MODEL DESIGN FOR ALGORITHMIC EFFICIENCY IN ELECTROMAGNETIC SENSING

A Thesis
Presented to
The Academic Faculty

by

Kyle Ralph Krueger

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Electrical and Computer Engineering

Georgia Institute of Technology
December 2013

MODEL DESIGN FOR ALGORITHMIC EFFICIENCY IN ELECTROMAGNETIC SENSING

Approved by:

Dr. Justin Romberg,
Committee Chair
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. James McClellan, Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Waymond Scott Jr., Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Aaron Lanterman
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Ghassan AlRegib
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Michael Lacey
School of Mathematics
Georgia Institute of Technology

Date Approved: 7 November 2013

This dissertation is dedicated to

Stephanie, Annabelle, and the loving memory of our baby in Heaven.

Thank you for your love and support.

ACKNOWLEDGEMENTS

At this time I would like to thank the people that provided me with everything necessary to achieve this monumental goal in my life. Without them, this would not have been possible.

The first person I would like to thank is my wife, Stephanie. She has been with me since my second year of high school and has seen me go through wanting to take a career in music at the University of Illinois, all the way up through pursuing my Ph.D. at Georgia Institute of Technology. She moved out of state and supported me through all the long nights and the odd hours. She encouraged me when things looked the darkest, and that is something I could never fully express my appreciation for. But most importantly, she gave me my two most important titles; she made me a husband and father.

The next person I would like to thank will not be able to read this for a few more years, my daughter Annabelle Rae. She reminds me every single day what is important and has provided me the motivation to work as hard as I can to make life for her the best I possibly can. I hope that this experience has laid the foundation to allow me to provide her the opportunity to do anything.

I also must express my sincerest gratitude to my family, starting with my parents, Scott and Kim Krueger. Supporting me through sports, music, schooling, and teaching me the importance of being first and foremost a good person. Always having them by my side regardless of the circumstances provided me with the courage to try new things regardless of the probability of failure. Without their unwavering support, this process would have been much more difficult. To my brothers and best friends, Ryan and Taylor Krueger, whose own pursuits and friendship have been inspiring

through the long journey. I cannot wait to spend more time with them now that this process is over! To my in-laws: Megan Krueger; Ray and Karen Simar; and Jason and Jennifer Schwartz.

I would also like to thank the two most influential people in this process, my advisors, Dr. James McClellan and Dr. Waymond Scott, Jr. The excitement they have for their research and their students continually kept me excited about the project. Their advising strategy of keeping in constant contact but allowing me to find my own path was extremely beneficial to the success of my Ph.D. I always felt as though I could come to them with any problems and knew that I would be provided nonjudgmental advice. I hope that we are able to keep in contact in the future.

Finally, I would like to thank some of my friends and colleagues. Troy Ruths, who has been a friend of mine since elementary school, and has gone through so much with me throughout the years. Band in middle school, co-captains of the basketball team in high school, and completing his Ph.D. at the same time as me in computer science at Rice University. I have always been inspired by your intelligence and work ethic, and you have always made me a better person. My Georgia Tech colleagues: Chris Turnes, Aurèle Balavoine, Chenchi Luo, Wei-Hsin Mu, Peter Tuuk, Carson Wick, Lingchen Zhu, Mario Malavè, Yoni Gabbay, William Mantzel, Ali Cafer Gürbüz, and Michael McFadden.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

The concern with non-military related landmine casualties still exists today, even though there is only one government, Syria, that has been recorded as actively using antipersonnel mines in 2012 [2]. In 2011, there were over 4000 reported casualties, over 11 deaths every day. This number is a third of what it was ten years ago, but has remained rather steady for the last three years because of increased conflict in countries such as Libya, Pakistan, and the Sudan [2]. There has been a large amount of research trying to identify, locate, and eliminate landmines in an effective and safe manner.

There are many difficulties with the task of eliminating the landmine threat. Mines are generally buried and concealed underground, so sensors must be used to detect their characteristics without being able to see them. A few devices that have been developed to help retrieve this information are ground-penetrating radars (GPR) and electromagnetic induction (EMI) sensors. Research has shown that GPR is an effective detector of subsurface targets, and that EMI sensors are able to both help in finding the location of the objects, and also in the discrimination between landmines and other subterranean targets [3, 4]. Parameter detection of unseen objects through a sensing medium is an important research topic, and is not limited to only the landmine-detection problem.

Many algorithms throughout the years have been developed in an attempt to detect these unknown parameters through the use of dictionary matching. Dictionary matching is a technique where a large database is created in order to match measured data against known-response data to determine the parameters of the measured data.

The simplest way to analogize this is with a word dictionary. A word dictionary is filled with the definitions of many words. The measured data in an environment would be any written or spoken material. If someone did not know what a word meant, they would go through a dictionary and match the word that they saw out in the world to the one in the dictionary and receive the definition. There are three main steps to this process: build a dictionary, create a book containing all the possible known words that could show up; collect the data, find an unknown word; and find a match for the data in the dictionary, look up the word and read the definition. In the landmine detection problem the measured data would be a sensor response and the dictionary would be made up of enumerating an approximated model of the expected response of the system given some particular target parameters. The matching part can be done with a number of different techniques such as backprojection (BP), orthogonal matching pursuit (OMP), compressive sensing (CS), and many others. The general detection flow used in dictionary matching algorithms can be seen in Figure 1. These algorithms involve creating a physical model for what the user would anticipate to receive if the parameters of the unseen target were known. For the model to be as general and as accurate as possible, it can require an extremely large amount of computer storage and can be computationally inefficient to apply. These computational constraints make most of these algorithms in their most basic forms impractical for real-time applications.

The objective of this work is to exploit special properties in the physical models to reduce the computational complexity of the landmine parameter-estimation problem. Reducing the computational complexity enables these problems to become more practical while preserving, or in some cases increasing, the effectiveness of the detection.

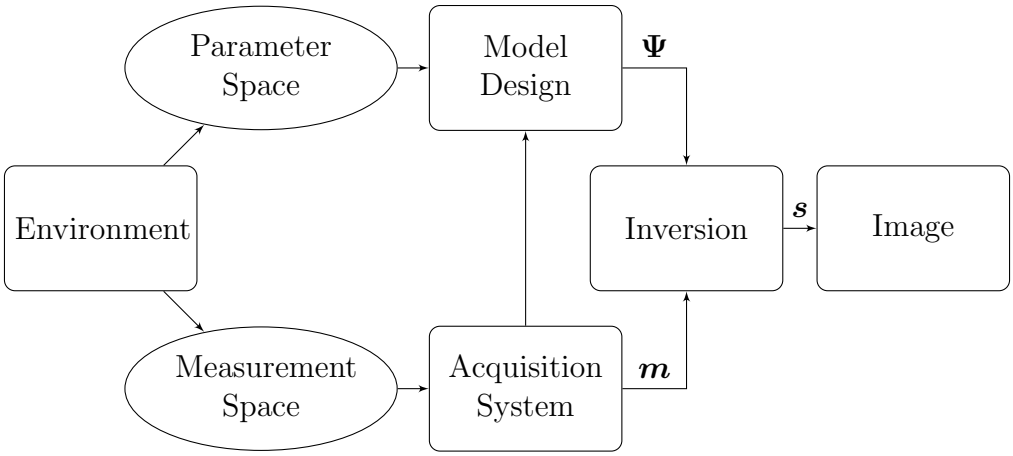


Figure 1: Detection flow for a GPR system with model-based inversion.

1.1 Data-acquisition Systems

There are two primary data-acquisition systems that will be studied in this research. The two systems are a ground-penetrating radar (GPR) system, and an electromagnetic induction (EMI) system. A method of acquiring data with the GPR called synthetic-aperture radar (SAR) will also be discussed.

1.1.1 Ground-penetrating radar

GPR systems have been shown to be an effective tool for imaging subterranean targets such as landmines [5–7]. GPR has been in use since the early 1970's and was designed originally to be used in landmine detection [8]. The traditional GPR system sends out an electromagnetic pulse from the transmitter. The pulse reflects from a target, and the reflection is detected by the receiver. The time delay between transmitting the signal and receiving the reflection is recorded to determine the target location [7]. The GPR can detect most targets since it can sense differences between the dielectric, magnetic, or conductive properties between the target and the soils. In the case where the transmitter and receiver are scanned together, the point-target model is

$$r(t, \boldsymbol{l}_s; \boldsymbol{l}_t) = \frac{\sigma}{S(d(\boldsymbol{l}_s, \boldsymbol{l}_t))} \delta(t - 2\frac{d(\boldsymbol{l}_s, \boldsymbol{l}_t)}{v}), \tag{1}$$

when a pulse, $\delta(t)$, is sent from the transmitter at location $\boldsymbol{l}_s$, and reflected off a target at location $\boldsymbol{l}_t$, with a reflection coefficient, σ. The function d is the euclidean distance of the electromagnetic signal path from the source to the target, S is the spreading function, and v is the velocity of the EM wave through a medium.

Another way of collecting data with a GPR is to use a stepped-frequency GPR (SFGPR). Instead of sending out a short pulse, the SFGPR successively sends out many short sinusoids at different frequencies over a specific bandwidth and calculates the phase difference at the receiver to measure distance and then locate the target. An advantage to using a SFGPR over the time impulse method is that it allows for a much wider bandwidth to be covered because of the narrow instantaneous bandwidth [5]. The narrow instantaneous bandwidth also provides resistance to noise and interference [9]. A disadvantage to using SFGPR is that it can take a long time to acquire data and the gain cannot be increased with increasing range. The target response of a SFGPR,

$$R(\omega, \boldsymbol{l}_s; \boldsymbol{l}_t) = \frac{\sigma}{S(d(\boldsymbol{l}_s, \boldsymbol{l}_t))} e^{j\omega 2\frac{d(\boldsymbol{l}_s, \boldsymbol{l}_t)}{v}}, \tag{2}$$

has a phase shift instead of a time delay.

However, collecting data like a traditional radar is not very effective for the GPR case. Traditional radar is generally looking for targets at an along-track resolution on the order of meters or even kilometers. When trying to locate very shallow targets, the desired resolution is on the order of centimeters. For a traditional radar to achieve these fine along-track resolutions, the system requires either an extremely short wavelength, or a very large physical radar. A solution to this problem is to use a synthetic-aperture technique. Synthetic-aperture radar (SAR) has been shown to be effective in GPR applications [3, 6, 10–13].

SAR is a data-acquisition technique that moves the antenna to different locations along a path and coherently adds the received samples together to create a higher resolution image in the along-track dimension [14–16]. The trade-off in this system is

increased along-track resolution for increased computational complexity. The system does not make independent decisions at each scan position, it uses the collection of all scans to make the best image. An example of a typical 2D scan grid for a 3D GPR imaging problem can be seen in Figure 2.

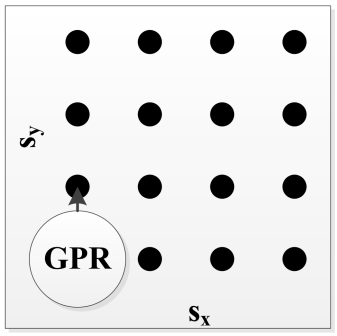


Figure 2: 2D scan grid for GPR data acquisition.

SFGPR can be hindered by the long data-acquisition times, especially for array systems that have several transmitting antennas because the transmit antennas are not used simultaneously. For instance, consider a vehicle mounted GPR that needs to acquire 400 frequencies while the vehicle advances 2 cm to obtain a 2 cm aperture spacing for the synthetic aperture. If it is assumed that the GPR has five transmit antennas and requires 100 μs to acquire each frequency, the vehicle could then travel at a maximum speed of approximately

$$\begin{aligned} max\ speed &= \frac{aperture\ spacing}{frequency\ time \times num\ TX\ antennas \times num\ frequencies} \\ &= \frac{0.02}{0.0001 \times 5 \times 400} \\ &= 0.1\text{m/s}, \end{aligned} \tag{3}$$

which is impractically slow for many applications. Another example comes from the Geneva International Centre for Humanitarian Demining (GICHD) that has published a list of specifications for state of the art GPR landmine detectors [17]. The hand-held detectors in the GICHD analysis have an optimum sweep speed between 0.2

and 1 m/s which is the speed at which the device can be moved spatially to achieve optimal detection accuracy. While a sweep speed of 1 m/s is probably adequate for a hand-held GPR, a sweep speed of 0.2 m/s would be uncomfortably slow to use.

Data-acquisition speed is more critical for vehicle mounted GPR systems as they are used to search larger areas and faster vehicle speed is desirable. In the GICHD equipment catalog, the vehicles with mounted GPRs can travel at speeds between 0.2 and 2 m/s which are much slower than desired. The data-acquisition time is not as problematic for TPGPR, but it is still an issue because multiple shots are still needed.

1.1.2 Electromagnetic induction sensor

EMI sensors have been shown to be effective in localizing and discriminating subterranean targets [18–23]. The focus in this thesis is on frequency-domain wideband EMI systems which are scanned over a target. The frequency-domain response of the target depends on the orientation, geometry, and material of the target. The frequency-domain response can be used as a "fingerprint" for that type of target [18]. A simple dipole model is adequate for many targets of interest and can be written in the frequency domain as

$$r(\omega, \boldsymbol{l}_s; \boldsymbol{l}_t, \boldsymbol{o}_t) = C\boldsymbol{g}^H(\boldsymbol{l}_s - \boldsymbol{l}_t)\boldsymbol{R}^H(\boldsymbol{o}_t)\boldsymbol{M}(\omega)\boldsymbol{R}(\boldsymbol{o}_t)\boldsymbol{f}(\boldsymbol{l}_s - \boldsymbol{l}_t). \tag{4}$$

ω is the frequency, C is a constant defined by characteristics of the transmit and receive coils, $\mathbf{g}(\boldsymbol{l}_s - \boldsymbol{l}_t)$ and $\mathbf{f}(\boldsymbol{l}_s - \boldsymbol{l}_t)$ are 3×1 vectors containing the spatial components of the magnetic field on each receiver coil and transmitter coil respectively. A reciprocity argument is used so the fields are those when both the receiver and transmitter coils are used as sources. $\boldsymbol{R}(\boldsymbol{o}_t)$ is a pure rotation matrix rotating by a

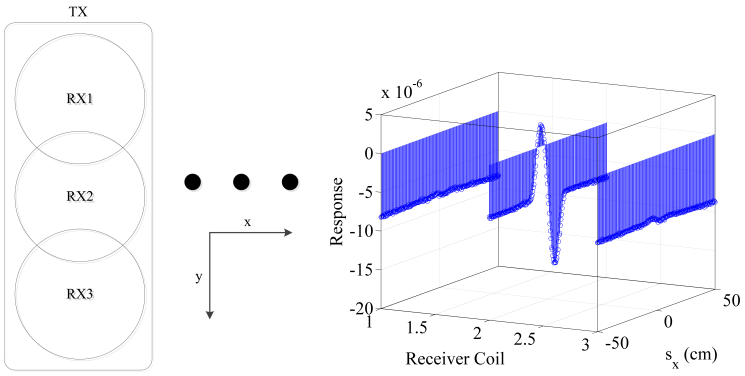
three-dimensional (3D) angle $\boldsymbol{o}_t = (\alpha_t, \beta_t, \gamma_t)$ and

$$\begin{aligned} \boldsymbol{M}(\omega) &= D_0\boldsymbol{\Lambda}_0 - \sum_{k=1}^{N_\zeta} D_k \left(\frac{j\omega/\zeta_k}{1 + j\omega/\zeta_k} \right) \boldsymbol{\Lambda}_k \\ &= \sum_{k=0}^{N_\zeta} D_k p(\omega, \zeta_k) \boldsymbol{\Lambda}_k, \end{aligned} \tag{5}$$

is the magnetic polarizability of a specific target in terms of its relaxation frequencies, ζ_k, and the 3×3 real, symmetric, 3×3 tensor, $\boldsymbol{\Lambda}_k$ [1, 24–26].

The EMI system is not a radar, so the coherent sum strategy of SAR does not directly apply here. Nonetheless, collecting the data in a similar way along a path and storing it, provides an increased number of looks at the target. This type of acquisition allows for more information to be collected, and can help give more accurate parameter estimation. It is also a key component to a multiple-measurement technique used to accurately estimate the DSRF of the target [27].

The specific data-acquisition system that is used in all the experiments in this work is set up with a single transmit coil, and three receive coils. The sensor position, $\boldsymbol{l}_s = (x_s, y_c, 0)$, is scanned in the along-track dimension but looks to image targets in all three spatial coordinates, $\boldsymbol{l}_t = (x_t, y_t, z_t)$. The coil location y_c is the y offset for each receiver coil. A visualization of the sampling pattern can be seen in Figure 3(a). Having the three receive coils lined up in the cross-track y dimension helps with extracting the target location. The data is collected at 21 wideband frequencies for each downrange location. A raw set of along-track measurements at each receive coil and at one frequency can be seen in Figure 3(b). The raw measurements have a significant DC offset that must be eliminated before the detection stage. Typically the DC offset is eliminated with a zero-mean downrange filter.



(a) (b)

Figure 3: (a) Sampling pattern (b) and raw measurements.

