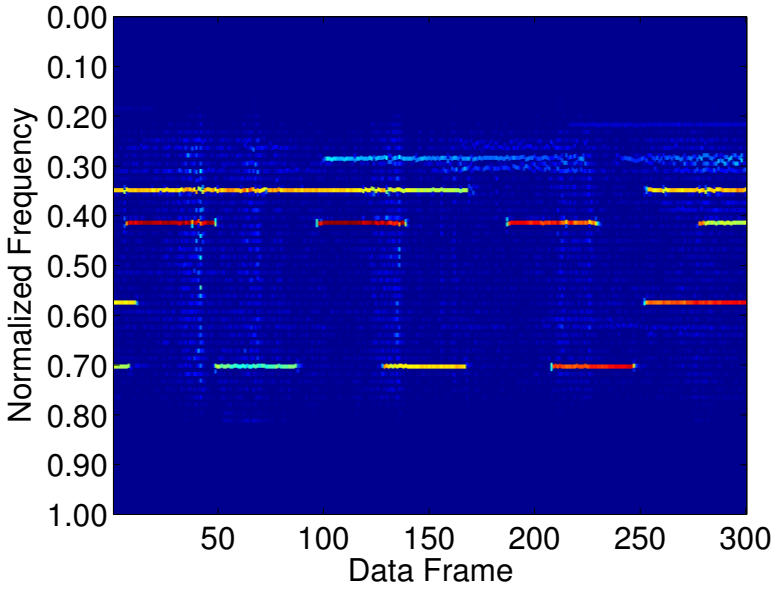
frequency-sparse signal, with some errors. Using a two times oversampled dictionary, the elements used to reconstruct a given transmitter can be quite spread apart. As oversampling increases, the sparse recovery appears to more closely approximate the center frequencies of the transmitters. It is important to note, however, that the BPDN solution still uses several elements to approximate each transmitter. Overall, the solutions seem reasonable, but the reconstruction depends heavily on the selection of the initial sampling grid. As the BPDN solutions use several elements to describe each transmitter, it may also be able to generate sparser solutions to represent this data.

After attempting recovery with BPDN, the SLPSR technique was applied to this dataset to see if the sparse recovery could be improved by allowing for parameter perturbations. The data was the same used for the BPDN recovery. Figure 4.3 shows the results of the SLPSR algorithm for the same parameter ϵ used for the BPDN recovery, with no frequency oversampling. The magnitude of the recovered sparse signal is plotted at the point corresponding to the final frequency parameter (the original frequency parameter plus the recovered parameter perturbation). The resulting parameter perturbations are shown by recovery index.

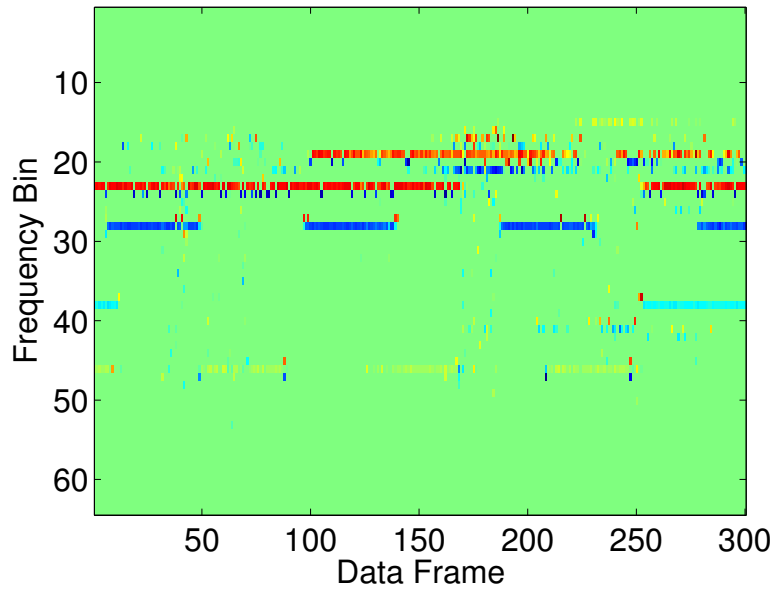
Reconstruction with the SLPSR technique also seems to produce qualitatively reasonable frequency-sparse reconstructions of this data. The recovered magnitudes match well with the transmitters observed in the spectrogram. The recovered parameter perturbations appear to place the transmitters at the correct frequencies. What is especially interesting is that this is achieved without a highly oversampled signal dictionary.

This result demonstrates that the SLPSR technique can represent frequency-sparse, real-world data without the need to generate highly oversampled dictionaries. Another observation is that the SLPSR technique appears to use fewer elements to describe each transmitter. It is possible that the SLPSR technique is actually recovering a sparser solution than the oversampled BPDN solution.

The contrast between recovery with BPDN and SLPSR can be seen more clearly when examining the magnitude plots in detail. Figure 4.4 shows a magnified version of the plots for BPDN recovery with two- and four-times oversampled dictionaries. From the magnified plots, it is clear that each transmitter is reconstructed using several frequency elements. The elements

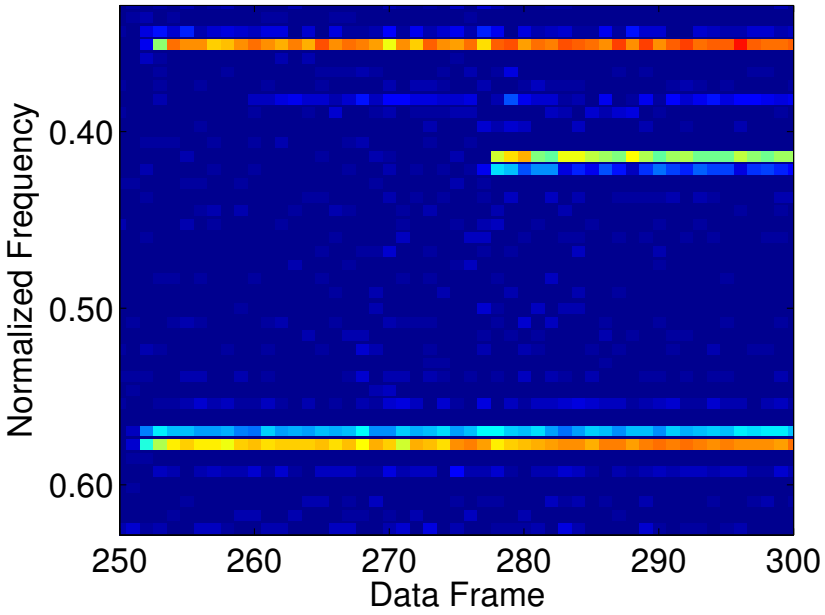


(a) Coefficient Magnitude of sparse recovery using SLPSR, no oversampling

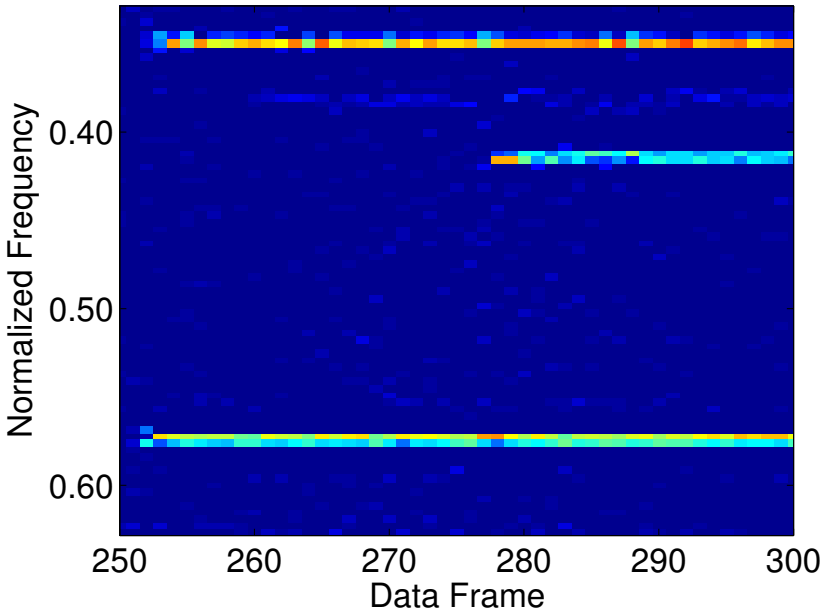


(b) Parameter Perturbations recovered with SLPSR, no oversampling

Figure 4.3: Example sparse recovery of RF data with the SLPSR technique. Reconstructions are done with length-64, non-overlapping segments of data. This corresponds exactly to the time domain data used to generate the spectrogram in Figure 4.1. Magnitude plots are generated by plotting the recovered coefficient magnitude for each frame using a dense grid of frequencies. Parameter perturbations are shown for each recovery index.



(a) Coefficient Magnitude of sparse recovery using BPDN, two-times oversampled dictionary, magnified



(b) Coefficient Magnitude of sparse recovery with BPDN, four-times oversampled dictionary, magnified

Figure 4.4: Example sparse recovery of RF data with BPDN. These plots show a magnified version of Figure 4.2 around data frames 250 to 300. BPDN recovery depends on the oversampling and uses several elements to describe each transmitter.

used vary from frame to frame. This problem still arises in the four-times oversampled dictionary. Although the BPDN solution seems qualitatively close to the underlying data, the solution is dependent on the initial sampling of the parameter space and is not necessarily the sparsest solution.

Figure 4.5 shows the magnified magnitude plot for the SLPSR technique. Here it appears that one large element is used to describe the transmitters. The frequency estimate varies somewhat from frame to frame, but seems quite consistent. This solution appears to explain the data as well as the BPDN solution while producing a sparser solution. A numerical comparison of performance using these two methods can help elucidate the differences between them.

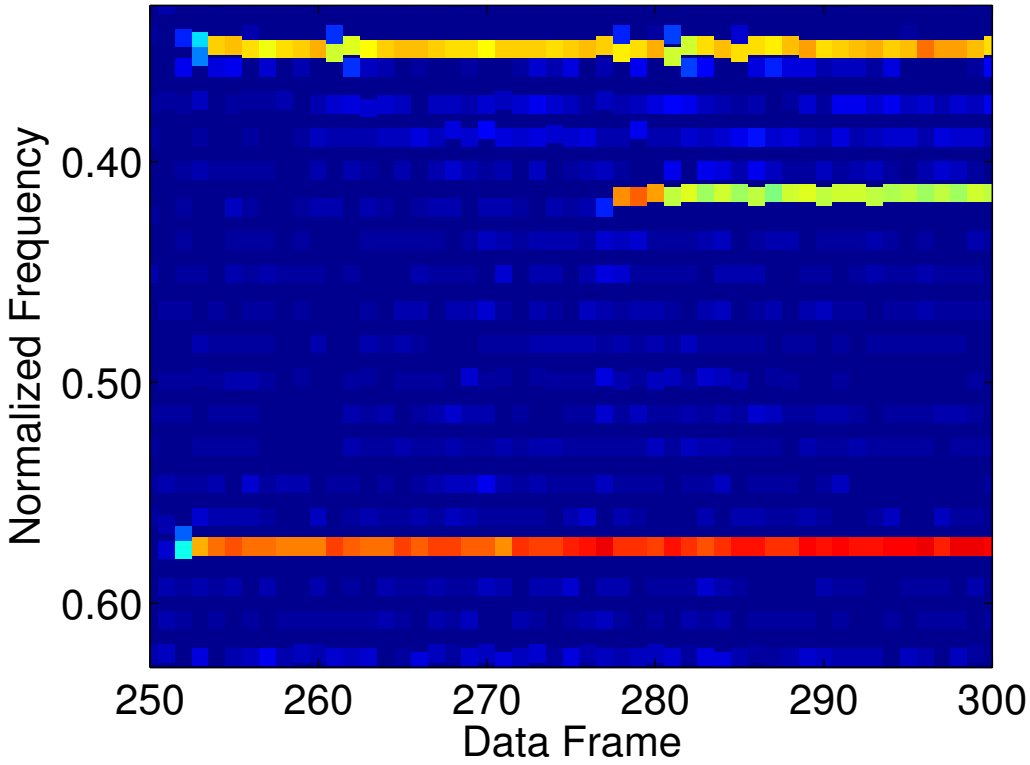


Figure 4.5: Coefficient magnitude recovered by the SLPSR method. This plot is a magnification of frames 250 to 300 of Figure 4.3. The SLPSR recovery seems to produce very sparse solutions that estimate the transmitters in the data.

To help quantitatively compare the two methods, a short segment of the radio frequency data was isolated. This short segment of data appears to have only two distinct transmitters, one of which ends transmission midway

through the data segment. For this example, length-16 windows were used to allow for reconstruction with many different oversampling factors and parameter values. The spectrogram of this data is shown in Figure 4.6 with length-16 Hann windows with a 50% overlap. The two transmitters can be identified with this window length, but have quite broad main peaks due to the short window size.