1.2 Inversion Algorithms

There are many different types of inversion algorithms that are used to extract parameters from measured data. The inversion algorithms discussed in this thesis rely on a model that is enumerated into a dictionary. The rest of this section will be split into five subsections. The first, subsection 1.2.1, will show how to create a dictionary from a nonlinear model, the next four subsection will discuss the basics of four different inversion algorithms that can use dictionary matching: BP, OMP, CS, and semidefinite programming (SDP).

1.2.1 Dictionary creation

The creation of the dictionary is a simple, yet important step to all of these inversion techniques. The idea is to create a matrix that will contain all of the expected responses of the system for all possible variable settings of the model. The variables are in two sets: unknown, or target parameter variables: location, orientation, and type are a few; and known, or measurement variables: sensor positions, frequencies, and time for example. The dictionary matrix column space consists of enumerating the target parameters and the row space consists of enumerating the measurement variables. For instance, in (4), ω and $\boldsymbol{l}_s$ are measurement variables, because they

are the known support of the response and would populate the rows of the matrix. Each possible target parameter would add an additional column to the matrix. For example, if there are ten frequencies, $N_\omega = 10$; five scan positions, $N_{\boldsymbol{l}_s} = 5$; four possible target locations, $N_{\boldsymbol{l}_t} = 4$; and six possible orientations, $N_{\boldsymbol{o}_t} = 6$;

$$\boldsymbol{\Psi} = \left[\begin{array}{c|c|c|c} r(\omega^1, \boldsymbol{l}_s{}^1; \boldsymbol{l}_t{}^1, \boldsymbol{o}_t{}^1) & r(\omega^1, \boldsymbol{l}_s{}^1; \boldsymbol{l}_t{}^1, \boldsymbol{o}_t{}^2) & \cdots & r(\omega^1, \boldsymbol{l}_s{}^1; \boldsymbol{l}_t{}^4, \boldsymbol{o}_t{}^6) \\ r(\omega^1, \boldsymbol{l}_s{}^2; \boldsymbol{l}_t{}^1, \boldsymbol{o}_t{}^1) & r(\omega^1, \boldsymbol{l}_s{}^2; \boldsymbol{l}_t{}^1, \boldsymbol{o}_t{}^2) & \cdots & r(\omega^1, \boldsymbol{l}_s{}^2; \boldsymbol{l}_t{}^4, \boldsymbol{o}_t{}^6) \\ \vdots & \vdots & \ddots & \vdots \\ r(\omega^{10}, \boldsymbol{l}_s{}^5; \boldsymbol{l}_t{}^1, \boldsymbol{o}_t{}^1) & r(\omega^{10}, \boldsymbol{l}_s{}^5; \boldsymbol{l}_t{}^1, \boldsymbol{o}_t{}^2) & \cdots & r(\omega^{10}, \boldsymbol{l}_s{}^5; \boldsymbol{l}_t{}^4, \boldsymbol{o}_t{}^6) \end{array}\right] \tag{6}$$

becomes a matrix of size $N_\omega N_{\boldsymbol{l}_s} \times N_{\boldsymbol{l}_t} N_{\boldsymbol{o}_t} = 50 \times 24$.

The structure of the dictionary is the main focus of this research. In current methods, simply enumerating all the parameters through discretization is inefficient for problems like GPR and EMI where there are many variables. Not even considering the computational time of the different inversion algorithms that employ the matrix-vector multiplication, simply storing the dictionary can scale by $\mathcal{O}(N^6)$ for 3D GPR, and $\mathcal{O}(N^9)$ for 3D EMI. A simple computational example of what the scaling means for computer memory, for the smaller GPR problem, is appropriate to understand the scope at what is required for these problems. The storage requirements, with discretizations of each variable set at 100, would be $N = N_\omega N_{\boldsymbol{l}_s} = 10^6$ and $P = N_{\boldsymbol{l}_t} = 10^6$ for a total size of $NP = 10^{12}$ elements. Each element is going to be a complex double of size 8 Bytes, creating a storage requirement of 8 TBytes. Since the application for these types of problems are commonly held on small, mobile devices, 8 TBytes to store a part of the data used for the algorithm is unacceptable. Ways to address this issue are dealt with in the following chapters.

1.2.2 Backprojection

The basis for BP is matched filtering. The idea is to use the expected signal response at the receiver, after transmission and reflection, and correlate it with the received

signal response. If the correlation value is high, then the expected target parameter is present, if the correlation is low, then the expected parameter is absent [28]. This idea is present in time-domain BP. If the target response model is used from (1) with a target at location $\boldsymbol{k}$, then the 3D image would be

$$\xi(\boldsymbol{l}_t) = \sum_{\boldsymbol{l}_s} r(t - \frac{2}{v} d(\boldsymbol{l}_s, \boldsymbol{l}_t), \boldsymbol{l}_s; \boldsymbol{k}). \tag{7}$$

This can be done similarly in the frequency domain. The equation in (2) is used with a target at location $\boldsymbol{k}$, and is correlated with the expected response at each possible target location to get the image,

$$\xi(\boldsymbol{l}_t) = \sum_{\boldsymbol{l}_s} \sum_{\omega} \left| r(\omega, \boldsymbol{l}_s; \boldsymbol{k}) e^{-j\omega 2 \frac{d(\boldsymbol{l}_s, \boldsymbol{l}_t)}{v}} \right|. \tag{8}$$

The correlation advantage can be seen within the properties of the complex exponential function. For instance, take two time delays, τ_1 and τ_2, and look at one scan,

$$\left| e^{j\omega\tau_1} e^{-j\omega\tau_2} \right| = \left| e^{-j\omega(\tau_2 - \tau_1)} \right|. \tag{9}$$

The value in (9) is going to quickly approach zero if τ_1 and τ_2 are different, and will approach one as they get close together. Frequency-domain BP works well because there is a decline at the rate of the drop off of the magnitude of a sinc function in the coherence of surrounding pixels. However, the finer the discretization in the image domain, the higher the coherence becomes in the surrounding pixels. As the discretization gets finer, the theoretical targets get closer together, which gives closer time delays, and higher correlation values.

BP can also be done using a dictionary, $\boldsymbol{\Psi}$, by simply performing a vector-matrix multiplication. The response, $r(\omega, \boldsymbol{l}_s; \boldsymbol{k})$, needs to be vectorized into $\boldsymbol{r}$, which is $N_\omega N_{\boldsymbol{l}_s} \times 1$. The imaging equation becomes

$$\boldsymbol{\Xi} = \boldsymbol{\Psi}^H \boldsymbol{r}. \tag{10}$$

$\boldsymbol{\Xi}$ is capitalized even though it is technically a vector in (10). This is because $\boldsymbol{\Xi}$ should be viewed, after reshaping, as a 3D image. BP has some computational disadvantages. Either all of the scan data needs to be stored, which in a 3D image would scale on the order of N^6, or every pixel response needs to be calculated during run-time, which can be computationally inefficient.

1.2.3 Orthogonal matching pursuit

OMP is an inversion technique similar to matched filtering and BP, except it is an iterative greedy algorithm that attempts to select the best possible single response at each iteration [29]. OMP has been shown to work with data collected with EMI and GPR systems [4, 30]. This is a sparsity driven approach, which can obtain sparser solutions than BP, but requires more processing time. OMP is an iterative approach using BP, least-squares inversion, and a stopping condition. The stopping condition can be a residual bound, total number of iterations, or something else appropriate for the specific problem.

Input: A dictionary, $\boldsymbol{\Psi}$, where $\boldsymbol{\psi_t}$ represents the t^{th} column of $\boldsymbol{\Psi}$; a measurement vector, $\boldsymbol{r}$; and a stopping condition.
Output: An estimated image, $\boldsymbol{\Xi}$; and vector of indices, $\boldsymbol{\lambda}$; an update, least-squares matrix, $\boldsymbol{\Gamma}$; an approximation, $\boldsymbol{a}$, of $\boldsymbol{r}$; and an update residual, $\boldsymbol{\eta}$.
Initialize $\boldsymbol{\eta} = \boldsymbol{r}$;
Initialize $\boldsymbol{\Gamma}$ to be an empty matrix;
Initialize $\boldsymbol{\lambda}$ to be an empty matrix;
while *stopping condition is not met* **do**
  $t = \arg\max_t |\langle \boldsymbol{\psi_t}, \boldsymbol{\eta} \rangle|$;
  $\boldsymbol{\lambda} = \boldsymbol{\lambda} \cup t$;
  $\boldsymbol{\Gamma} = \boldsymbol{\Gamma} \cup \boldsymbol{\psi_t}$;
  $\boldsymbol{p} = \arg\min_{\boldsymbol{p}} \|\boldsymbol{\eta} - \boldsymbol{\Gamma p}\|_2$;
  $\boldsymbol{a} = \boldsymbol{\Gamma p}$;
  $\boldsymbol{\eta} = \boldsymbol{r} - \boldsymbol{a}$;
end
$\boldsymbol{\Xi}(\boldsymbol{\lambda}) = \boldsymbol{p}$

Algorithm 1: OMP algorithm.

1.2.4 Compressive sensing

Since CS was introduced a few years ago, there has been a large desire for researchers to find ways to use this tool [31–35]. The idea of compressive sensing is to be able to dramatically reduce the samples required to be taken of a sparse signal. A sparse signal, in discrete time, is one where the signal vector has very few nonzero elements. The most important aspect is to be able to represent the signal in a sparse way. For example, a signal $\boldsymbol{x}$ might not be sparse in one domain, but it might be in another. The signal $\boldsymbol{x}$ could be a single-frequency sinusoid in the time domain. The signal $\boldsymbol{x}$ would not be sparse in time, but if the Discrete Fourier Transform (DFT) matrix, $\boldsymbol{F}$, is used to represent $\boldsymbol{x}$,

$$\boldsymbol{x} = \boldsymbol{F}\boldsymbol{s}, \tag{11}$$

then it is easy to see that $\boldsymbol{s}$ is sparse with only one nonzero element corresponding to the frequency of the sinusoid.

Sparse representation combines well with the dictionary approach that has been used in the previous sections. Whereas $\boldsymbol{r}$ is certainly not sparse, transforming it using the dictionary $\boldsymbol{\Psi}$, would be sparse assuming the number of target parameters is very low. The response in terms of a sparsifying transform, $\boldsymbol{\Psi}$, and a sparse vector, $\boldsymbol{s}$, is

$$\boldsymbol{r} = \boldsymbol{\Psi}\boldsymbol{s}. \tag{12}$$

It is possible that $\boldsymbol{r}$ could contain multiple, additive targets.

Now that the idea of sparsity has been introduced and married with the notation from the previous sections, the CS ideas can be explained. The idea is to project the measurements onto a lower-dimensional space, while still allowing the problem to be inverted. This is done using a projection matrix, $\boldsymbol{\Phi}$, of size $N_{cm} \times N_m$, with the sparsifying transform, $\boldsymbol{\Psi}$, of size $N_m \times N_p$. N_m is the number of measurements, $N_{cm} \ll N_m$ is the number of compressed measurements, and N_p is the number of

parameters in the dictionary. The new representation of the compressed response is

$$\boldsymbol{b} = \boldsymbol{\Phi r} = \boldsymbol{\Phi \Psi s} = \boldsymbol{\Theta s}. \tag{13}$$

The projection matrix, $\boldsymbol{\Phi}$, must be designed so that $\boldsymbol{s}$ can be recovered even from the reduced number of measurements. This can be done as long as $\boldsymbol{\Theta}$ satisfies

$$1 - \epsilon_{RIP} \leq \frac{\|\boldsymbol{\Theta s}\|_2}{\|\boldsymbol{s}\|_2} \leq 1 + \epsilon_{RIP}, \tag{14}$$

the restricted isometry property (RIP), where $\epsilon_{RIP} > 0$ [34]. Creating a matrix $\boldsymbol{\Phi}$ that directly satisfies the RIP requires an unsatisfactory complexity. The complexity issue can be avoided by using random matrices for $\boldsymbol{\Phi}$. For example, if an independent and identically distributed (IID) Gaussian random matrix is used for $\boldsymbol{\Phi}$, then RIP is satisfied with extremely high probability as long as the number of compressed measurements obeys

$$N_{cm} \gtrsim \mu^2(\boldsymbol{\Phi}, \boldsymbol{\Psi}) \log(N_m) K. \tag{15}$$

K is the number of nonzero elements in $\boldsymbol{s}$ and μ is the mutual coherence [36].

The inversion process to recover the signal, or image, $\boldsymbol{s}$, is the next step. The signal must be sparse, so it is ideal to exploit this fact and only look for sparse signals. This would naturally lead to solving the optimization problem,

$$\hat{\boldsymbol{s}} = \min_{\boldsymbol{s}} \|\boldsymbol{s}\|_0 \; s.t. \; \boldsymbol{\Theta s} = \boldsymbol{b}, \tag{16}$$

for the noiseless case. The ℓ_0 norm counts the number of nonzero entries in a vector. Minimizing the ℓ_0 norm is both non-convex, and has combinatorial complexity. The RIP allows for a convex relaxation of the optimization problem in (16) using the ℓ_1 norm,

$$\hat{\boldsymbol{s}} = \min_{\boldsymbol{s}} \|\boldsymbol{s}\|_1 \; s.t. \; \boldsymbol{\Theta s} = \boldsymbol{b}. \tag{17}$$

The optimization problem in (17) can be solved with linear programming.

The noiseless case is fairly uninteresting, since in almost every practical application there would be some form of noise, or expected error, present. The measurement equation changes to account for the additive noise vector, $\boldsymbol{\eta}$,

$$\boldsymbol{b} = \boldsymbol{\Phi m} = \boldsymbol{\Phi}(\boldsymbol{r} + \boldsymbol{\eta}). \tag{18}$$

The optimization from (17) must also be changed to allow for the noise,

$$\hat{\boldsymbol{s}} = \min_{\boldsymbol{s}} \|\boldsymbol{s}\|_1 \; s.t. \; \|\boldsymbol{\Theta s} - \boldsymbol{b}\|_2 < \epsilon_2. \tag{19}$$

The optimization in (19) is called basis pursuit de-noising (BPDN) [37]. BPDN places an allowable bound on the residual between the estimated signal response, and the received signal response. There is another important form of de-noising used in these types of problems called the Dantzig Selector [38]. The optimization problem from (19) becomes

$$\hat{\boldsymbol{s}} = \min_{\boldsymbol{s}} \|\boldsymbol{s}\|_1 \; s.t. \; \|\boldsymbol{\Theta}^H(\boldsymbol{\Theta s} - \boldsymbol{b})\|_2 < \epsilon_d. \tag{20}$$

when using the Dantzig Selector. The Dantzig Selector constrains the size of the residual correlated with the CS matrix, $\boldsymbol{\Theta}$, instead of just constraining the residual. This has been shown to be an effective de-noising tool in GPR applications [13].

Some advantages with CS are that it reduces the storage requirements and produces a sparse solution. CS can also be used to reduce data-acquisition times, a constraint in the SFGPR system. However, sometimes the amount of compression is still not enough for practical problems and the complexity remains too high. A few disadvantages with CS are that it can have longer computation times than BP or OMP, and designing $\boldsymbol{\Phi}$ is not always straight forward in a practical system.

For a TPGPR, CS can also provide simplifications to the hardware design, while still decreasing the data-acquisition times and achieving acceptable detection accuracy. CS could also be used to decrease the synthetic-aperture spacing and/or increase the size of the antenna array for both the hand-held and vehicle mounted GPR systems. In all of CS applications, the objective is to improve the system performance

while still forming accurate subsurface images.

1.2.5 Semidefinite programing

Semidefinite programming (SDP) is a technique used to extract matrices with certain properties, such as positive semidefiniteness (PSD) [39]. SDP has been used as a tool for matrix completion, as well as low-rank matrix recovery problems. The type of SDP that is interesting to our application, is that of trace minimization, which is a convex relaxation on the rank-minimization problem for square matrices [40]. The basic optimization problem of interest,

$$\begin{aligned} \min \quad & tr(\boldsymbol{X}) \\ \text{s.t.} \quad & \boldsymbol{X} \succeq 0 \\ & \| \boldsymbol{m} - \mathcal{A}(\boldsymbol{X}) \|_2 < \epsilon_2, \end{aligned} \tag{21}$$

minimizes the trace of a matrix $\boldsymbol{X}$ which minimizes the sum of the eigenvalues if $\boldsymbol{X}$ is square, subject to $\boldsymbol{X} \succeq 0$, which is the PSD constraint, and a data fidelity term to account for noise or modeling error. $\mathcal{A}(\boldsymbol{X})$ represents some function of the data X, for dictionary matching, it would simply be a function that reshapes the matrix into a vector, and multiplies it by the dictionary. For example,

$$\begin{aligned} \mathcal{A}(\boldsymbol{X}) &= \boldsymbol{\Psi} reshape(\boldsymbol{X}) \\ &= \boldsymbol{\Psi x}. \end{aligned} \tag{22}$$

There is a tensor structure in the EMI model that will be examined in Chapter 3 that lends itself to using this type of inversion to efficiently extract parameters that were difficult or inefficient to extract using brute force dictionary enumeration.