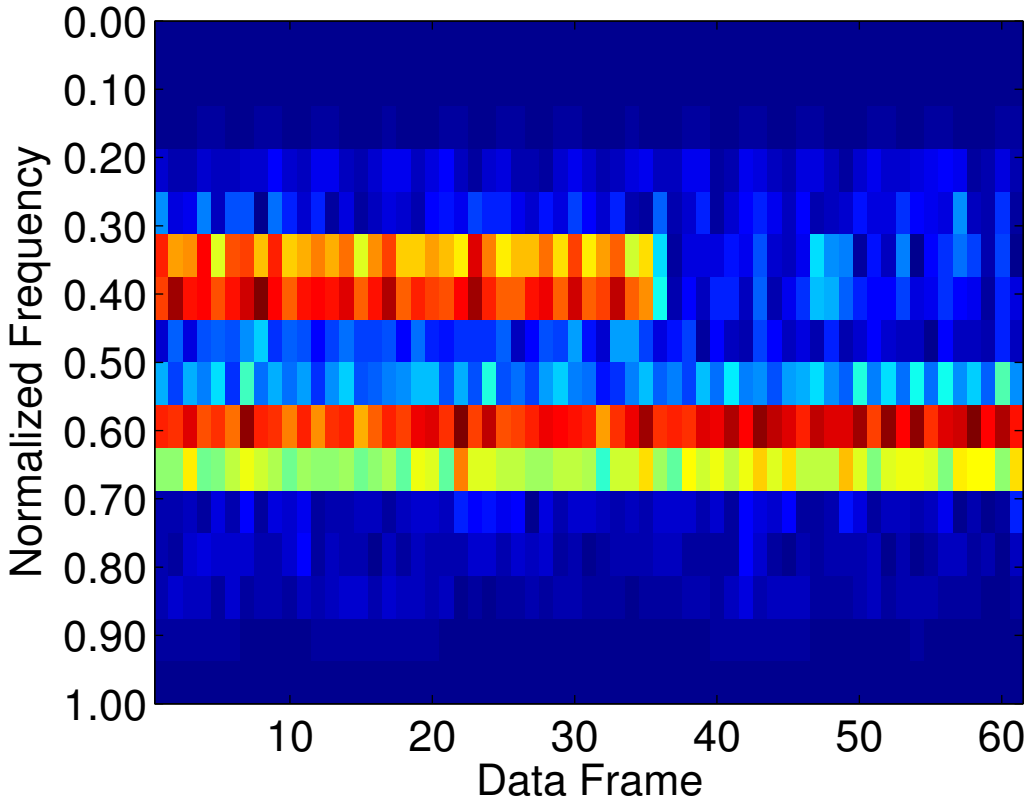


Figure 4.6: Magnitude Spectrogram of a short example section of RF spectral data containing two transmitters. The spectrogram is formed with length-16 Hann windows with a 50% overlap.

BPDN and SLPSR solutions were obtained using length-16 segments of this data with no overlap. The parameter ϵ was varied from ten to twenty percent of the total norm of each data segment. The parameter λ was varied as described above.

Figures 4.7 and 4.8 show the recovery of this data using SLPSR and BPDN for the same value for ϵ. The BPDN recovery is shown with a four-times oversampled dictionary, and the SLPSR solution for a dictionary without oversampling. Both solution methods seem to produce sensible estimates

of the transmitters, but the SLPSR solution uses fewer elements per transmitter. In this simple example, it is possible to estimate the properties of the transmitters using long data windows and quantitatively compare the recovery using the two methods.

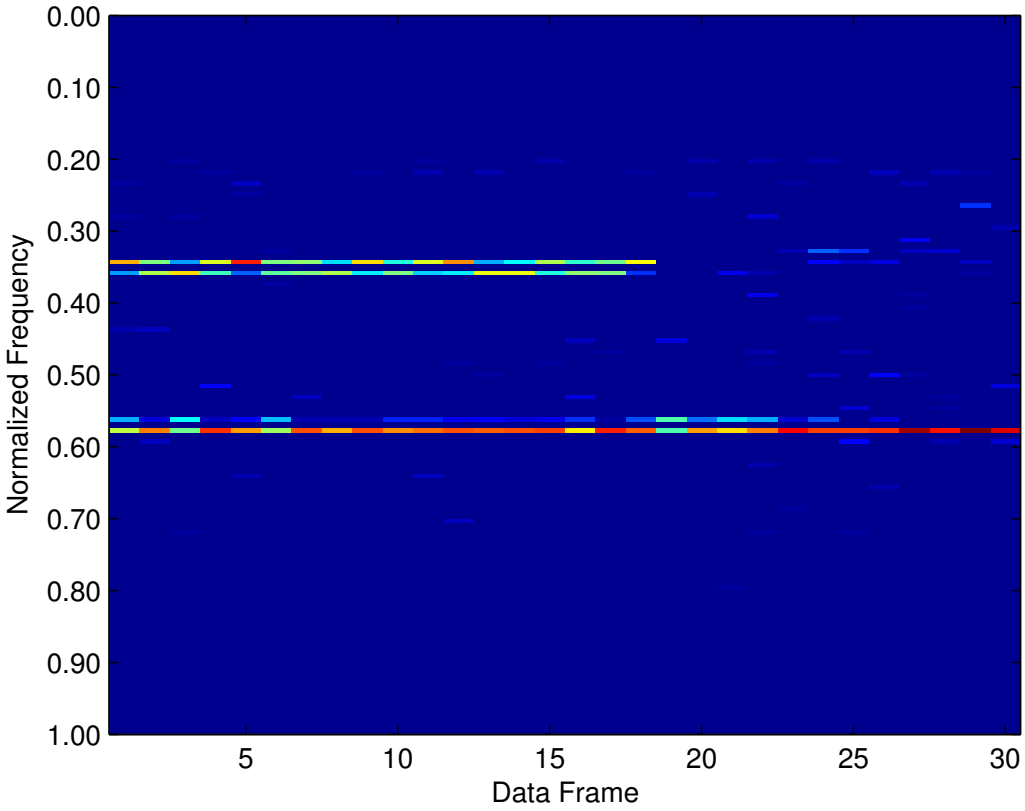
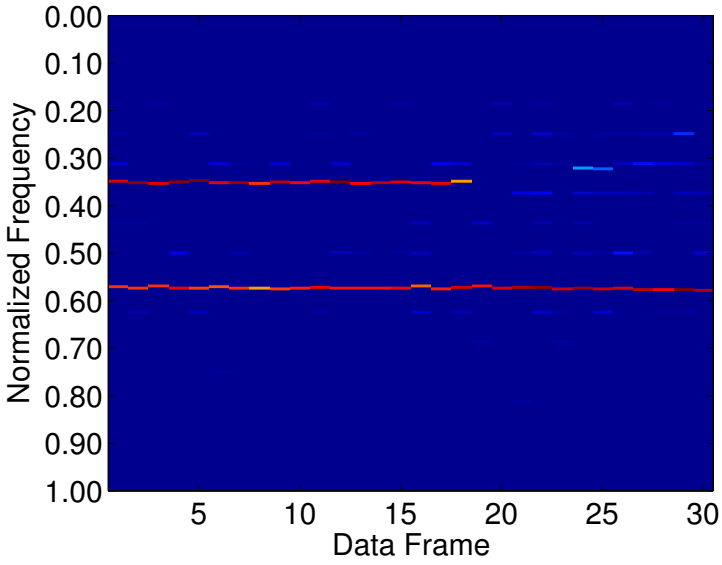


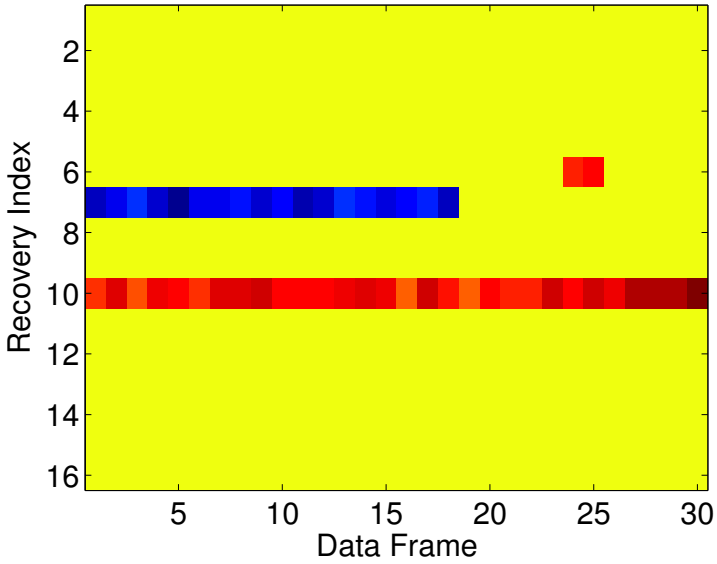
Figure 4.7: Example sparse recovery of a short segment of RF data with BPDN. Reconstructions are done with length-16, non-overlapping segments of data using a four-times oversampled dictionary. This corresponds exactly to the time domain data used to generate the spectrogram in Figure 4.6. The BPDN seems to provide a reasonable sparse recovery of this data, however several frequency elements are used to describe each transmitter.

For this case, the transmitter center frequencies were estimated using the peaks of the DFT of the signal taken over the entire window over which both transmitters were active. These estimates were then taken as the true frequency parameters for the length-16 reconstructions. The frame in which one transmitter appears to have turned off was omitted from the comparison, giving twenty-nine frames.

Using the SLPSR and BPDN algorithms with the same parameters used



(a) Coefficient Magnitude of sparse recovery using SLPSR, no oversampling



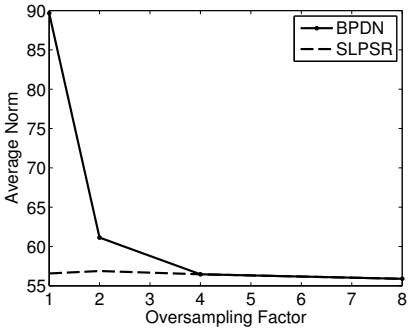
(b) Parameter perturbations recovered using SLPSR, no oversampling

Figure 4.8: Example sparse recovery of a short segment of Radio Frequency data with the SLPSR technique. Reconstructions are done with length-16, non-overlapping segments of data. This corresponds exactly to the time domain data used to generate the spectrogram in Figure 4.6. Magnitude plots are generated by plotting the recovered coefficient magnitude at each recovered parameter value. Parameter perturbations are shown for each recovery index.

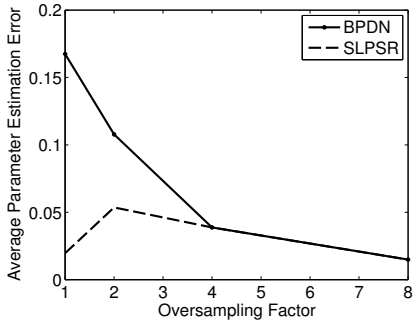
to generate Figures 4.7 and 4.8, several metrics were calculated and averaged over twenty-nine frames. This was repeated for different oversampling factors ranging from one to eight. First was the ℓ_1 norm of the recovered sparse coefficients $\mathbf{x}$. Second was the parameter estimation error from the transmitter frequencies. This was found by taking the closest parameter corresponding to a nonzero element of $\mathbf{x}$ and computing the absolute difference to the frequency parameters derived from the data. Finally, the number of elements of $\mathbf{x}$ above a threshold of 0.01 was computed for each frame. These were then averaged over all frames and displayed as a function of the oversampling factor.

Figure 4.9 compares the recovery of this segment of data with BPDN and SLPSR. For higher oversampling factors, the BPDN and SLPSR solutions coincide; the SLPSR solution does not differ significantly from the BPDN solution. What is more interesting is the behavior at lower oversampling factors. The SLPSR solution has almost the same average ℓ_1 norm for all oversampling factors. The sparsest solution of all conditions tested corresponds to the SLPSR solution with no oversampling, using approximately 1.5 fewer elements on average than the BPDN solution with eight-times oversampling. This configuration also produces parameter estimation error on par with the oversampled BPDN solution. The SLPSR solution is much less dependent on the amount of oversampling than the BPDN solution and in fact produces sparser solutions with no frequency oversampling.

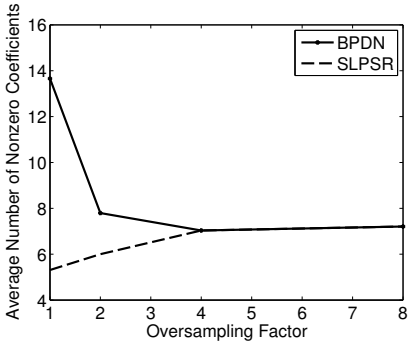
Overall, sparse recovery techniques are a reasonable approach to represent frequency-sparse RF data. Both the BPDN approach and the SLPSR solutions produce sharp estimates of the parameters peaks. True recovery error will still, of course, be bound by the statistical limits of the problem. The proposed SLPSR technique seems to offer two main advantages over the BPDN sparse recovery approach. First, the SLPSR recovery does not require a high oversampling factor in the initial parameter sampling. Second, SLPSR recovery with coarse parameter sampling produces sparser solutions while still providing recovery error on par with the oversampled BPDN recovery. These reconstruction results demonstrate the recovery of real-world sparse signals and parameter perturbations using a parameterized measurement model. Moreover, this model can be extended to better reconstruct the signal when the assumption of frequency-sparse signals is violated by the time dynamics of the signal.



(a) Average recovered ℓ_1 Norm of BPDN and SLPSR recovery



(b) Average Parameter estimation error of BPDN and SLPSR recovery



(c) Average number of nonzero components in BPDN and SLPSR recovery

Figure 4.9: Quantitative comparison of the recovered ℓ_1 norm, parameter estimation error, and number of nonzero components using SLPSR and BPDN on frequency-sparse RF data. The reconstructions were done with non-overlapping, length-16 segments of the data shown in Figure 4.6. The results are plotted against the oversampling factor used to generate the initial parameter sampling.

4.2 Improving Modeling Using Transition Elements

In the real-world RF data studied in this section, the signal is block-wise frequency-sparse. The transmitters, however, are broadcasting Morse code, which necessitates the transmitter turning on and off. These transition periods are not frequency sparse. To create a truly sparse representation of this RF data, these on-transition and off-transition events must be modeled as dictionary elements.

This observation raises a much more general problem for sparse recovery in real signal processing applications. Many signals are not frequency-sparse,

but composed of complex time-frequency events. This section demonstrates how a sparse recovery problem can be formulated by generating a parameterized measurement model for these events. Although different parameterized models will be required for different applications, this technique has impact in a wide set of applications including radar, acoustics, and biomedical signal processing.

In the example RF data, it appears that only on-transition and off-transition events are of interest. This approach, however, could be generalized to other time-frequency events.