1.3 Noise Parameter Selection

Another consideration is the selection of the ϵ parameters for algorithms like OMP, CS, and SDP. For ϵ_2, if the noise power, σ^2, is known, $\epsilon_2 = \sqrt{N}\sigma$. For ϵ_d, if the noise

power is known, then $\epsilon_d = \sqrt{2\log(N)}\sigma$. However, for practical systems, estimating the noise level accurately can be difficult; if so, there are a few techniques that can be used in the selection of the parameters. The first is using an L-curve, which, in theory, is simple, but can be extremely computationally intense. It is an iterative method to create a curve of sparsity vs. ϵ. For a range of values of ϵ, either (19) or (20) is solved repeatedly and the sparsity noted. The resulting plot of sparsity vs. ϵ will have a distinct "knee" and the ϵ value at the knee of the curve is selected to obtain the best solution. An example of an L-curve can be seen in Figure 4. A second, and much more computationally efficient method, is cross validation (CV) [41]. The cross validation method was shown to be effective in compressive sensing applications to GPR by Gürbüz et. al. and will be used as the method for selecting the error parameters for some of the inversion algorithms used in this thesis [13]. The process involves splitting the measurements into two separate groups. The measurement vector, $\boldsymbol{m}$, which is of length $N_{\boldsymbol{m}}$, should be split into an estimation set of length $N_E < N_{\boldsymbol{m}}$ and cross-validation set of length $N_V = N_{\boldsymbol{m}} - N_E$. An example of a CV algorithm used specifically with a Dantzig selector problem for CS and GPR is taken from Gürbüz et. al. and seen in Algorithm 2 [12]. The algorithm for CV using the quadratic constraint in (19) can be found in the original Boufounos et. al. work. The value of α is set to 0.99 to make sure that the data is under fit.

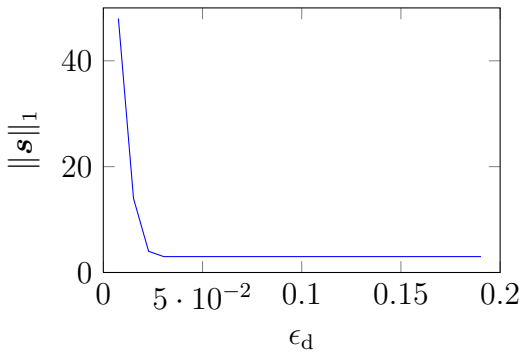


Figure 4: L-curve for ϵ_d selection.

Input: Evaluation dictionary, $\boldsymbol{\Theta}_E$; evaluation measurement vector, $\boldsymbol{b}_E$; validation dictionary, $\boldsymbol{\Theta}_V$; validation measurement vector, $\boldsymbol{b}_V$.
Output: The allowable error parameter, ϵ_d.
Initialize $\alpha = 0.99$;
Initialize $\hat{\boldsymbol{b}} = 0$;
Initialize $\epsilon_d = \alpha \left\| \boldsymbol{\Theta}_E^H \boldsymbol{b}_E \right\|_\infty$;
while $\left\| \boldsymbol{\Theta}_V^H (\boldsymbol{b}_V - \boldsymbol{\Theta}_V \hat{\boldsymbol{b}}) \right\|_\infty < \epsilon_d$ **do**
  $\hat{\boldsymbol{b}} = \min_{\boldsymbol{b}} \|\boldsymbol{b}\|_1 \ s.t. \ \left\| \boldsymbol{\Theta}_E^H (\boldsymbol{\Theta}_E \boldsymbol{b} - \boldsymbol{b}_E) \right\|_\infty < \epsilon_d$;
  $\epsilon_d = \left\| \boldsymbol{\Theta}_V^H (\boldsymbol{b}_V - \boldsymbol{\Theta}_V \hat{\boldsymbol{b}}) \right\|_\infty$;
end

Algorithm 2: CV algorithm for Dantzig selector.

1.4 Accuracy Testing

There are a few different tests for accuracy that are used in this research. For the GPR problem, it is typical to have multiple targets, but the support error is more important than the amplitude because the support is what determines the location of the target. So a metric needs to be used to handle evaluating the support. For the EMI problem, because of the ability to dissect a target into its electromagnetic dipoles and image each one of them independently, it is rare to come across multiple identical targets. Most of our laboratory experiments involve only a single target. There are also many laboratory experiments performed on the same target with many different location and orientation parameters, so an error histogram is used to evaluate the limitations of the data-acquisition system and the inversion algorithms to help improve them in the future.

1.4.1 Earth mover's distance

A suitable metric is needed to evaluate accuracy of the estimated solution vector in the GPR problem. Typically, something like probability of detection or mean-squared error (MSE) is used in evaluating the result. For this specific application, since the concern is more with the estimated location than with the amplitude, a measure called

the earth mover's distance (EMD) can be used [42]. The EMD takes into account the error between the support, as well as the error in amplitudes. Just like MSE, a lower EMD constitutes a "better" solution. For example, if a mine was detected, but its location was off by 1 cm, the EMD would be lower than if the same mine location was off by 30 cm. However, if the MSE were used, the error would be identical for both cases assuming they had equal amplitudes. A simple 1D example of EMD vs. MSE can be seen in Figure 5. The EMD is calculated using a fast method from Pele et. al. [43].

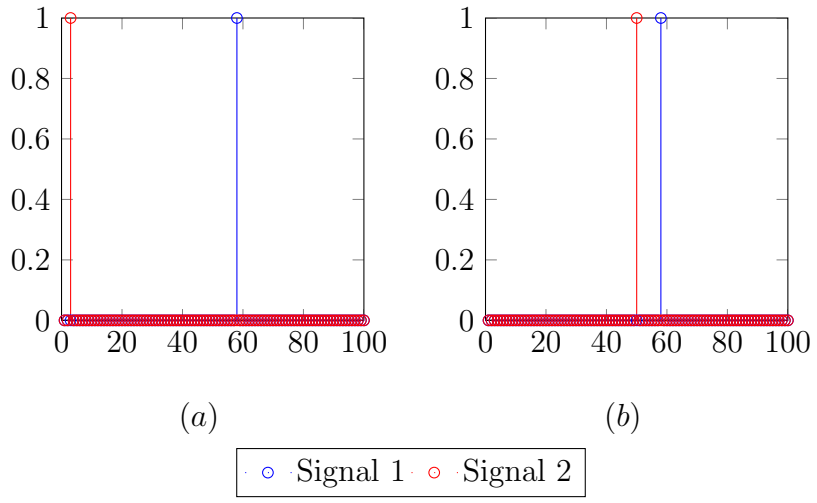


Figure 5: Signal examples with equal MSE but (a) high EMD (b) low EMD.

A quick example of EMD analysis for a small GPR problem using established algorithms will help illustrate how this process will work. A simple single target environment is set up to be simulated. The BP solution is shown in Figure 6(a). The CS, OMP, and compressed orthogonal matching pursuit (COMP) solutions are identical and shown in Figure 6(b). The BP image is much less sparse than the other algorithms, and thus has a much higher base EMD than the other sparse algorithms, which can be seen in Figure 6(c). It can also be used to evaluate a CS based algorithm, like in Figure 6(d), which would help to identify the number of compressed

measurements needed to have high probability of accurate recovery.

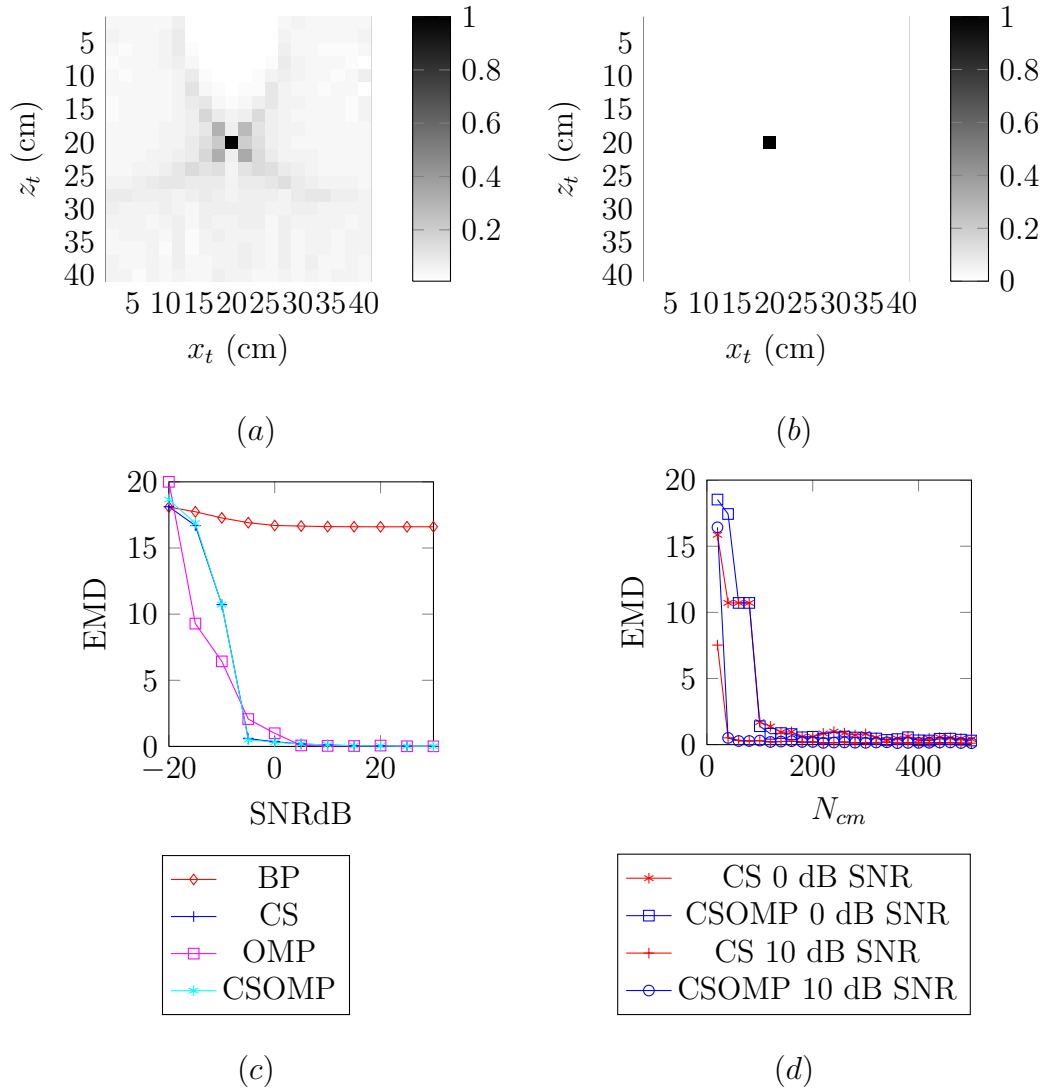


Figure 6: Solution examples of (a) BP and (b) CS, COMP, and OMP of a single target. EMD analysis of (c) all four algorithms with respect to SNR and (d) compressed algorithms with respect to number of compressed measurements N_{cm}.

1.4.2 Error histograms

The error histogram concept is fairly simple for determining probability of error in the EMI problem. In a single target scenario, where orientation and location are being extracted, a simple distance error and angle error can be calculated. The solution

is considered to be a single error in either distance or orientation if the calculated error is more than a specified threshold. The number of errors for a particular target parameter are counted, and the percentage of total errors is recorded and plotted in a multidimensional histogram. An example of a location error histogram can be seen in Figure 7. This was created by imaging many parameter combinations of a metal loop in a lab with maximal correlation, taking the maximal pixel from the BP solution as the target location. The model and data-acquisition system used were the ones described in subsection 1.1.2. This type of image is useful because it shows the limitations of the sensor. The heavy errors at the deep offset locations show that typically the sensor is going to do a better job of imaging objects shallow, and close to the center of the sensor array. This will be used in the analysis of the EMI inversion method and will ultimately help in determining some functional changes that need to be made to the hardware system.

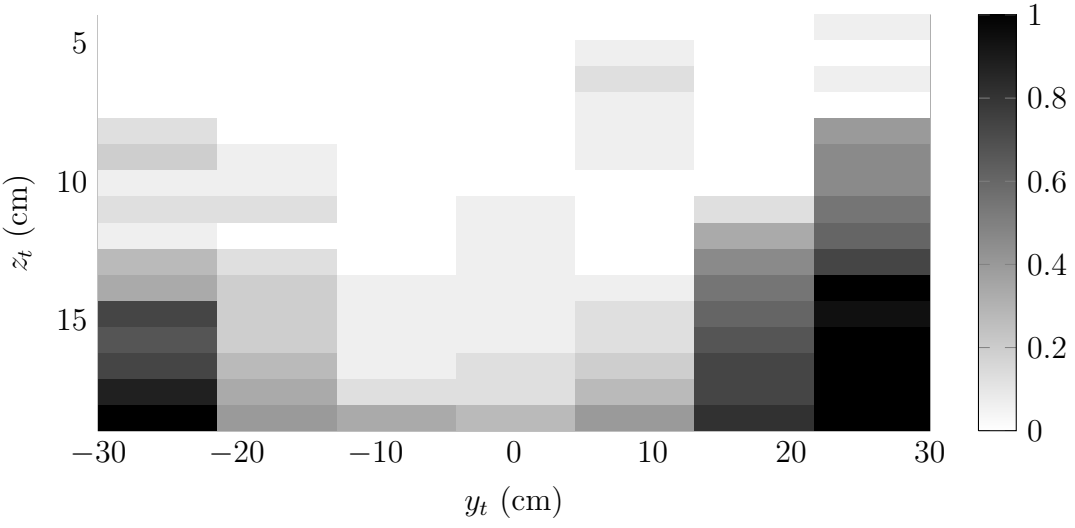


Figure 7: Location error histogram from an EMI laboratory experiment.

1.5 Outline

The remainder of the document is organized into three chapters. Chapter 2 discusses the shift-invariance property that increases the efficiency of the imaging algorithms for the GPR problem. Chapter 3 introduces a "tensor amplitude" which can reduce the complexity of the EMI problem while increasing accuracy. The final chapter is a short conclusion wrapping up the work and showing possible avenues for improvement in the future.

CHAPTER II

INCREASING EFFICIENCY FOR THE GPR PROBLEM

The previous methods for performing GPR imaging have been discussed in Chapter 1. Although the recent work in CS has been shown to effectively reduce the data-acquisition time, it has not been practically applied because of its large computational complexity. The computational complexity comes primarily from the requirements of storing and applying $\mathbf{\Psi}$. In this thesis, a method is developed which dramatically reduces the scalability of all dictionary matching problems, from $\mathcal{O}(N^6)$ to $\mathcal{O}(N^4)$ for storage and application time, by exploiting a translational invariance property that can be guaranteed if small considerations are taken during the data-acquisition steps. The remainder of the chapter will be split into six sections. Section 2.1 will introduce the shift invariance property with respect to the point-target model for GPR. Section 2.2 discusses how the shift invariance can be exploited in inversion algorithms to get the desired computational reductions. Section 2.3 shows some 2D and 3D simulations comparing exploiting versus not exploiting the translational invariance. Section 2.4 outlines a few laboratory experiments that were inverted using the new techniques. Finally, Section 2.5 and Section 2.6 will show a simulation for how a CS GPR could be designed and give brief conclusions, respectively.

2.1 Model Setup and Implementation

The GPR problem can be described as a parameter-detection problem that relies on model-based inversion. The data flow and processing blocks for this detection system were shown in Figure 1. There are two sources of sampling: acquisition sampling and model discretization. Acquisition sampling is where a sensor measures the

environment at many locations. Model discretization is where a dictionary, $\mathbf{\Psi}$, is constructed using a sampling of a continuous model created from the knowledge of the data-acquisition system and the parameters that need to be extracted. The measurements collected with the data-acquisition system and the dictionary $\mathbf{\Psi}$ are combined during the inversion step to obtain a 3D image that estimates the environment.

2.1.1 Response model

The data-acquisition technique for a GPR includes the creation of a synthetic aperture by moving a sensor to the positions, $\boldsymbol{l}_s = (x_s, y_s, z_s)$. For a 3D image, the scan positions, $\boldsymbol{l}_s$, would be indexed over a grid in the 2D plane as was shown in Figure 2, z_s is generally held constant. In the case where the transmitter and receiver are scanned together, the scalar point-target model is

$$R(\omega, \boldsymbol{l}_s; \boldsymbol{l}_t) = \frac{\rho}{S} e^{j\omega\tau(\boldsymbol{l}_s, \boldsymbol{l}_t, c, v)}, \tag{23}$$

where a series of stepped-frequency signals, at frequency ω, are sent from a transmitter located at $\boldsymbol{l}_s$, and then reflected off a target at location $\boldsymbol{l}_t = (x_t, y_t, z_t)$. The target has a reflection coefficient, ρ, and S is the electromagnetic spreading function which may or may not be known. The time delay function τ uses an approximation of Snell's law to calculate the wave path through air with velocity c and at the boundary of a medium with velocity v [44]. An example of the wave path from a bistatic GPR traveling into the ground and reflecting off a target can be seen in Figure 8, where $\tau = \tau_1 + \tau_2 + \tau_3 + \tau_4$.