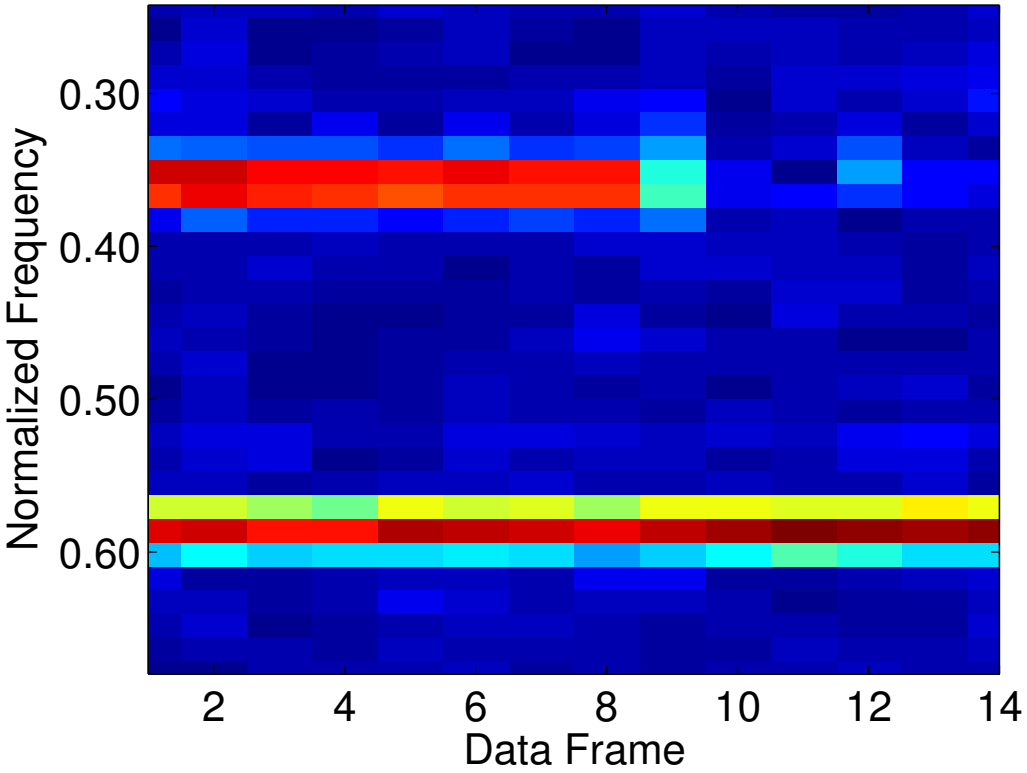


Figure 4.10: DFT magnitude spectrogram using the same segment of data in Figure 4.6 with length-64 Hann windows and 50% overlap. A ringing effect is noticeable when the transmitter turns off. This effect is not frequency-sparse.

Figure 4.10 shows the effect of transitions for a simple segment of data containing two transmitters. This figure shows the magnitude spectrogram for a length-64 window with 50% overlap. One of the transmitters turns off partway through this data. There is a noticeable ringing effect here due to this time event. At this frame, the data is not a frequency-sparse signal as

defined in Section 3.2. A new formulation is needed to capture this dynamic.

To apply the SLPSR algorithm to such events, a differentiable, parameterized model is required. By choosing a sigmoid ramp function for this data, it is possible to define a differentiable element representing on and off frequency events. This ramp function transitions from zero to one around a center frequency, with the ramp smoothness controlled by a slope parameter. The sigmoid ramp is parameterized by the transition time t_i, slope parameter k_i, and frequency ω_i. Dictionary elements modeling transitions from the off to on state (on-transition) can be defined as

$$\mathbf{a}_{on}(\omega_i, k_i, t_i) = \begin{bmatrix} \frac{1}{\sqrt{M}} * 1 * \frac{1}{1+\exp(-k_i*(0-t_i))} \\ \frac{1}{\sqrt{M}} * \exp(\imath * \omega_i * 1) * \frac{1}{1+\exp(-k_i*(1-t_i))} \\ \frac{1}{\sqrt{M}} * \exp(\imath * \omega_i * 2) * \frac{1}{1+\exp(-k_i*(2-t_i))} \\ \cdots \\ \frac{1}{\sqrt{M}} * \exp(\imath * \omega_i * (M-1)) * \frac{1}{1+\exp(-k_i*((M-1)-t_i))} \end{bmatrix} \tag{4.1}$$

where M is the length of the measurement vector. Similarly, transitions from the on to off state (off-transition) can be modeled as

$$\mathbf{a}_{off}(\omega_i, k_i, t_i) = \begin{bmatrix} \frac{1}{\sqrt{M}} * 1 * \frac{1}{1+\exp(k_i*(0-t_i))} \\ \frac{1}{\sqrt{M}} * \exp(\imath * \omega_i * 1) * \frac{1}{1+\exp(-k_i*(1-t_i))} \\ \frac{1}{\sqrt{M}} * \exp(\imath * \omega_i * 2) * \frac{1}{1+\exp(k_i*(2-t_i))} \\ \cdots \\ \frac{1}{\sqrt{M}} * \exp(\imath * \omega_i * (M-1)) * \frac{1}{1+\exp(k_i*((M-1)-t_i))} \end{bmatrix} \tag{4.2}$$

These dictionary events can be added to augment the signal dictionary defined in Section 3.2. An initial sampling of center times and slope parameters must be chosen. This generates an augmented dictionary capable of capturing transition events.

To apply the SLPSR algorithm, one would like to recover a set of parameter perturbations to correct for the initial sampling. This requires the definition of the Jacobian for this problem. Defining the Jacobian requires the definition of several functions capturing how the model changes with the

new parameters. For on-transition events, define $\partial \mathbf{a}_{\omega_i,on}$ as

$$\partial \mathbf{a}_{\omega_i,on}(\omega_i, k_i, t_i) = \begin{bmatrix} 0 \\ \frac{\imath}{\sqrt{M}} \exp(\imath * \omega_i * 1) \frac{1}{1+\exp(-k_i*(1-t_i))} \\ \frac{2\imath}{\sqrt{M}} \exp(\imath * \omega_i * 2) \frac{1}{1+\exp(-k_i*(2-t_i))} \\ \cdots \\ \frac{(M-1)\imath}{\sqrt{M}} \exp(\imath * \omega_i * (M-1)) \frac{1}{1+\exp(-k_i*((M-1)-t_i))} \end{bmatrix} \tag{4.3}$$

define $\partial \mathbf{a}_{k_i,on}$ as

$$\partial \mathbf{a}_{k_i,on}(\omega_i, k_i, t_i) = \begin{bmatrix} \frac{1}{\sqrt{M}} * 1 * \frac{(0-t_i)\exp(-k_i(0-t_i))}{(1+\exp(-k_i(0-t_i)))^2} \\ \frac{1}{\sqrt{M}} \exp(\imath * \omega_i * 1) \frac{(1-t_i)\exp(-k_i(1-t_i))}{(1+\exp(-k_i(1-t_i)))^2} \\ \frac{1}{\sqrt{M}} \exp(\imath * \omega_i * 2) \frac{(2-t_i)\exp(-k_i(2-t_i))}{(1+\exp(-k_i(2-t_i)))^2} \\ \cdots \\ \frac{\exp(\imath*\omega_i*(M-1))}{\sqrt{M}} \frac{((M-1)-t_i)\exp(-k_i((M-1)-t_i))}{(1+\exp(-k_i((M-1)-t_i)))^2} \end{bmatrix} \tag{4.4}$$

and define $\partial \mathbf{a}_{t_i,on}$ as

$$\partial \mathbf{a}_{t_i,on}(\omega_i, k_i, t_i) = \begin{bmatrix} \frac{1}{\sqrt{M}} * 1 * \frac{(-k_i)\exp(-k_i(0-t_i))}{(1+\exp(-k_i(0-t_i)))^2} \\ \frac{1}{\sqrt{M}} \exp(\imath * \omega_i * 1) \frac{(-k_i)\exp(-k_i(1-t_i))}{(1+\exp(-k_i(1-t_i)))^2} \\ \frac{1}{\sqrt{M}} \exp(\imath * \omega_i * 2) \frac{(-k_i)\exp(-k_i(2-t_i))}{(1+\exp(-k_i(2-t_i)))^2} \\ \cdots \\ \frac{\exp(\imath*\omega_i*(M-1))}{\sqrt{M}} \frac{(-k_i)\exp(-k_i((M-1)-t_i))}{(1+\exp(-k_i((M-1)-t_i)))^2} \end{bmatrix} \tag{4.5}$$

A similar set of equations can be defined for the off-transition events. These functions can be used to form an augmented Jacobian, using a similar method as with the frequency-sparse signals. For each parameter, the column of the Jacobian is summed over all columns of the dictionary in which the parameter appears. Now this augmented model can be used to recover a wider range of signals than possible with the frequency-sparse dictionary.