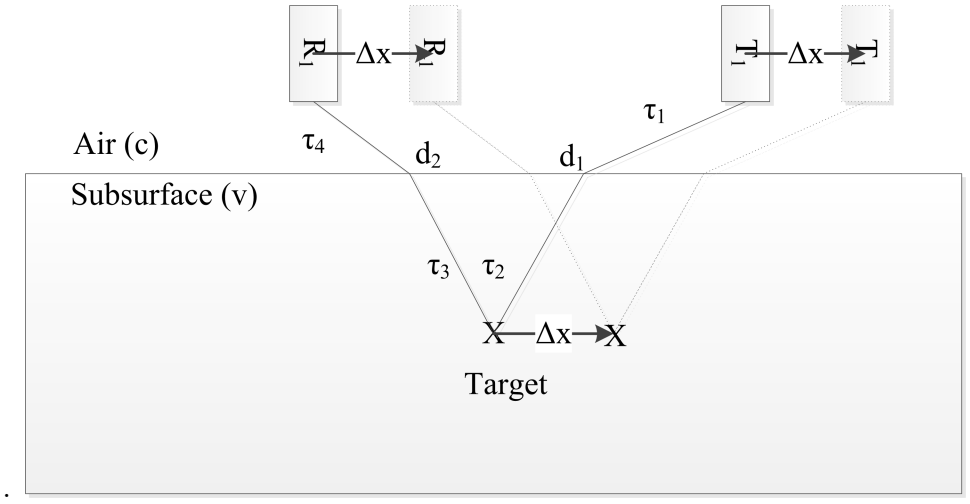


Figure 8: EM path through multiple mediums.

Examples of a simulated time-domain and frequency domain measurement can be seen in Figure 9. There will also be a reflection from the air-ground interface, and eliminating this ground response is an active research topic in its own right. In particular, there is some work associated with ground removal for GPR geared towards a CS application introduced by Tuncer et. al., but ground removal will not be considered in this thesis [45]. That is, the simulations shown here use a single medium with no velocity changes. The subsurface laboratory experiments in subsection 2.4.2 have a fairly uniform ground response, so it is possible to subtract out the ground bounce from the measurements since the height of the sensor is known in the controlled lab experiment.

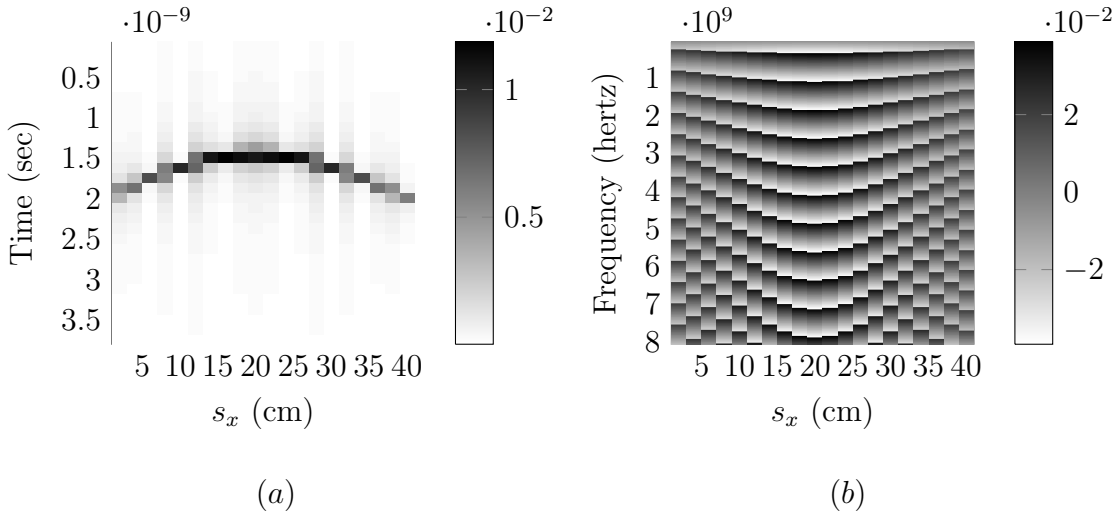


Figure 9: Simulated measurements for 2D in (a) time domain (b) frequency domain showing the magnitude multiplied by the phase.

The model dictionary, $\mathbf{\Psi}$, that can be used with any number of different imaging techniques can be created by enumerating (23) for all possible parameter discretizations. The first step is to determine which parameters are associated with the measurements and which will be extracted in finding the targets. The highest priority variable in a landmine detection system is target location, $\boldsymbol{l}_t$. The spreading factor and the strength of the target can be combined into a single amplitude, $s(\boldsymbol{l}_t)$, and would not require enumeration. For the remainder of this section, only the SFGPR case will be studied so the frequency ω and sensor locations $\boldsymbol{l}_s$ are the measurement parameters. The discussion could easily be switched to TPGPR. The SFGPR model for a single target can be rewritten as,

$$\begin{aligned} R(\omega, \boldsymbol{l}_s; \boldsymbol{l}_t) &= s(\boldsymbol{l}_t) e^{j\omega\tau(d(\boldsymbol{l}_s, \boldsymbol{l}_t), c, v)} \\ &= s(\boldsymbol{l}_t)\psi(\omega, \boldsymbol{l}_s; \boldsymbol{l}_t), \end{aligned} \tag{24}$$

where $d(\boldsymbol{l}_s, \boldsymbol{l}_t)$ is a 3D distance function. The measurements are made at a finite set of frequencies ω and a finite number of sensor locations $\boldsymbol{l}_s = (x_s, y_s)$

One column vector of the dictionary matrix $\mathbf{\Psi}$ is the vector $\boldsymbol{\psi}(\boldsymbol{l}_t)$ whose entries

are all the measurements created by evaluating the model $R(\omega, \boldsymbol{l}_s; \boldsymbol{l}_t)$ for a fixed $\boldsymbol{l}_t$ while enumerating all possible triples of the 3 measurement space parameters, ω, x_s and y_s. The final dictionary is created by concatenating the target location response vectors as the column space,

$$\boldsymbol{\Psi} = \left[\; \boldsymbol{\psi}(\boldsymbol{l}_t{}^1) \;\middle|\; \boldsymbol{\psi}(\boldsymbol{l}_t{}^2) \;\middle|\; \cdots \;\middle|\; \boldsymbol{\psi}(\boldsymbol{l}_t{}^{N_{l_t}}) \; \right]. \tag{25}$$

where the $N_{\boldsymbol{l}_t}$ values of $\boldsymbol{l}_t$ are obtained by enumerating all possible triples of the target location parameters, x_t, y_t, and z_t. If the number of values for each parameter is N, then $\boldsymbol{\Psi}$ is an $N^3 \times N^3$ matrix. For example, with $N = 100$ the dictionary matrix is $10^6 \times 10^6$ with one trillion (10^{12}) entries.

Using the dictionary matrix in (25), the response vector can be expressed as,

$$\boldsymbol{r} = \sum_{\boldsymbol{l}_t} s(\boldsymbol{l}_t)\boldsymbol{\psi}(\boldsymbol{l}_t) = \boldsymbol{\Psi}\boldsymbol{s}, \tag{26}$$

where $\boldsymbol{s}$ is a sparse vector which is only nonzero at the target locations. The indexing of the vector $\boldsymbol{r}$ must follow the enumeration of the triple (ω, x_s, y_s) used for the measurement vectors $\boldsymbol{\psi}(\boldsymbol{l}_t)$. The indexing of $\boldsymbol{s}$ must follow that of the triple (x_t, y_t, z_t) used for the target locations.

The inversion process is done to recover $\boldsymbol{s}$ from the measurements,

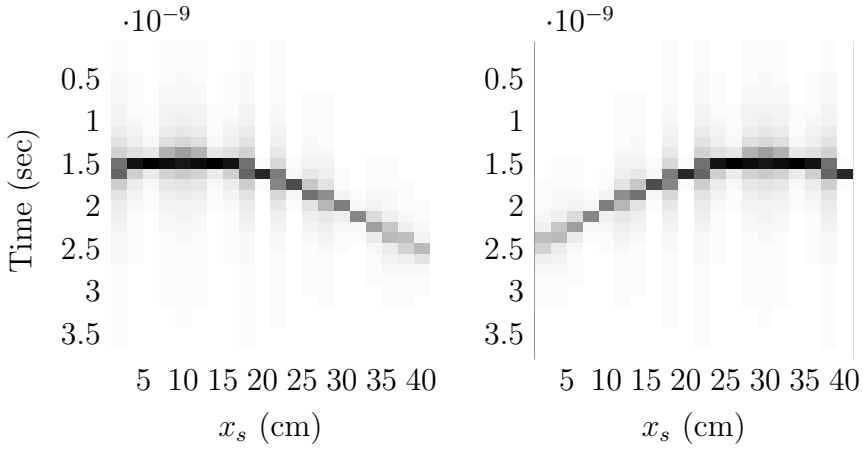
$$\boldsymbol{m} = \boldsymbol{r} + \boldsymbol{\eta}, \tag{27}$$

where $\boldsymbol{\eta}$ is an additive-noise vector. However, the size of $\boldsymbol{\Psi}$ and the computational complexity of the inversion algorithms make 3D imaging problematic for real-world applications. To address this issue, some structural changes are presented that can be made to simplify the way that the dictionary $\boldsymbol{\Psi}$ is created, stored, and applied in the different inversions.

2.1.2 Shift-invariance property

A simplification of $\boldsymbol{\Psi}$ is possible because the GPR acquisition system can have the extremely powerful property of spatial-shift invariance [46–48]. A graphical example of

what the time-domain measurements look like for a target that has been horizontally shifted at the same depth can be seen in Figure 10.



(a) (b)

Figure 10: Simulated time-domain measurements of shifted targets.

The key idea is that the model response (23) at the horizontal aperture of sensors will shift in tandem with horizontal shifts in the target positions at a fixed depth. This is true because the distance function $d(\boldsymbol{l}_s, \boldsymbol{l}_t)$ shown in Figure 8 does not change with equal horizontal shifts of the sensor and target. However, the computation takes place with discrete grids for the positions $\boldsymbol{l}_s$ and $\boldsymbol{l}_t$, so the grids must also support the shift invariance. While it is not necessary in the general case for the sensor locations $\boldsymbol{l}_s$ to be uniformly spaced, in this thesis it will be assumed that $\boldsymbol{l}_s$ is uniformly spaced. If $\boldsymbol{l}_s$ is not physically uniformly spaced, it can be interpolated onto a uniformly spaced grid. To show how the shift-invariance property simplifies the computation for the collection of SAR measurements, a 2D example will be used.

First, rewrite the response vector from (26) as a sum of products,

$$r(\omega, x_s) = \sum_{z_t} \sum_{x_t} s(x_t, z_t) e^{-j\omega\tau(x_s, x_t, z_t)}. \tag{28}$$

Next, discretize the x dimension as follows: $m\Delta x$ for $m = 1, 2, \ldots N_{x_s}$ for x_s and $(h + \alpha)\Delta x$ for $h = 1, 2, \ldots N_{x_t}$ for x_t where α is a constant value $0 \leq \alpha < \Delta x$. The x

discretization creates a new representation for the response vector,

$$r(\omega, m\Delta x) = \sum_{z_t} \sum_{h=1}^{N_{x_t}} s((h+\alpha)\Delta x, z_t) e^{-j\omega\tau(m\Delta x, h\Delta x, \alpha, z_t)}. \tag{29}$$

If the time delay τ is examined for a monostatic system measuring in a single medium, we obtain

$$\begin{aligned} \tau(m\Delta x, h\Delta x, \alpha, z_t) &= (1/c)\sqrt{(m\Delta x - (h+\alpha)\Delta x)^2 + z_t^2} \\ &= (1/c)\sqrt{((m-h)-\alpha)^2(\Delta x)^2 + z_t^2}. \end{aligned} \tag{30}$$

Thus the time delay depends on the index difference $(m-h)$, and it can be shown that the inner sum in (29) is a discrete convolution. The time delay function τ keeps horizontal-shift invariance even when the system is not monostatic and when the medium velocity is allowed to change with z, but not with x or y. The horizontal-shift invariance can be visualized with Figure 8. If the target, T_1, and R_1 are shifted by a equal value, d_1 and d_2 will shift by the same amount, so τ will be the same as before the shift. Therefore, with a fixed Δx the exponential in (29) is a function of the index difference $(m-h)$.

$$e^{-j\omega\tau(m\Delta x, h\Delta x, \alpha, z_t)} = e^{-j\omega\tau((m-h)\Delta x, \alpha, z_t)} \tag{31}$$

Now the inner sum of (29) can be rewritten as a convolution because of index-shift invariance:

$$\begin{aligned} r(\omega, m\Delta x) &= \sum_{z_t} \underbrace{\left(\sum_{h=1}^{N_{x_t}} s((h+\alpha)\Delta x, z_t) e^{-j\omega\tau((m-h)\Delta x, \alpha, z_t)} \right)}_{\text{convolution w.r.t. } m} \\ &= \sum_{z_t} s((m+\alpha)\Delta x, z_t) \underset{m}{*} e^{-j\omega\tau((m)\Delta x, \alpha, z_t)} \\ &= \sum_{z_t} s((m+\alpha)\Delta x, z_t) \underset{m}{*} \psi(\omega, m\Delta x, \alpha, z_t). \end{aligned} \tag{32}$$

The convolution representation shown in (32) uses the same dictionary as was created in (24), but the discretization for x_t has been replaced by the constant α. Changing the discretization from x_t to α is significant because it drops the storage requirements from N_{x_t} to 1. However if there is a desire to upsample the image locations, e.g., to get upsampling by a factor of two, $\alpha_1 = 0$ and $\alpha_2 = 0.5\Delta x$ could be used and the process would need to be repeated for each α_i. With these steps outlined for the forward operator, the adjoint is fairly trivial. Transitioning back to matrix notation, shift invariance leads to a Toeplitz or block-Toeplitz structure in $\boldsymbol{\Psi}$. For simplicity consider an example with α_1, α_2, and $N_\alpha = 2$, which could be expanded trivially. The columns correspond to the sensor positions, x_s, the rows correspond to the simulated target locations, x_t, and the entries in the $\boldsymbol{D}^H$ matrix correspond to the distance between x_s and x_t. The distance between x_s and x_t is going to be $|m - h - \alpha_i|$. A reasonable assumption that is made here is that $N_{x_t} = N_{x_s}$. When they are not equal, the Toeplitz structure would just not be square. The difference matrix is

$$\boldsymbol{D}^H = \begin{bmatrix} \alpha_1 & \Delta x - \alpha_1 & \cdots & (N_{x_s} - 1)\Delta x - \alpha_1) \\ \alpha_2 & \Delta x - \alpha_2 & \cdots & (N_{x_s} - 1)\Delta x - \alpha_2) \\ \Delta x + \alpha_1 & \alpha_1 & \cdots & (N_{x_s} - 2)\Delta x - \alpha_1) \\ \Delta x + \alpha_2 & \alpha_2 & \cdots & (N_{x_s} - 2)\Delta x - \alpha_2) \\ \vdots & \vdots & \ddots & \vdots \\ (N_{x_t} - 1)\Delta x + \alpha_1 & (N_{x_t} - 2)\Delta x + \alpha_1 & \cdots & \alpha_1 \\ (N_{x_t} - 1)\Delta x + \alpha_2 & (N_{x_t} - 2)\Delta x + \alpha_2 & \cdots & \alpha_2 \end{bmatrix}. \quad (33)$$

This matrix it is not Toeplitz, but with a slight reorganization of the rows, it can

become block Toeplitz with N_α blocks,

$$\boldsymbol{D}^H = \left[\begin{array}{cccc} \alpha_1 & \Delta x - \alpha_1 & \cdots & (N_{x_s}-1)\Delta x - \alpha_1) \\ \Delta x + \alpha_1 & \alpha_1 & \cdots & (N_{x_s}-2)\Delta x - \alpha_1) \\ \vdots & \vdots & \ddots & \vdots \\ (N_{x_t}-1)\Delta x + \alpha_1 & (N_{x_t}-2)\Delta x + \alpha_1 & \cdots & \alpha_1 \\ \hline \alpha_2 & \Delta x - \alpha_2 & \cdots & (N_{x_s}-1)\Delta x - \alpha_2) \\ \Delta x + \alpha_2 & \alpha_2 & \cdots & (N_{x_s}-2)\Delta x - \alpha_2) \\ \vdots & \vdots & \ddots & \vdots \\ (N_{x_t}-1)\Delta x + \alpha_2 & (N_{x_t}-2)\Delta x + \alpha_2 & \cdots & \alpha_2 \end{array}\right]. \quad (34)$$

This difference matrix, $\boldsymbol{D}$, is built directly into the representation matrix,

$$\boldsymbol{\Psi}(\omega_1, \boldsymbol{D}, z_t) = e^{j\omega\tau(\boldsymbol{D}, t_{z_1})}, \quad (35)$$

giving $\boldsymbol{\Psi}$ a block structure with $N_\alpha N_\omega N_{z_t}$ blocks of size $N_{x_s} \times N_{x_t}$. It is well known that a block-Toeplitz matrix can be stored with a single vector for each block.[1]

[1]For the examples given in this thesis it will be assumed that the sensor spacing and the simulated target spacing are identical, $N_\alpha = 1$, $\alpha_1 = 0$.

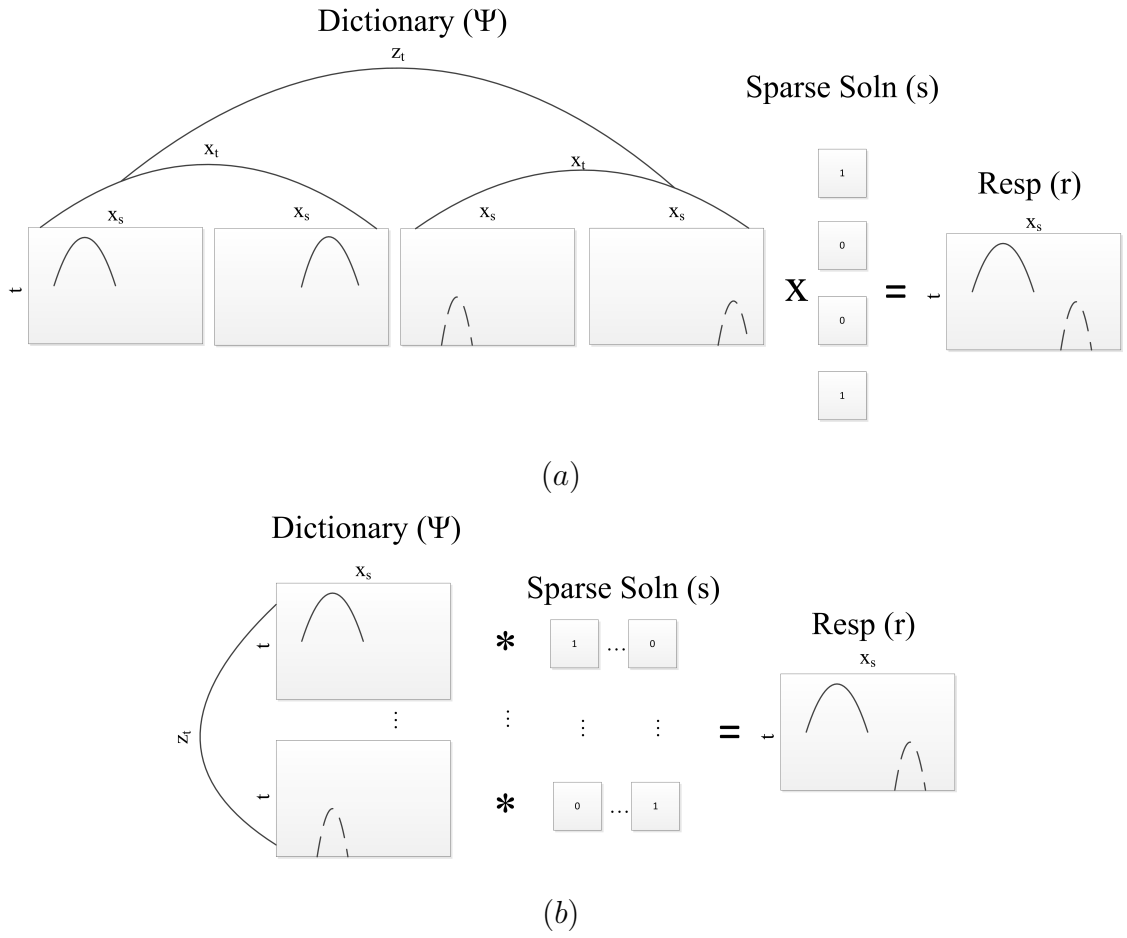


Figure 11: Dictionary implementation (a) explicit enumeration with matrix multiplication, (b) exploiting shift invariance by using correlation.

Toeplitz matrices have been shown to be an effective way to reduce computational complexity in random sampling matrices for CS [36, 49]. A graphical example of the structural changes that would need to be made to the storage and application of the $\mathbf{\Psi}$ matrix used for 2D imaging in the time domain can be seen in Figure 11. Figure 11(a) shows the traditional dictionary, where every simulated target position $\boldsymbol{l}_t$ is enumerated into the columns of $\mathbf{\Psi}$ and the dictionary is applied using standard matrix-vector multiplication. Figure 11(b) shows a reduced size $\mathbf{\Psi}$, where $N_\alpha = 1$, that does not enumerate in the horizontal dimensions and is applied using a convolution operator

along the horizontal position, instead of a simple matrix-vector multiply. The particular $\boldsymbol{\Psi}$ structure shown in Figure 11(b) has the added bonus that the computational operations required are $\mathcal{O}(N\log(N))$, by using the FFT, instead of $\mathcal{O}(N^2)$ for each dimension where the shift invariance can be exploited. A traditional (explicit) $\boldsymbol{\Psi}$ used to image 3 dimensions can be stored and applied in $\mathcal{O}(N^6)$, assuming all measurements and parameters are equally discretized. On the other hand, when $\boldsymbol{\Psi}$ has the Toeplitz structure in both horizontal $\boldsymbol{l}_t$ dimensions equivalent to $\boldsymbol{l}_s$, taking advantage of the FFT would reduce the storage to $\mathcal{O}(N^4)$ and the computation to $\mathcal{O}(N^4)$. A flow chart for the special properties of the GPR and the resultant dimensionality reductions can be seen in Figure 12. The addition of using a CS inversion would allow for a further reduction in the frequency domain by using $N_{c\omega} \ll N_\omega$ compressed frequencies. A graphical example of the element reduction in using both compressive sensing and exploiting the translational invariance can be seen in Figure 13. The take away from Figure 13 is that each representation can accurately invert a set of measurements, while the method exploiting translational invariance with the FFT is far more efficient.

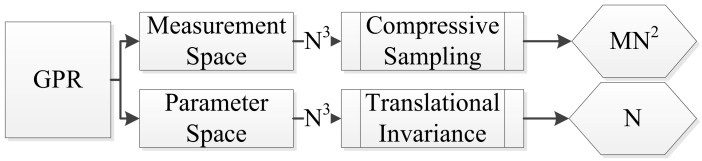


Figure 12: Storage requirements for the GPR system.

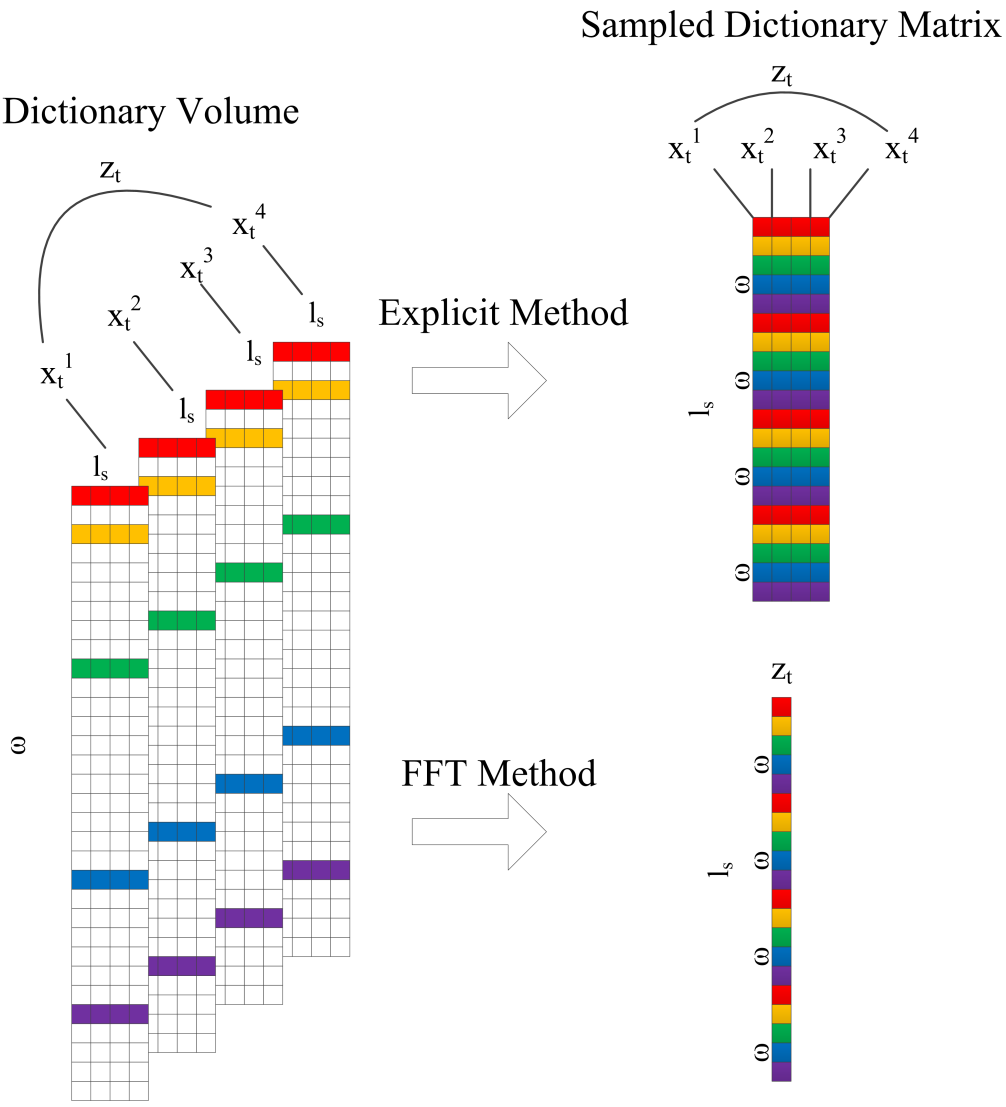


Figure 13: Data volume for different $\boldsymbol{\Psi}$ representations.

2.2 Implementation Specifics for Structure Change

Now that the structure within the dictionary has been identified, the inversion algorithms no longer use a simple matrix-vector multiplication to apply $\boldsymbol{\Psi}$. The matrix-vector multiplication has to be replaced with a specifically designed function to perform the equivalent of the forward, $g_{\boldsymbol{\Theta}}$, and adjoint, $g_{\boldsymbol{\Theta}}^H$, operators of a reduced matrix $\boldsymbol{\Psi}_\alpha$ to the sparse vector $\boldsymbol{s}$ [47]. It is important to note that $\boldsymbol{\Psi}_\alpha$ is built by enumerating (24), where $\boldsymbol{l}_t$ is only enumerated for z_t and not x_t or y_t.

The first step is to make sure that the discretization of $\boldsymbol{l}_s$ and $\boldsymbol{l}_t$ provide the translational invariance discussed in subsection 2.1.2. Next, identify how the convolution operation from Figure 11(b) is going to be performed to take advantage of the translational invariance. The FFT can be used to perform circular convolution efficiently, and with the use of a zero-padding operator, Z, linear convolution. The FFT can only be performed in the horizontal dimensions if the time and frequency samples are the same for each $\boldsymbol{l}_s$. Since there is translational invariance in both dimensions of $\boldsymbol{l}_s$, the zero padding must take place in both dimensions. The simplest thing to do is to add $N_{x_s}/2$ discretizations to the beginning and the end of the x_s dimension, and do the same for y. The zero pad will allow for shifts to take place within the desired range of $\boldsymbol{l}_s$ without wrap-around effects from the circular convolution. A slightly more efficient way would be to zero-pad with a number based on the maximal beamwidth of the antenna in space. Then, the FFT needs to be taken in both x and y of $\boldsymbol{l}_s$. The same operations must be performed to the corresponding x and y of $\boldsymbol{l}_t$ in the sparse vector $\boldsymbol{s}$. These are then combined through a series of index multiplications shown specifically in Algorithm 3. The summation over z_t should be interpreted as a for loop over depth that adds the results of fast FFT (horizontal) convolution done for each depth. The final step is to then compress the measurements with $\boldsymbol{\Phi}$ if a compression algorithm is being used, like CS. In the case where no compression is required, $\boldsymbol{\Phi}$ can be removed or equivalently set to the identity matrix.

Now that a function to perform the forward operation has been created, a similar function for the adjoint operation can be described. The adjoint matrix $\boldsymbol{\Psi}^H$ is the conjugate transpose, so it possesses Toeplitz sub-blocks, and fast FFT convolution can be done. In addition, the adjoint can be created by working the forward algorithm backwards. The specifics for how to perform the adjoint operator $g^H_{\boldsymbol{\Theta}}$ can be found in Algorithm 4.

The description of the algorithms above can be modified to get a more time

Input: Compression matrix, $\boldsymbol{\Phi}$; response dictionary, $\boldsymbol{\Psi}_\alpha$; sparse vector, $\boldsymbol{s}$
Output: compressed measurement vector $\boldsymbol{b}$
Reshape $\boldsymbol{\Psi}_\alpha$ and $\boldsymbol{s}$ to have a dimension for every variable (for 3D imaging $\boldsymbol{\Psi}_\alpha$ is 4D and $\boldsymbol{s}$ is 3D);
for *all* ω **do**
  $\tilde{\boldsymbol{m}} = 0$;
  for *all* z_t **do**
    $\tilde{\boldsymbol{s}}(k_x, k_y, z_t) = \text{FFT}_x(\text{FFT}_y\{Z(\boldsymbol{s}(x_t, y_t, z_t))\})$;
    $\tilde{\boldsymbol{\psi}}_g(\omega, k_x, k_y, z_t) = \text{FFT}_x(\text{FFT}_y\{Z(\boldsymbol{\psi}_\alpha(\omega, k_x, k_y, z_t))\})$;
    $\tilde{\boldsymbol{m}}(\omega, k_x, k_y) = \tilde{\boldsymbol{m}}(\omega, k_x, k_y) + \tilde{\boldsymbol{\psi}}_g(\omega, k_x, k_y, z_t)\tilde{\boldsymbol{s}}(k_x, k_y, z_t)$;
  end
  $\boldsymbol{m}(\omega, x_s, y_s) = Z^{-1}\{\text{IFFT}_{k_y}(\text{IFFT}_{k_x}\{\tilde{\boldsymbol{m}}(\omega, k_x, k_y)\})\}$
end
Reshape $\boldsymbol{m}(\omega, k_x, k_y,)$ into a vector $\boldsymbol{m}$;
$\boldsymbol{b} = \boldsymbol{\Phi}\boldsymbol{m}$;

Algorithm 3: Forward function algorithm $\boldsymbol{b} = g_{\boldsymbol{\Theta}}(\boldsymbol{\Phi}, \boldsymbol{\Psi}_\alpha, \boldsymbol{s})$ for SFGPR.

Input: Compression matrix, $\boldsymbol{\Phi}$; response dictionary, $\boldsymbol{\Psi}_\alpha$; compressed measurement vector $\boldsymbol{b}$.
Output: Sparse vector, $\boldsymbol{s}$.
Reshape $\boldsymbol{\Psi}_\alpha$ to have a dimension for every variable (for 3D imaging $\boldsymbol{\Psi}$ is 4D);
$\boldsymbol{m} = \boldsymbol{\Phi}^H\boldsymbol{b}$;
for *all* z_t **do**
  $\tilde{\boldsymbol{s}} = 0$;
  for *all* ω **do**
    $\tilde{\boldsymbol{m}}(\omega, k_x, k_y) = \text{FFT}_x(\text{FFT}_y\{Z(\boldsymbol{m}(\omega, x_s, y_s))\})$;
    $\tilde{\boldsymbol{\psi}}_\alpha(\omega, k_x, k_y, z_t) = \text{FFT}_x(\text{FFT}_y\{Z(\{\boldsymbol{\psi}_\alpha(\omega, x_t, y_t, z_t)\})\})$;
    $\tilde{\boldsymbol{s}}(k_x, k_y, z_t) = \tilde{\boldsymbol{s}}(k_x, k_y, z_t) + \tilde{\boldsymbol{\psi}}_\alpha^H(\omega, k_x, k_y, z_t)\tilde{\boldsymbol{m}}(\omega, k_x, k_y)$;
  end
  $\boldsymbol{s}(x_t, y_t, z_t) = Z^{-1}\{\text{IFFT}_{k_y}(\text{IFFT}_{k_x}\{\tilde{\boldsymbol{s}}(k_x, k_y, z_t)\})\}$;
end

Algorithm 4: Adjoint function algorithm $\boldsymbol{s} = g_{\boldsymbol{\Theta}}^H(\boldsymbol{\Phi}, \boldsymbol{\Psi}_\alpha, \boldsymbol{b})$ for SFGPR.

efficient implementation. Many of the steps can be performed off-line, for instance calculating $\tilde{\boldsymbol{\Psi}}_\alpha$ and taking the transform of $\boldsymbol{b}$. Also, the for loops are easy to vectorize for faster computation when using MATLAB. For example, running BP using $g_{\boldsymbol{\Theta}}^H$ on a 3D simulation, that is run and analyzed in subsection 2.3.2, takes 5.28 seconds if the multiplies and adds are done in a loop, and 0.13 by using vectorized operations to perform the same task.

The algorithms described in Algorithm 3 and Algorithm 4 describe $g_{\boldsymbol{\Theta}}$ and $g_{\boldsymbol{\Theta}}^H$ if a general $\boldsymbol{\Phi}$ were to be used in a compression algorithm. However, to use a general $\boldsymbol{\Phi}$ requires a complete sampling during data-acquisition and eliminates the usefulness of compressive algorithms in terms of data acquisition. Making sure that $\boldsymbol{\Phi}$ is a random sampling matrix where the random frequencies, or random time samples, at each $\boldsymbol{l}_s$ are equal, would allow for $\boldsymbol{\Phi}$ to be applied to $\boldsymbol{\Psi}_\alpha$ ***before*** the zero pad and FFT take place. Applying $\boldsymbol{\Phi}$ before the zero pad and FFT allows for the compressed sampling to be performed during data acquisition. In other words, if $\boldsymbol{\Phi\Psi}$ has the same shift-invariant property as $\boldsymbol{\Psi}$, then $\boldsymbol{\Phi}$ can be applied before the zero pad and FFT operations.