The SLPSR solution using only the frequency-sparse dictionary has significant error during transition elements. Figure 4.11 shows the recovered SLPSR magnitudes. This recovery uses a length-64 window and a two-times oversampled dictionary. At Frame 5 there is significant distortion due to the transition event. This frame is shown in Figure 4.12. There is clear distortion around the frequency that transitions from on to off. The augmented dictionary may be able to more accurately model this signal.

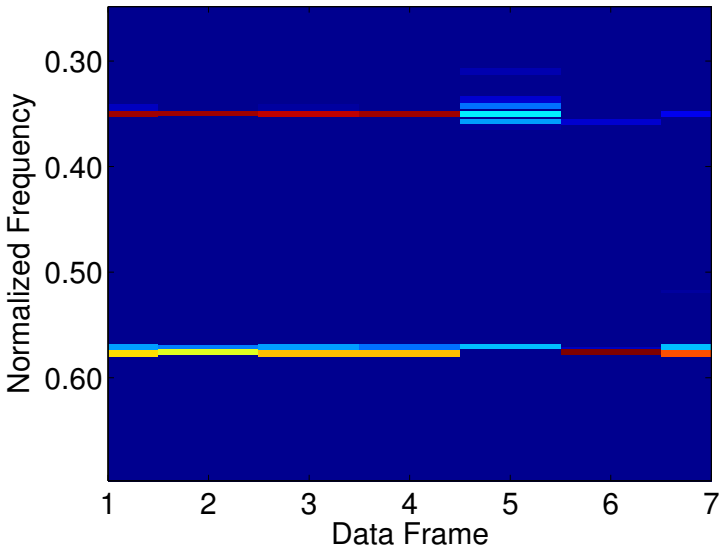


Figure 4.11: Magnitude of SLPSR recovery for length-64 segments of data with a two-times oversampled dictionary. This is the same data as in Figure 4.10. At Frame 5 there is significant distortion due to the transition event.

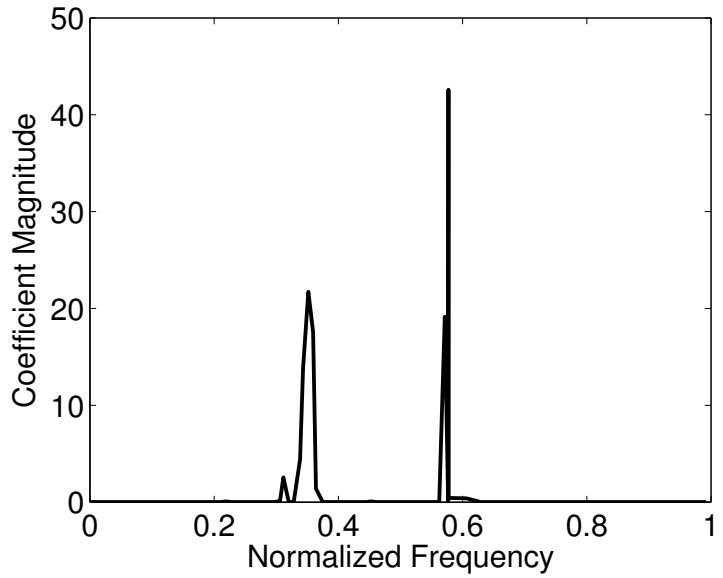
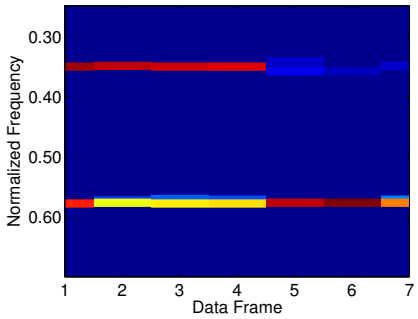


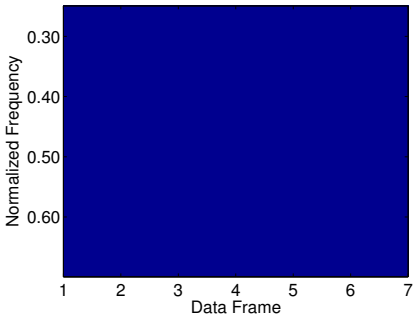
Figure 4.12: Recovered SLPSR coefficient magnitude for Frame 5 of the reconstructed signal in Figure 4.11. Around the transmitter which transitions from on to off there are significant distortion artifacts.

Now consider the case of SLPSR recovery using the augmented dictionary. For this example, the transition time was set at 32 samples (of a 64 sample window) for each frequency sample. The slope parameter was fixed at two for

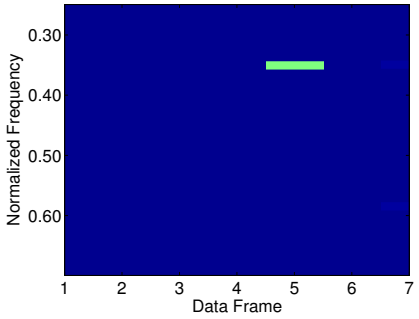
all frequencies. The frequency parameters were then perturbed to improve recovery. Except for the parameter λ, the same settings were used as for reconstruction using only the frequency-sparse dictionary. Figure 4.13 shows the reconstructions using this data. The transition elements are only nonzero at the transition point. For blocks where the data is frequency-sparse, the transition elements are not used.



(a) Magnitude of frequency-sparse coefficients, two-times oversampled



(b) Magnitude of on-transition event coefficients, two-times oversampled

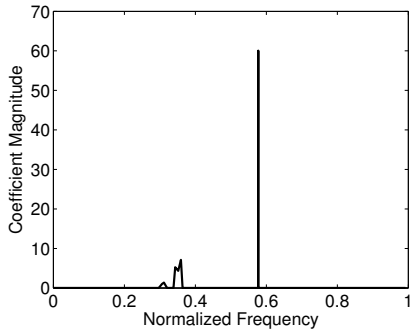


(c) Magnitude of off-transition event coefficients, two-times oversampled

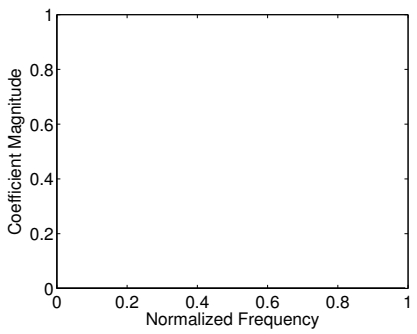
Figure 4.13: Recovery of data shown in Figure 4.10 using SLPSR with a two-times oversampled dictionary. This dictionary includes on-transition and off-transition events. Figure 4.13(a) shows the recovery for the frequency elements. Figure 4.13(b) shows the recovery for the on-transition elements. Figure 4.13(c) shows the recovery for the off-transition elements.

When there is a transition from on to off at Frame 5, the corresponding element is used to reconstruct the signal. The transition frame, Frame 5, is shown in Figure 4.14. Compared to the reconstruction with only the frequency-sparse dictionary, the reconstruction uses fewer large elements. There is still some distortion due to the transition event, but it is greatly reduced. The elements corresponding to the transition event clearly indicate

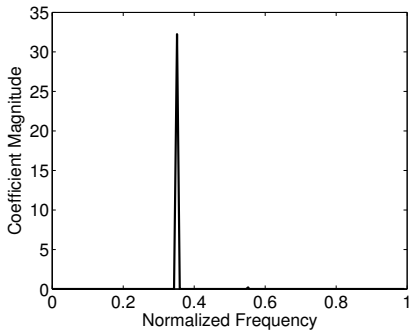
that the transition has been captured in the recovery.



(a) Coefficient magnitude of frequency-sparse elements, Frame 5



(b) Coefficient magnitude of on-transition elements, Frame 5



(c) Coefficient magnitude of off-transition elements, Frame 5

Figure 4.14: Recovery of Frame 5 shown in Figure 4.13 using SLPSR with a two-times oversampled dictionary. This dictionary includes on-transition and off-transition events. Figure 4.14(a) shows the recovery for the frequency-sparse elements. Figure 4.14(b) shows the recovery for the on-transition elements. Figure 4.14(c) shows the recovery for the off-transition elements.

Modeling the on-transition and off-transition events allows for the sparse recovery of the continuous-wave Morse code signals studied in this chapter. Capturing the transition events allows for sparse recovery even as transmitters show simple time dynamics. More importantly, the same methodology can be used to develop augmented dictionaries to match a wide range of parameterized time-frequency events. The SLPSR technique can then be used to recover sparse coefficients and parameter perturbations from these signals.

This chapter has demonstrated that joint recovery of parameter perturbations and sparse coefficients can improve reconstructions of real-world frequency-sparse data. The frequency-sparse model can also be augmented

to include transition events to capture the time dynamics of the signal. Overall, the proposed algorithm is able to recover sparser solutions than BPDN while using a coarsely sampled dictionary.

CHAPTER 5

DISCUSSION AND CONCLUSIONS

Recovery of signals from parameterized models is a common problem that arises in a diverse range of applications. This thesis has developed a novel sparse recovery formulation for parameterized signal models that aims to jointly recover a sparse signal and a sparse set of parameter perturbations. Given an initial sampling of the parameter space, this allows for the recovery of a sparse signal and the correction of the initial parameter sampling. The fundamental intuition is that the recovered signal is only truly sparse when the underlying signal parameters are represented in the signal dictionary.

In order to solve the proposed problem, the SLPSR algorithm can be used to find first-order critical points. As this recovery algorithm only guarantees convergence to a first-order critical point, the SLPSR solution will depend on the initialization of the sparse signal $\mathbf{x}$ and the initial parameter set. Choosing the initial parameter set will depend on the parameterized model used. For this thesis, the parameter set was chosen as an equally spaced grid over the possible range of parameters.

Using simulated data of frequency-sparse signals, it was demonstrated that the most accurate recovery of the unknown coefficients and parameters was for a coarse sampling of the parameter space using the SLPSR algorithm. With a coarse sampling, the proposed algorithm outperforms state-of-the-art approaches in the recovery of the unknown signal coefficients, even when other techniques use highly oversampled dictionaries. This suggests that, at least for the frequency-sparse model, better recovery is possible with a dictionary that samples the space less densely but contains less well-correlated elements. This is an interesting alternative to many current approaches, which simply rely on highly oversampling the parameter space.

A potential extension of this approach would be to consider model-based sparsity. Model-based approaches have been used to allow for sparse recovery when there is a relationship between the elements of the signal dictionary.

This was applied in [13] to allow for reconstruction of frequency-sparse signals. The model restricted recovery of any solution containing two dictionary elements that were too well correlated. The recovery guarantees of Model-based CS may be used to determine when recovery is possible in a given parameterized signal model.

The proposed technique was compared against the two most related state-of-the-art sparse recovery techniques, STLS and Perturbed CS solved with the AA-P-BPDN algorithm. The AA-P-BPDN is closely related to the proposed sparse recovery problem, but assumes a strictly linear model. The AA-P-BPDN algorithm also requires the parameter perturbations to be constrained by the ℓ_∞ norm. This linearized model can be implemented in the formulation proposed in this thesis. By taking $\mathbf{A}(\omega + \mathbf{d}\omega) = \mathbf{A} + \mathbf{B} * \Delta$ where Δ is a diagonal matrix with entries given by $\mathbf{d}\omega$, one can generate a linearized model for SLPSR. The STLS technique allows for an unstructured perturbation. This may be appropriate when the signal model is partially unknown or not differentiable. Choosing an appropriate method to compensate for unknown dictionaries will be highly problem dependent. Different parameterized measurement models may benefit more from one method over the others. This will need to be explored in the context of a specific model.

An issue that has not been addressed yet in this thesis is the computational complexity of the proposed algorithms. The BPDN algorithm has proven to be very popular partially because it can be efficiently solved using common convex programming solvers. Algorithms which account for errors in the signal dictionary, unfortunately, have a more complicated form. The STLS, AA-P-BPDN, and SLPSR algorithms all attempt to solve nonlinear, non-convex programming problems. All three of these methods remain tractable, however, and all require approximately the same computational complexity. This involves solving a standard convex programming problem followed by a constrained least-squares problem. This is then iterated many times. A true complexity analysis and runtime test of these algorithms is beyond the scope of this thesis.

Another key issue is the fundamental limits for sparse recovery of signals with parameterized measurement models. Estimation theory can provide restrictions on how well parameters can be estimated in the presence of noise, essentially providing lower-bounds to the variance of an estimator. For frequency-sparse signals, a resolution analysis would naturally follow.

Whether or not joint recovery of sparse signals and parameter perturbations can achieve these bounds remains to be seen.

5.1 Conclusions and Future Work

This thesis has explored the problem of applying sparse recovery techniques to signals with parameterized models. Parameterized measurement models arise in large number of important applications such as imaging and array processing. Improving reconstruction and recovery techniques could have major impact on areas such as medical imaging, radar, and sonar. In these applications, the desired signal is also often sparse, in the sense that very few elements of the signal model are required to construct the signal. In this case, sparse recovery techniques may provide important performance improvement.

In such applications, it is desirable to apply sparse recovery techniques, but the signal dictionary is not completely known. This is because the signal parameters are unknown a priori. Generally, a highly oversampled dictionary has been used for sparse recovery. Current approaches are limited in how to adapt to unknown parameters.

This thesis introduced a new problem formulation to jointly recover sparse signals and sparse parameter perturbations. The key intuition is that in order to recover the sparse signal, the correct parameters must also be recovered. A solution to this formulation can be found using an iterative algorithm that solves a series of linearized subproblems. This recovery algorithm converges to a first-order critical point of the problem.

This new algorithm produces a set of strictly feasible iterates that will never increase the ℓ_1 norm of the recovered solution past the ℓ_1 norm of the BPDN solution used for initialization. This recovery algorithm allows for a wide variety of parametrized models, giving more flexibility than previous sparse solution methods with perturbed sensing matrices.