For the rest of this chapter, the BPDN will be used for CS applications, and (19) becomes

$$\hat{\boldsymbol{s}} = \min_{\boldsymbol{s}} \|\boldsymbol{s}\|_1 \qquad s.t.\ \|g_{\boldsymbol{\Theta}}(\boldsymbol{s}) - \tilde{\boldsymbol{b}}\|_2 < \epsilon_2. \tag{36}$$

The reason BPDN is going to be used as opposed to the Dantzig selector that was used in the previous work, is simply because SPGL1 supports BPDN and is much more computationally efficient than some of the other algorithms like ℓ_1-MAGIC and CVX [50–52]. There is a slight change in (36), $\tilde{\boldsymbol{b}}$ is used as the compressed measurement vector instead of $\boldsymbol{b}$, so it is important to remember that the measurements must be zero-padded and passed through the FFT to match the output of $g_{\boldsymbol{\Theta}}$.

2.2.1 Designing Φ for compression in GPR

The first step of the process is designing a sampling scheme to either reduce the sampling time with compressed sensing, or to increase the performance of the system without increasing the data-acquisition time. In compressed sampling, the compression matrix, $\boldsymbol{\Phi}$, needs to satisfy the restricted isometry property (RIP) discussed in subsection 1.2.4. Again, the mutual coherence is

$$\mu(\boldsymbol{\Phi}, \boldsymbol{\Psi}) = \max_{\substack{\boldsymbol{\phi}_k \in Rows(\boldsymbol{\Phi}) \\ \boldsymbol{\psi}_t \in Cols(\boldsymbol{\Psi})}} |\langle \boldsymbol{\phi}_k, \boldsymbol{\psi}_t \rangle| \tag{37}$$

[33]. The rows of $\mathbf{\Phi}$ are normalized to $\|\boldsymbol{\phi}_k\|_2^2 = N$ and the columns of $\mathbf{\Psi}$ are normalized to one. The desired value for $\mu(\mathbf{\Phi}, \mathbf{\Psi})$ is as close to 1 as possible.

There are three different types of $\mathbf{\Phi}$ that are routinely examined: Gaussian, Type I; Bernoulli ± 1, Type II; and a random subset of the identity matrix, Type III. These $\mathbf{\Phi}$ structures and their naming convention are discussed and analyzed in Gürbüz et. al. [12]. For the sampling bound and mutual coherence calculations, the $\mathbf{\Psi}$ matrix will be have sampling parameters $N_f = N_t = 401$, $N_{\boldsymbol{l}_s} = 20$, and $N_{\boldsymbol{l}_t} = 20 \times 20 = 400$.

2.2.1.1 Time-pulse $\mathbf{\Phi}$

Type I , where $\mathbf{\Phi}$ is a matrix whose entries come from a normal distribution $\mathcal{N}(0, 1)$, is widely used in the literature for CS. Also, since the time-pulse measurement is by definition sparse in the time dimension, using a $\mathbf{\Phi}$ that is spread out is beneficial because it is very different from $\mathbf{\Psi}$. Table 1 shows the coherence $\mu(\mathbf{\Phi}, \mathbf{\Psi})$ values for Type I, II, and III matrices. The value for Type I is 4.5 which means that recovery is possible if only approximately 3% of the total measurements are retained.

Type II matrices have entries taken from a Bernoulli ± 1 process. Type II matrices have properties similar to that of Type I matrices, in that it is spread out in the dimension where $\mathbf{\Psi}$ is sparse. This similarity leads to an even lower value of 2.5 for $\mu(\mathbf{\Phi}, \mathbf{\Psi})$, which corresponds to only requiring approximately 1% of the total measurements to recover a single target.

For Type III, $\mathbf{\Psi}$ is a random selection matrix, which does just what the name suggests, it takes a random subset of all the measurements. However, Type III matrices are not structured to work well with the time-pulse system because both matrices $\mathbf{\Phi}$ and $\mathbf{\Psi}$ are sparse in the measurement domain. For example, when a vector of length 50 with a sparsity of one is being sampled, the only way to guarantee that a random selection matrix will obtain any information about the signal, is to take all 50 samples. Type III matrices have a much higher value of 20 for $\mu(\mathbf{\Phi}, \mathbf{\Psi})$,

which corresponds to a requirement that over half the samples must be taken.

Table 1: Mutual coherence values for different $\mathbf{\Phi}$ with TPGPR.

Type	Description	$\mu(\mathbf{\Phi}, \mathbf{\Psi})$	$\approx M$
I	$\mathcal{N}(0,1)$	4.5	240
II	Bernoulli	2.5	80
III	Random Sampling	20	5200

The analysis of the different $\mathbf{\Phi}$ matrices is only important if they can be effectively applied during the data-acquisition process, through some type of hardware design, pulse specification, or creative sampling structure. As described in the work by Gürbüz et. al., Type I and Type II can be applied using hardware mixers and low-pass filters to allow for the inner product to be taken with a random signal. However, creating Gaussian random pulses at radar rates, generally GHz, is difficult. Creating the random signal vectors for Type II is far more reasonable using state machines [12]. Type III matrices are extremely easy to implement by subsampling in either the sensor position domain, as seen in Figure 14 where the vertical black lines correspond to $\boldsymbol{l}_s$ positions that are not sampled, or the time domain. Random subsampling could be ideal for a system where the movement was not constant, but the sampling time was. As long as the $\boldsymbol{l}_s$ positions are recorded accurately at each sample, the random spatial sampling would be effective.

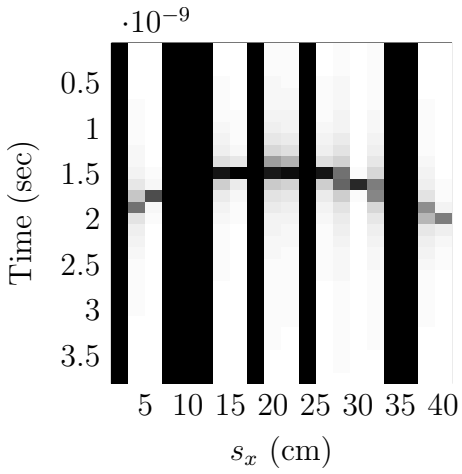


Figure 14: Subsampled time-domain measurement.

Random vectors can also be applied in another way, without requiring hardware to perform random inner products. Romberg introduced a method called Random Convolution, built on the idea that random Toeplitz and circulant matrices possess the same properties that allow for conventional random matrices to fit within the RIP [36, 49]. The added advantage of using these structured matrices, is that they can be efficiently applied with the Fast Fourier Transform (FFT), because they are the matrix representations of convolution. Equation (1) can be rewritten as a convolution,

$$r(t, \boldsymbol{l}_s; \boldsymbol{l}_t) = \frac{\rho}{S} p(t) * \delta(t - \tau(\boldsymbol{l}_s, \boldsymbol{l}_t, c, v)). \tag{38}$$

A generic signal can be sent from the receiver, $p(t)$. If $p(t)$ can be constructed to be a known, pseudo-random signal with either a $\mathcal{N}(0, 1)$ distribution or a Bernoulli ± 1, the projection of random vectors can be done through the convolution of the transmitted pulse and the target response with no additional mixing hardware required. The mutual coherence is going to be similar to that of a Type I or Type II $\boldsymbol{\Phi}$ matrix. Depending on the length of $p(t)$, the resulting measurement response will be, in effect, spread out in time. This will allow for a random sampling step to be performed after convolution to complete the compressive measurements. To recreate the effect of

convolution with $p(t)$ and subsequent random sampling for $\boldsymbol{\Psi}$, all that needs to be done is to apply a Toeplitz matrix, $\boldsymbol{P}$, to the left of $\boldsymbol{\Psi}$ followed by the application of a Type III $\boldsymbol{\Phi}$ matrix. The matrix $\boldsymbol{P}$ is built with a generating vector, $\boldsymbol{p}$, corresponding to the discretization of $p(t)$.

Using the random convolution method can dramatically reduce the complexity of the hardware in the GPR. In fact, GPRs already exist that use pseudorandom M-sequence pulses as the transmitted signals [53]. Also, it has been shown that pseudorandom sampling in the time domain is an effective strategy for typical GPR systems because it can eliminate the need for using a signal delay line [54, 55]. A framework for the design of a CS TPGPR will be explored in detail in Section 2.5. The random sampling could be done in a much shorter sweep time than the conventional sequential sampling while achieving similar accuracy. The use of random convolution and a random sampling GPR make compressive sensing seem as though it was created directly with a radar application in mind. The random convolution method could be paired easily with random spatial sampling for additional data-acquisition reductions.

2.2.1.2 Stepped-frequency $\boldsymbol{\Phi}$

The same analysis can be done for the SFGPR case as in the previous subsection for the TPGPR case. However, it is clear from analysis of Table 2 that all sampling structures for $\boldsymbol{\Phi}$ will work well. The major advantage here is that Type III matrices, which are by far the easiest to implement in a practical system, actually work the best. These matrices are completely incoherent with the $\boldsymbol{\Psi}$ matrix created for SFGPR and this is not surprising, because $\boldsymbol{\Psi}$ is very similar to a Fourier matrix, which is very dissimilar to an identity matrix. Because of this fact, there is simply no reason in this scenario to use anything other than Type III matrices. The dramatic reduction in sampling size seen by using a Type III matrix also has the advantage that it can easily and dramatically reduce the data-acquisition time for SFGPR, which can be a

constraint large enough to discourage its use in practical systems.

Table 2: Mutual coherence values for different $\mathbf{\Phi}$ with SFGPR.

Type	Description	$\mu(\mathbf{\Phi}, \mathbf{\Psi})$	$\approx M$
I	$\mathcal{N}(0,1)$	3.2	135
II	Bernoulli	3	120
III	Random Sampling	1	15

2.3 Simulation Using Functional Dictionary

The forward $g_{\mathbf{\Theta}}$ and adjoint operators $g^H_{\mathbf{\Theta}}$ can be used in different ways to perform inversion of the GPR measurements. In some cases the operators are applied only once, or a few times, in other algorithms thousands of times. This section is split into three important subsections. Subsection 2.3.1 directly compares the computation times of the three algorithms with and without exploiting translational invariance. Only 2D examples can be run in this case because the 12 GByte computer used to run the simulations was unable to run the 3D inversions using explicit matrices. Subsection 2.3.2 shows the different algorithms successfully being inverted in 3D while comparing their capabilities in dealing with compressed measurements. The 3D compressed algorithm simulation helps show the advantages of using an algorithm that looks for sparsity over those, like BP, that do not. The last simulation in subsection 2.3.3 is set up to compare solving a full 3D inversion exploiting translational invariance, and creating a 3D image out of 2D slices using the explicit matrix method. The 2D slice inversion is unable to take advantage of the full synthetic aperture, and will introduce imaging artifacts that are not present in the full 3D inversion.

2.3.1 2D comparison to previous methods

The easiest way to compare the new functional methods with older explicit ones is through small 2D simulations. 2D simulations are used because 3D simulations with

explicit enumeration are impossible; they quickly scale beyond the storage capabilities of the computer used for the experiments. For example, a 3D problem, solved using the explicit matrix method, with equal N discretizations, and a computer with 4 GBytes of dedicated data memory could have a maximum N of approximately 28. Increasing the data memory size to 12 GBytes only brings this value up to 33, and this assumes the entire data memory is used only for $\mathbf{\Psi}$. However, using the functional method enables a maximum N of 150 in the 4 GBytes computer and 196 in the 12 GBytes computer.

Figure 15 compares the measured times for a 2D GPR problem run using BP and OMP, using different size spatial dimensions, $N = \sqrt[5]{N_{\boldsymbol{l}_s} N_{\boldsymbol{l}_t}}$, and fixed $N_\omega = 401$ frequencies, to show how the algorithms scale as the dimensionality of the image space increases. Functional BP (fBP) is around 15 times faster than BP for $N = 70$. Functional OMP (fOMP) is also around 15 times faster than OMP and is actually about four times faster than BP for $N = 70$. These values are again just for the 2D inversion where there is only one dimension containing the redundancy, in 3D there are two. To see what the gains would be for the 3D problem, the approximate savings ratios for time and storage are calculated for the 2D and 3D solvers in Figure 16. To check how close the approximation is to reality, the ratios for experimental processing for $N = 70$ came out to be about 15 times, Figure 16 shows about 35 times for the 2D processing at $N = 70$, which is not going to account for any overhead or lower order processing times. It would be safe to say that there is an order of magnitude reduction. This means for a 3D problem with size $N = 70$ that is solved with one of the discussed algorithms, there would be a reduction of approximately three orders of magnitude to the time and storage by using the functional algorithms. The functional and explicit models are also compared using a CS algorithm in Figure 17, and again the functional method is much more time efficient by about a factor of five at $0.2{\times}10^6$ measurements and a factor of 15 at 10^6 measurements.

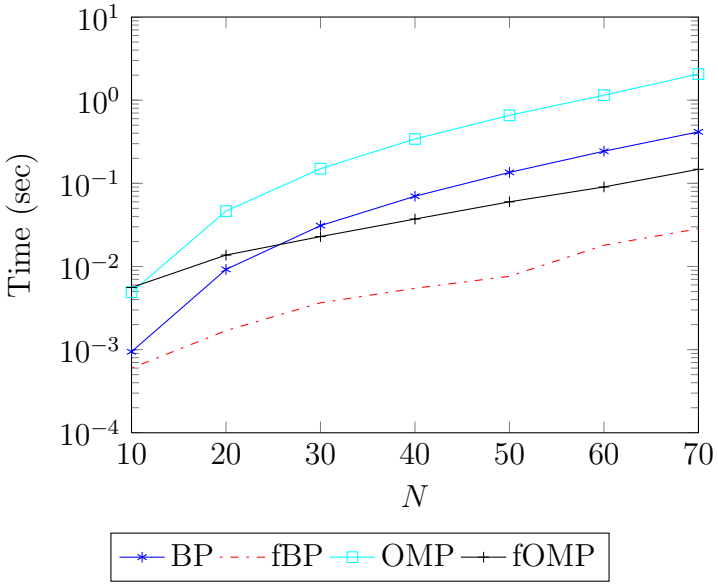


Figure 15: Comparing timing of BP with an explicit matrix to the functional implementation (fBP) and OMP using an explicit matrix to the functional implementation (fOMP).

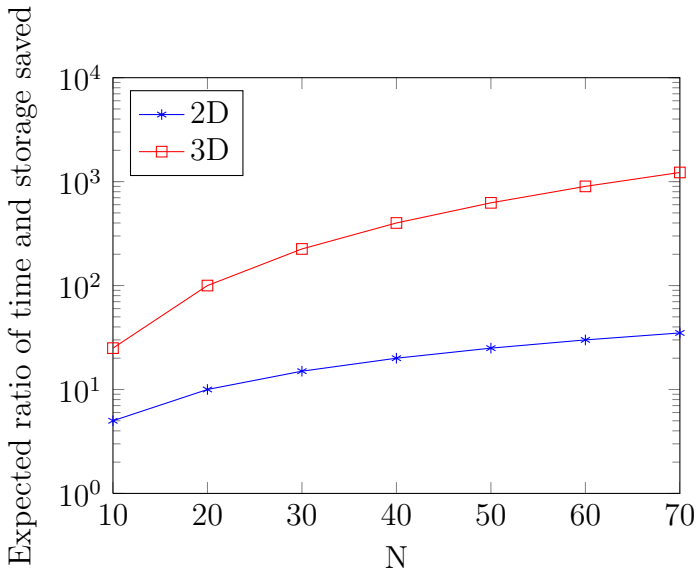


Figure 16: Approximately the ratio of storage and time saved by using the $g_{\mathbf{\Theta}}$ and $g_{\mathbf{\Theta}}^{H}$ instead of explicit matrix multiplication for different values of N.

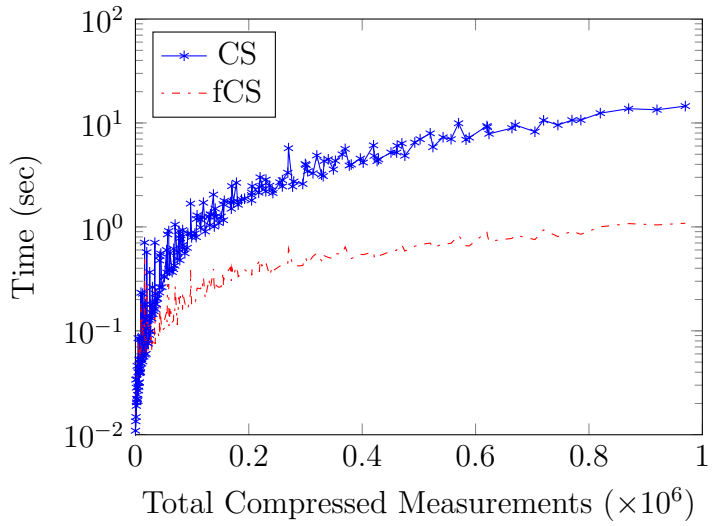


Figure 17: Timing comparison of functional and explicit CS for different numbers of compressed measurements.

2.3.2 3D compressed simulations

The slow data-acquisition time of SFGPR have led to heightened interest in algorithms that use compressed measurements. The following simulated example will only use compressed algorithms. Compressed algorithms are those at which a sampling scheme is applied to the measurements, and the dictionary before use in the algorithm. For this application, the same random selection of stepped frequencies is taken at each individual scan position, $\boldsymbol{l}_s$, along the synthetic aperture.

The simulation run was on a problem size that would be reasonable in a real application. The image grid has a resolution of 2 cm and has 60 equal discretizations for x_t, y_t, and z_t giving a total number of image pixels equal to $N_{\boldsymbol{l}_t} = 60^3 = 216,000$. As mentioned previously, the scan positions $\boldsymbol{l}_s = (x_t, y_t, 0)$, where $\alpha = 0$ and $\Delta x = 2\,\text{cm}$, allow for the translational invariance to be present. The possible frequencies, ω, used in the data collection were 379 frequencies in the range $2\pi(500\,\text{MHz})$ to $2\pi(8.06\,\text{GHz})$. The compression takes place in the selection of a random set of the same ten frequencies at each $\boldsymbol{l}_s$. The total explicit dictionary size for this problem would be

$N_{\boldsymbol{l}_s} N_\omega \times N_{\boldsymbol{l}_t} = 60^2(10)\times 60^3$ which turns out to be approximately 60 GBytes. The computer being used only has 12 GBytes of memory, so directly solving this problem with an explicit dictionary matrix is impossible without calculating the responses during run time in a loop. However, using the translational invariance exploit removes the need to store x_t and y_t and the size of the dictionary becomes $\mathcal{O}(N_{\boldsymbol{l}_s} N_\omega \times N_{z_t}) = 60^2(10)\times 60$ which is approximately 17 MBytes, a reduction by a factor of almost 10^3! The simulation environment consisted of three point targets with an SNR of 15dB. Figure 18 shows the results of the imaging with three different compressed algorithms. Two thresholds were used in the compressed BP (CBP) case in Figure 18(a) and (b), and it can be seen that changing the user defined thresholding can dramatically affect the output images. The compressed OMP (COMP) and CS algorithms in Figure 18(c) and (d), respectively, achieve exact reconstruction.

The measured timing statistics for this simulation can be seen in Table 3. A large portion of the time it takes to run the algorithms is associated with the time it takes to run the forward and adjoint operators. The ratio of the amount of time per function call is shown in the third column of Table 3. The time it takes to run COMP is almost exclusively dependent on the speed of $g_{\boldsymbol{\Theta}}$ and $g^H_{\boldsymbol{\Theta}}$, whereas a more complicated algorithm, CS, has a little more time devoted to overhead. For algorithms like CS, where the dictionary is going to be applied hundreds of times, any efficiency increase to the dictionary is going to scale linearly.

Table 3: Timing statistics for 3D simulation.

Algorithm	Time (sec)	Number of $g_{\boldsymbol{\Theta}}$ or $g^H_{\boldsymbol{\Theta}}$ calls	ratio Time/calls
CBP	0.13	1	0.13
COMP	1.35	12	0.11
CS	108.42	423	0.25

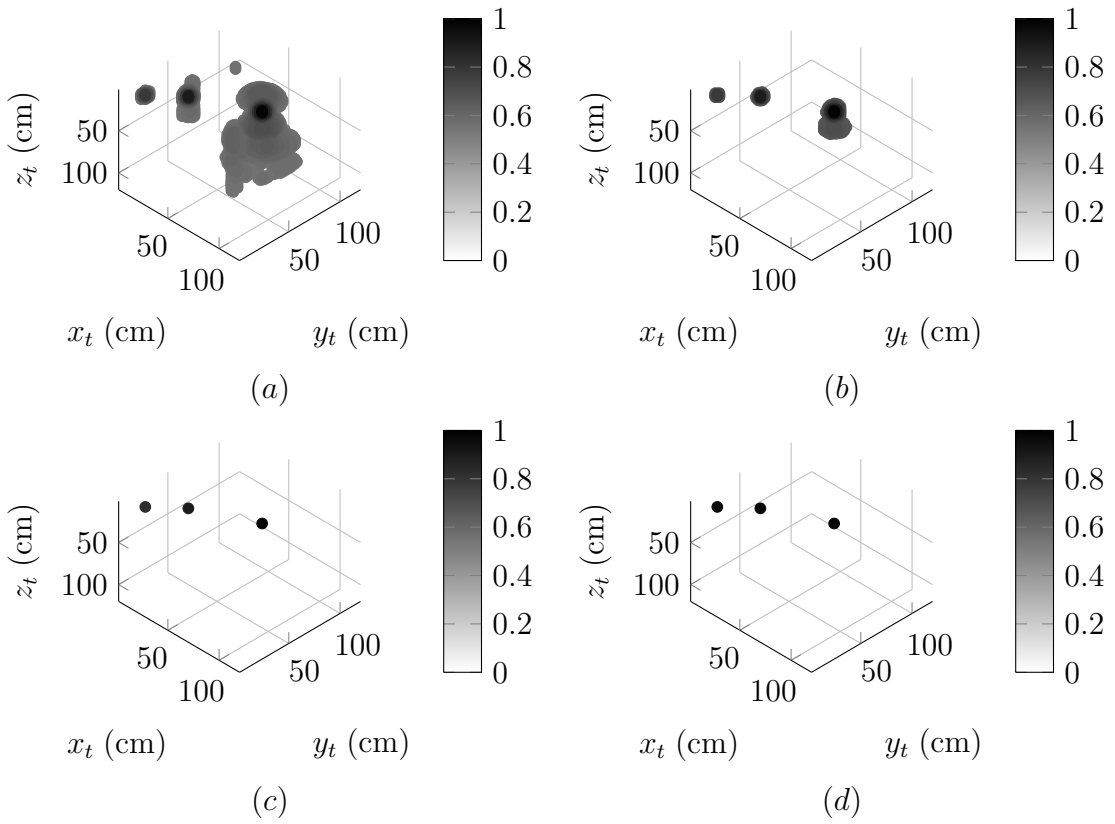


Figure 18: Images created from a simulated environment with (a) CBP with 5% threshold, (b) CBP with 10% threshold, (c) COMP, and (d) CS.

2.3.3 Comparing full 3D to sliced 2D

The computational inefficiency of previous methods has prevented the direct implementation of 3D inversion, so a sub-optimal approach was taken by solving small 2D slices and concatenating them together to form a 2.5D image [13]. If we compare the two approaches, it is easy is to show why solving the large 3D problem is important. It is important to note that the functional method can be done in 2D as well, and would be more time efficient than using the explicit method, as was shown in subsection 2.3.1.

This experiment consists of randomly placing two targets in a 3D volume and using CS to invert the measurements and create an image. The frequency range used is 0 MHz to 5.02 GHz with $N_\omega = 158$ equally spaced frequencies. The scan positions

correspond to a co-located transmitter and receiver. The transmitter locations are uniformly spaced in a 2D square from $y_s = -96$ cm to 90 cm at 6 cm intervals, giving $N_{y_s} = 32$, and similarly from $x_s = -96$ cm to 90 cm at 6 cm intervals, giving $N_{x_s} = 32$. The horizontal target locations are in the range $y_t = -90$ cm to 84 cm at 6 cm intervals, giving $N_{y_t} = 30$; likewise for x_t. The fact that the scan positions and the target locations have the same horizontal discretization leads to the translational-invariance simplification. The depth locations need not be constrained, but for this experiment, z ranges from 270 cm to 420 cm deep at 6 cm intervals, giving $N_{z_t} = 26$. The number of compressed measurements, N_{cm}, is selected between 0.1% and 2.4% of the total measurements $N_m = N_\omega N_{\boldsymbol{l}_s} = 158 \times 32^2 \approx 10^5$. The number of discretizations in the target location parameter space is $N_p = N_{\boldsymbol{l}_t} = 30^2 \times 26 \approx 10^4$. Without using CS or the functional representation, the explicit matrix would be of size $N_m \times N_p \approx 10^9$. However, employing compressive sensing and the functional representation, N_m becomes N_{cm}, $N_p = 26$, and the total storage requirements for a matrix using 2.4% of the total measurements becomes $N_{cm} \times N_p = 2400 \times 26 \approx 10^5$, a reduction of four orders of magnitude.

Figure 19 shows the detection accuracy advantages that are available when solving the full 3D imaging problem as opposed to imaging the volume with many 2D slices. When using 2D slices and not solving the full 3D problem, the synthetic aperture in the x_t dimension is not being directly exploited. The sparsity in the 2D slice image from Figure 19(b) shows the resolution issues in x_t. The full 3D inversion that is made possible by using the functional representations $g_{\boldsymbol{\Theta}}$ and $g^H_{\boldsymbol{\Theta}}$, is shown in Figure 19(c). The full 3D inversion provides a much higher resolution image in x_t than using 2D slices. The higher resolution is a byproduct of the fact that the full 3D inversion takes advantage of the full 2D array aperture. The accuracy metric used to analyze the solutions is the Earth Mover's Distance (EMD) [42]. The EMD is used, because unlike mean-squared error, EMD takes the support into account when calculating the

error. Targets with small error in location have a lower EMD than they would with MSE. An SNR analysis showing the EMD reduction produced by solving the full 3D inversion instead of the 2D slices is found in Figure 19(d).

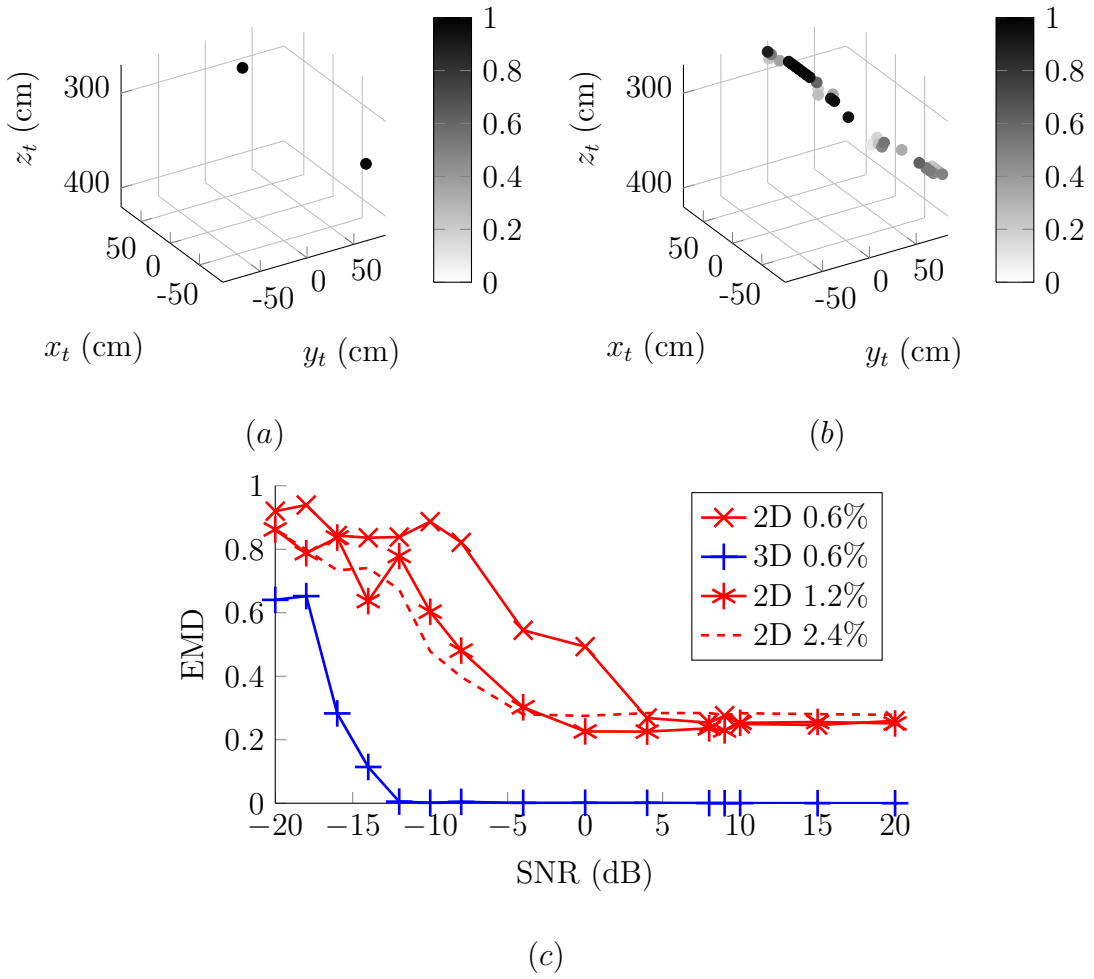


Figure 19: (a) Full 3D CS solution using FFT method with exact reconstruction, (b) solution using 2D slices, (c) EMD comparison of 2D slice and full 3D solutions.

2.4 Applied Performance

This section takes the experiment described in Counts et. al., and compares the methods described in Gürbüz et. al., with those of Krueger et. al. [3, 13, 46, 47].

The frequencies, ω, used in the data collection were $2\pi(60\,\text{MHz})$ to $2\pi(8.06\,\text{GHz})$ at $N_\omega = 401$ equally spaced intervals. The first experiment is a target above ground,

and the second experiment will be a collection of targets buried in sand. In order to make a comparison, an explicit dictionary CS method that images the 3D area with a collection of 2D slices and the proposed CS method that images the full 3D area will be used.

The solver that is used to perform the ℓ_1 minimization is again SPGL1 because of its computational efficiency and because it allows for functional representations of the forward and transpose matrix operators with BPDN [50]. The user chosen parameters that must be set in SPGL1 for the BPDN is the tolerance ϵ_2 from (36). The tolerance ϵ_2 for the full 3D solver was chosen using a customized CV algorithm which not only looks for divergence in error in the test and evaluation sets, but also looks for dramatic jumps in the number of nonzero targets. Increasing ϵ_2 will increase sparsity, but if it is too large, some targets will be missed. Decreasing ϵ_2 will increase the probability of including all present targets, but will also increase the probability of false alarms.

2.4.1 Air-target experiment

The air-target setup can be seen in Figure 20(a) [13]. There is a 1-in metal sphere placed on a Styrofoam platform off the ground and the radar is scanned above it. The 3D time-domain measurements of this experiment are seen in Figure 20(b) and the target can easily be seen even in the measurements. The discretizations used for this problem are as follows: x_s and y_s were both taken from $-50\,\text{cm}$ to $48\,\text{cm}$ at $2\,\text{cm}$ increments, x_t and y_t taken from $-48\,\text{cm}$ to $46\,\text{cm}$ at $2\,\text{cm}$ increments, and z_t taken from $40\,\text{cm}$ to $60\,\text{cm}$ with $1\,\text{cm}$ increments. This would correspond to an explicit dictionary size of $[379 \times 50^2 \times 48^2 \times 21]$ which is on the order of 10^{10}. Using CS and getting a reduction in frequencies to about 50, the dictionary is still on the order of 10^9. This means that to use an explicit dictionary, this problem must be solved in 2D slices, which does not harness the synthetic aperture in the extra scan dimension and

will lower resolution in that spatial dimension. If the functional method is used, there is an elimination of both horizontal spatial image locations in the dictionary for an uncompressed size of $[379 \times 50^2 \times 21]$ which is on the order of 10^7 and a compressed size with 50 frequencies of about 10^6. Using CS and the functional method garner a reduction of three orders of magnitude from the explicit dictionary CS case.

The solution using the explicit dictionary to create 2D slices and build them into a 3D image is seen in Figure 20(c). The resolution in this image is acceptable in y because this is the dimension where the 2D CS is calculated, but the x has lower resolution because the slice method does not exploit the synthetic aperture in that dimension at all. Finally, the full 3D solution using the functional representation of the dictionary can be seen in Figure 20(d). The 3D method is much sparser than the 2D slice solution again because it can take advantage of the synthetic aperture in both scan dimensions, not just one of them.