This novel problem formulation was applied to the recovery of frequency-sparse signals with both simulated and real-world measurements. Compared to existing sparse recovery algorithms, the proposed algorithm was able to achieve similar recovery error while producing sparser solutions. Interestingly, the sparsest solutions with the proposed algorithm was for signal dic-

tionaries with a coarse sampling. The proposed method is an interesting alternative to attempting sparse recovery with highly oversampled dictionaries. Of the sparse recovery methods tested on frequency-sparse signals, only the proposed algorithm recovered the unknown signal weights and parameters given noiseless measurements. All other methods demonstrated error, even with 16-times oversampled dictionaries.

A major future direction will be applying the proposed method, along with other sparse recovery methods, to a wide range of parameterized models. Important applications arise in array processing problems, such as beamforming. Another set of applications is when a signal is composed of shifted versions of a template waveform, as in echolocation or neural data processing. These models can be easily parameterized and, in certain cases, may be represented as a sparse signal.

There are also several interesting theoretical directions indicated by these findings. One direction is the reconstruction of data in dynamic cases, where the signal parameters change over time. Here, it may be possible to derive iterative recovery algorithms where signal weights and parameters are constrained by the previous solutions.

Another interesting theoretical direction arises from the observation that the best recovery with the SLPSR algorithm occurs with coarse oversampling factors. These dictionaries have less well-correlated elements. There may be interesting implications and connections to the recovery guarantees in CS, which are dependent on the correlation between dictionary elements. In particular, the recent work studying Model-Based CS may provide important tools to study recovery of sparse signals with parameterized dictionaries.

A further connection to explore is between sparse recovery algorithms and traditional parameter estimation techniques. Comparing these techniques will be necessary in order to understand the limits of parameter recovery error and resolution of sparse recovery algorithms in parameterized models.

This thesis has investigated an important class of signals with parameterized measurement models arising in areas such as imaging, model-fitting, and array signal processing. Given a set of measurements, one would like to recover the unknown signal coefficients and parameters. Unfortunately, a problem of this form typically has infinitely many solutions. Sparsity has proven to be a powerful tool for regularizing such reconstructions; this involves searching for a solution with as few nonzero elements as possible.

However, typical sparse recovery algorithms require a fixed signal dictionary. In previous work, this dictionary has been found by highly oversampling the parameter space. This thesis proposed an alternative approach, recovery of sparse signals and a set of sparse parameter perturbations to correct an initial set of parameters. This formulation allows for the recovery of sparse signals and the corresponding parameters without highly oversampled dictionaries. Numerical experiments and reconstructions of real-world data confirm that recovery is possible with coarsely sampled dictionaries, and, in fact, the proposed method recovers sparser solutions than standard sparse recovery with highly oversampled signal dictionaries. This may improve sparse recovery in a wide variety of applications involving sparse signals with parameterized measurement models, ranging from spectrum monitoring to echolocation.

REFERENCES

[1] R. C. Aster, B. Borchers, and C. H. Thurber, *Parameter Estimation and Inverse Problems*. Academic Press, 2012.

[2] A. Neumaier, "Solving ill-conditioned and singular linear systems: A tutorial on regularization," *SIAM Review*, vol. 40, no. 3, pp. 636–666, 1998.

[3] S. M. Kay, *Fundamentals of Statistical Signal Processing, Volume I: Estimation Theory*. Prentice Hall, 1993.

[4] H. Krim and M. Viberg, "Two decades of array signal processing research: The parametric approach," *Signal Processing Magazine, IEEE*, vol. 13, no. 4, pp. 67–94, July 1996.

[5] E. Candés, "Compressive sampling," in *Proceedings of the International Congress of Mathemeticians*, Madrid, Spain, Aug. 2006, pp. 1433–1452.

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

[7] 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.

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

[9] D. Malioutov, M. Çetin, and A. S. Willsky, "A sparse signal reconstruction perspective for source localization with sensor arrays," *Signal Processing, IEEE Transactions on*, vol. 53, no. 8, pp. 3010–3022, 2005.

[10] J. M. Kim, O. K. Lee, and J. C. Ye, "Compressive music: Revisiting the link between compressive sensing and array signal processing," *Information Theory, IEEE Transactions on*, vol. 58, no. 1, pp. 278–301, Jan. 2012.

[11] P. T. Boufounosa, P. Smaragdis, and B. Rajc, "Joint sparsity models for wideband array processing," in *Proceedings of SPIE*, vol. 8138, 2011, p. 81380K.

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

[13] M. F. Duarte and R. G. Baraniuk, "Sectral compressive sensing," 2013. [Online]. Available: dsp.rice.edu/files/publications/

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

[15] S. Gleichman and Y. Eldar, "Blind compressed sensing," *Information Theory, IEEE Transactions on*, vol. 57, no. 10, pp. 6958–6975, 2011.

[16] M. Aharon, M. Elad, and A. Bruckstein, "K-svd: An algorithm for designing overcomplete dictionaries for sparse representation," *Signal Processing, IEEE Transactions on*, vol. 54, no. 11, pp. 4311–4322, 2006.

[17] H. Zhu, G. Giannakis, and G. Leus, "Weighted and structured sparse total least-squares for perturbed compressive sampling," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, May, pp. 3792–3795.

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

[19] Z. Yang, C. Zhang, and L. Xie, "Stable signal recovery in compressed sensing with a structured matrix perturbation," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, Mar. 2012, pp. 2737 –2740.

[20] Z. Yang, C. Zhang, and L. Xie, "Robustly stable signal recovery in compressed sensing with structured matrix perturbation," *Signal Processing, IEEE Transactions on*, vol. 60, no. 9, pp. 4658–4671, 2012.

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

[22] A. Conn, N. Gould, and P. Toint, *Trust Region Methods*. Society for Industrial Mathematics, 1987, vol. 1.

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

[24] R. T. Rockafellar, "Directionally Lipschitzian functions and subdifferential calculus," *Proceedings of the London Mathematical Society*, vol. s3-39, no. 2, pp. 331–355, 1979.

[25] A. D. Ioffe, "Necessary conditions in nonsmooth optimization," *Mathematics of Operations Research*, vol. 9, no. 2, pp. pp. 159–189, 1984.

[26] J. Eriksson, E. Ollila, and V. Koivunen, "Essential statistics and tools for complex random variables," *Signal Processing, IEEE Transactions on*, vol. 58, no. 10, pp. 5400–5408, 2010.

[27] J. F. Sturm, "Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones," *Optimization Methods and Software*, vol. 11, no. 1-4, pp. 625–653, 1999.

[28] F. Alizadeh and D. Goldfarb, "Second-order cone programming," *Mathematical Programming*, vol. 95, pp. 3–51, 2003.

[29] D. G. Luenberger and Y. Ye, *Linear and Nonlinear Programming*. Springer, 2008.

[30] B. Guta, "Subgradient optimization methods in integer programming with an application to a radiation therapy problem," Ph.D. dissertation, Teknische Universität Kaiserlautern, Kaiserlautern, Germany, May 2003.

[31] R. Penrose, "A generalized inverse for matrices," in *Proceedings of the Cambridge Philosophy Society*, vol. 51, no. 406-413. Cambridge University Press, 1955, p. C655.

[32] S. Ji, Y. Xue, and L. Carin, "Bayesian compressive sensing," *Signal Processing, IEEE Transactions on*, vol. 56, no. 6, pp. 2346–2356, 2008.

[33] Y. Zhang, J. Yang, and W. Yin, "Yall1: Your algorithms for l1," 2011. [Online]. Available: yall1.blogs.rice.edu