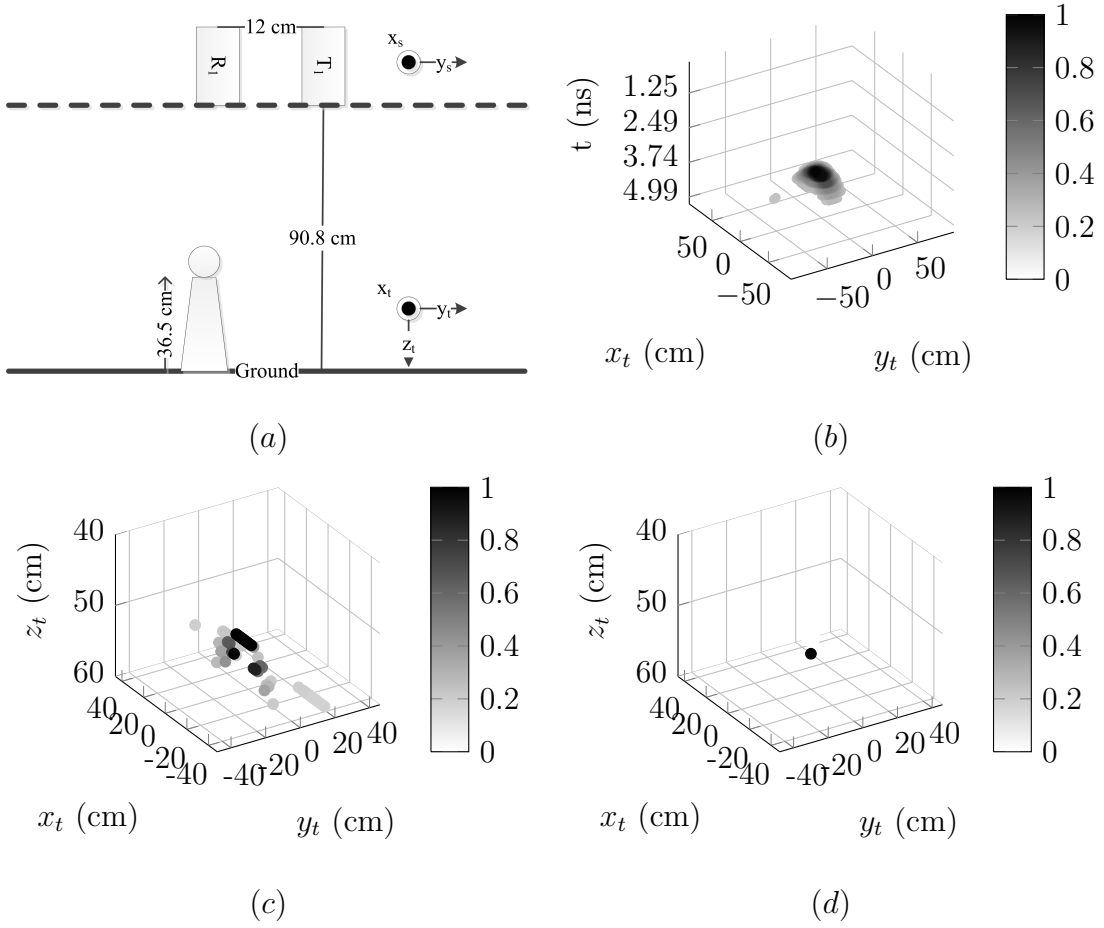


Figure 20: Air experiment for 1-in metal sphere (a) setup, (b) time-domain measurements, (c) solution using 2D slices, and (d) full 3D CS solution using the FFT.

2.4.2 Subsurface-target experiment

The final experiment is one that was performed and documented previously using standard BP in Counts et. al. [3]. The acquisition array setup for subsurface imaging can be seen in Figure 21(a) and the ground-truth location of the targets can be seen in Figure 21(b). A 2D slice CS algorithm was employed for inversion to avoid the computational inefficiencies in Gürbüz et. al. [13], but we saw in the previous section the issues that can arise from that type of inversion. However, the 2D slice method could also be improved by exploiting the translational invariance if it was desired to still be used. It would be faster and more storage efficient than using the explicit

method to calculate each slice because each slice would garner a computational time improvement to what was shown in subsection 2.3.1. The purpose behind inverting this data again is to show that a real-world experiment that was previously too computationally intense to invert with full 3D CS, can be completed with the exploitation of shift invariance. The discretizations used for this problem are as follows: x_s and y_s were both taken from $-60\,\text{cm}$ to $60\,\text{cm}$ at $2\,\text{cm}$ increments, x_t and y_t taken from $-58\,\text{cm}$ to $56\,\text{cm}$ at $2\,\text{cm}$ increments and z_t taken from $1\,\text{cm}$ to $20\,\text{cm}$ with $0.5\,\text{cm}$ increments. Figure 21(c) and (d) show the top view of the created images using uncompressed BP with 379 frequencies and CS with 100 frequencies. Both of these were computed with the functional implementation of the forward and adjoint operators. It is very difficult to distinguish where some of the weaker targets are in the BP image, but it becomes much clearer in the CS image. For example, the mines at $(x_t, y_t) = (-45, 5)$ and $(x_t, y_t) = (0, 50)$, and the cylinder at $(x_t, y_t) = (45, 50)$. The imaging improvement of CS over BP combined with the fact that CS can be done with much fewer measurements, is a substantial reason as to why CS would want to be used over BP. In the work by Gürbüz et. al., full 3D inversion for this experiment was impossible with CS, but now with the modification to the storage and application of $\mathbf{\Psi}$ it can be done.

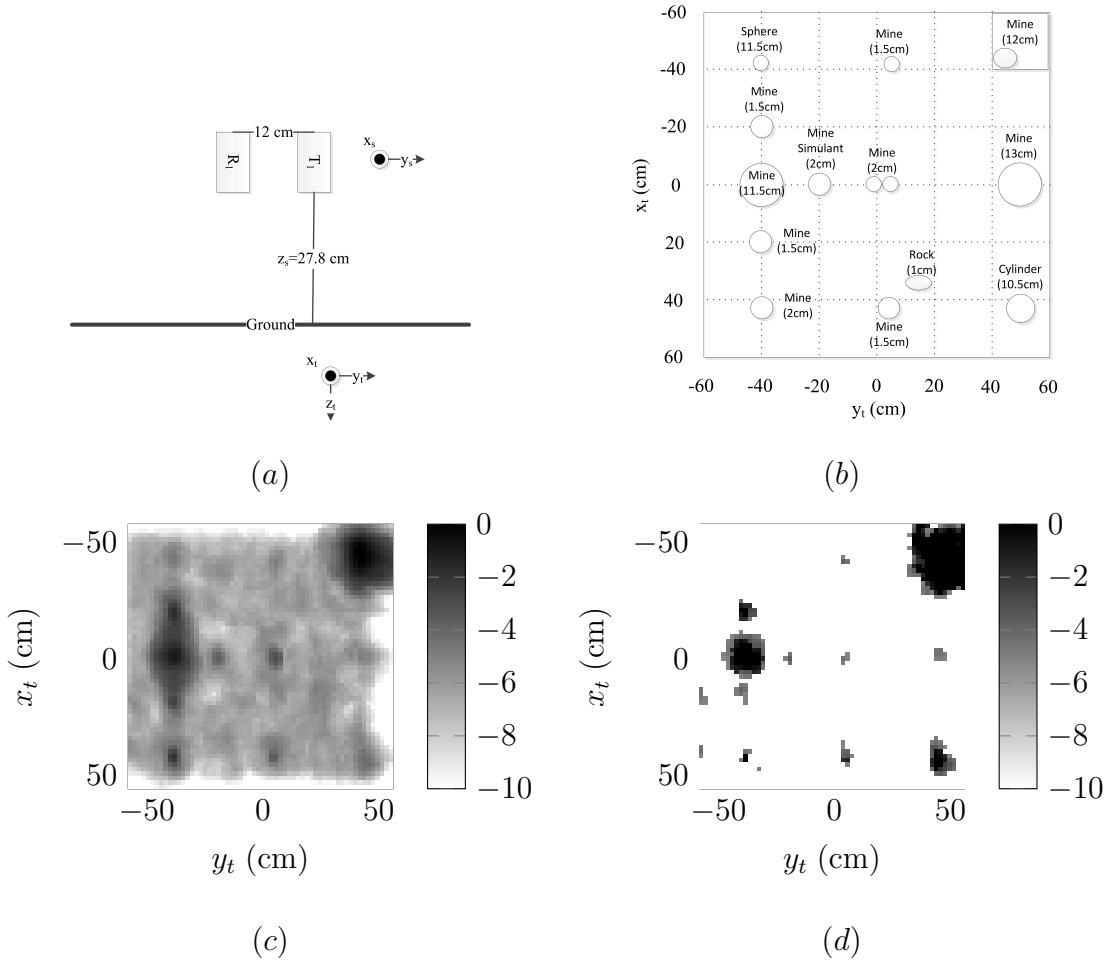


Figure 21: Underground experiment for multiple objects (a) sensor setup, (b) target locations, the values in parentheses correspond to the depths of the individual targets, (c) solution using BP, and (d) full 3D CS solution using functional $g_{\boldsymbol{\Theta}}$ method.

A problem with imaging these types of targets is that some of them are much larger than others and a point target model is not an ideal way to model them to promote sparsity in the images. By examining Figure 21(d), each pixel corresponds to a point target, and each point target is counted to determine the sparsity of the image. If a more accurate target model were available for the larger mines, the CS solution would perform much better while requiring fewer measurements. Fewer measurements would be required because the sparsity would dramatically decrease

by counting the number of mines as opposed to point targets. This is because the number of measurements required to provide a high probability of accurate recovery is dependent on the sparsity of the image, which can be seen in (15).

2.5 CS TPGPR Hardware Framework

The framework for designing a CS TPGPR system is fairly straight forward combining many common TPGPR techniques and established CS properties for the inversion algorithm. The main components that will be taken advantage of include: using randomly generated M-sequence pulses, pseudorandom sampling, and CS properties of Toeplitz structures and random convolution for both the application of the pulse, $p(t)$, and for the shift invariance of the spatial domain [36, 47, 49, 53].

2.5.1 Theoretical setup

As shown in Table 1, random sampling directly with a delta pulse GPR is very bad for the coherence, and thus the compressive sensing, but Type II matrices work very well in this regard. The first step is to understand how the construction of $p(t)$ can effect the GPR's detection abilities. An example of an M-sequence ± 1 signal can be seen in Figure 22. It is important to note that increasing the length of $p(t)$ can spread the response out in the time domain, giving a higher probability that there will be information selected if randomly spaced time samples are taken. The spreading of the response with different N_p length $p(t)$ can be seen in Figure 23.

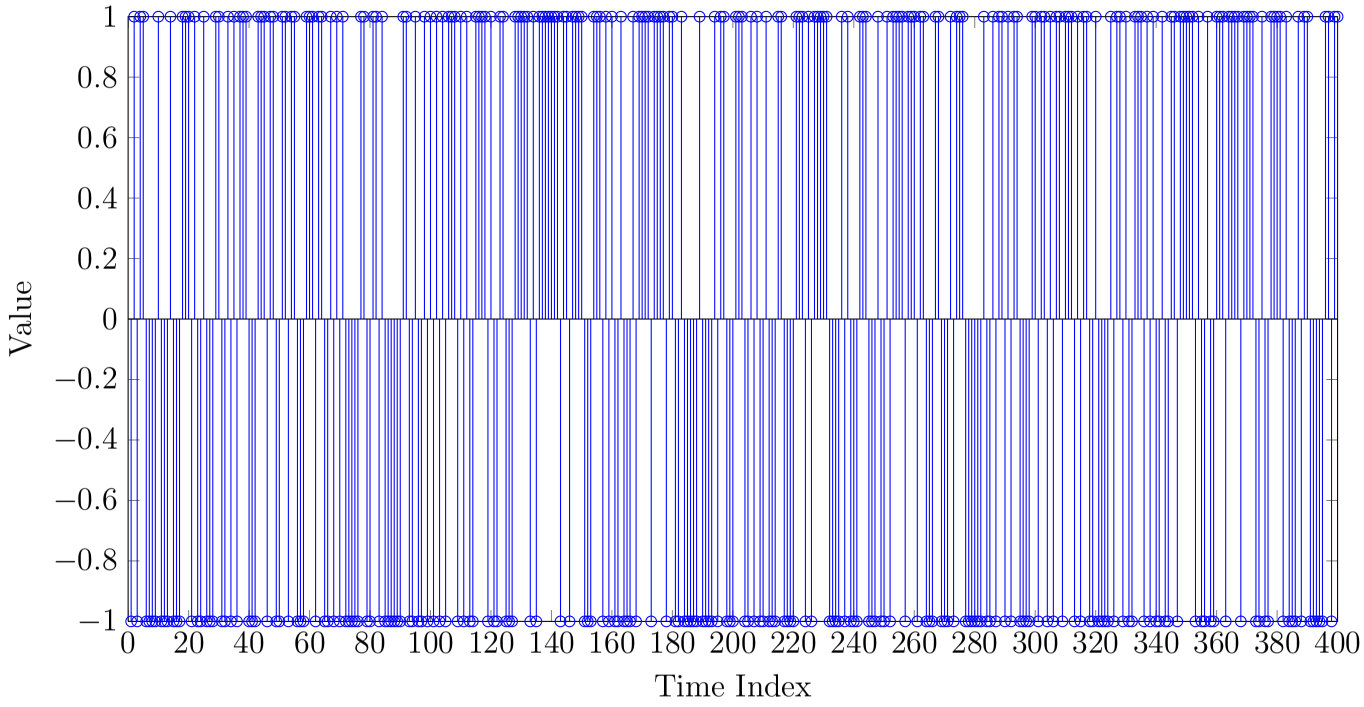


Figure 22: Example $p(t)$ of length $N_p = 400$.

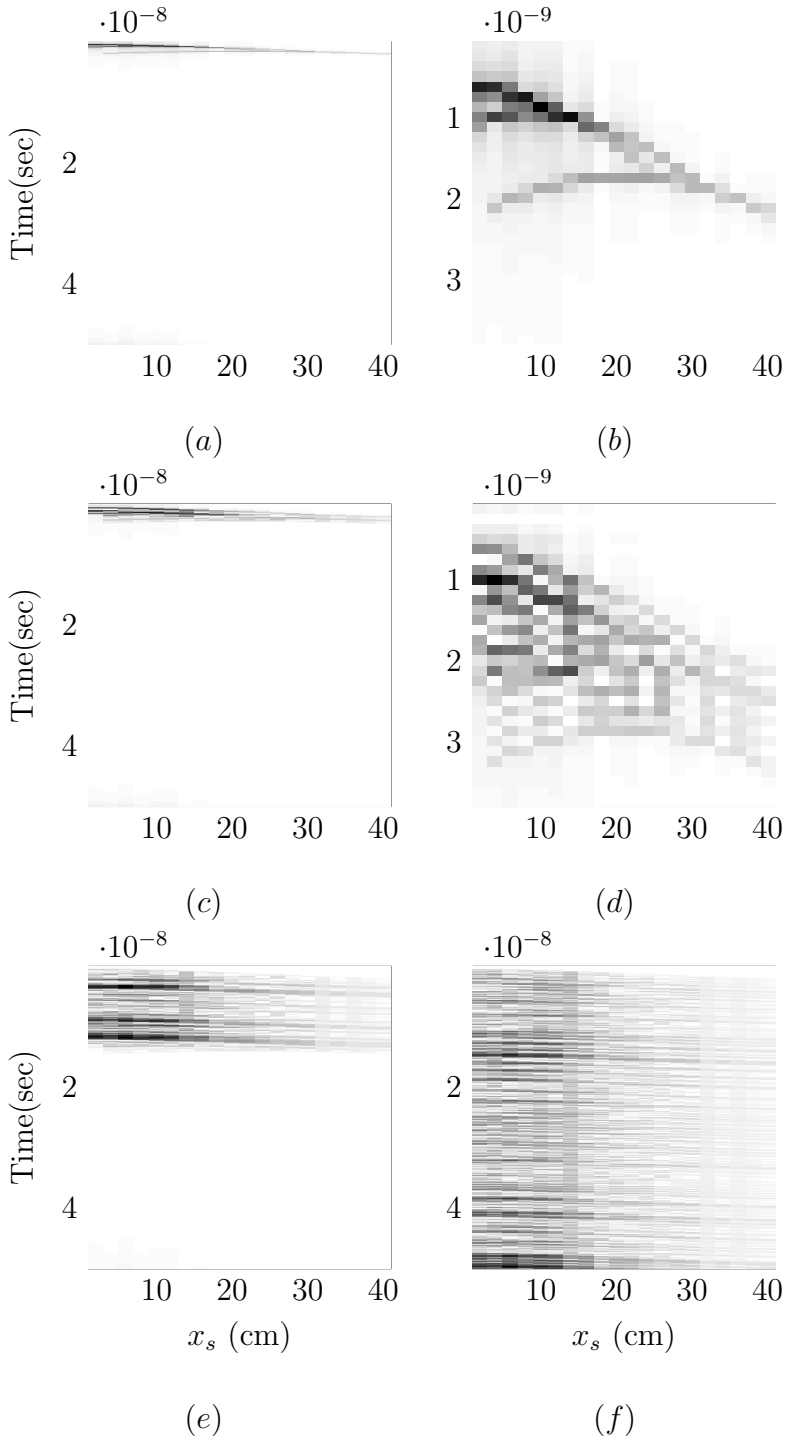


Figure 23: Showing simulated measurements with different $p(t)$ lengths. (a) $N_p = 1$, (b) $N_p = 1$ zoomed in, (c) $N_p = 10$, (d) $N_p = 10$ zoomed in, (e) $N_p = 100$, and (f) $N_p = 400$.

Increasing the probability of information in each sample is what can decrease the coherence of the dictionary. Again, lowering the coherence decreases the required number of random samples needed for CS to be valid. Using random sampling has already been shown to be a more efficient form of sampling for TPGPR than sequential because there is no need for a delay line. TPGPR systems typically use equivalent-time sampling and not real-time sampling so that they only take one sample per pulse, and thus must send out many pulses to construct one response signal [56]. For example, to create a response signal of length N_t, N_t pulses have to be sent. Random sampling allows for these N_t pulses to be sent in quicker succession, but they still all have to be sent. But, combining the M-sequence radar and only collecting $N_{ct} \ll N_t$ compressed time samples, creates a compression matrix $\mathbf{\Phi}$ that fits the RIP. A plot of the coherence values μ for different N_p length $p(t)$ pulses can be seen in Figure 24. Figure 24 confirms that the longer the pulse, the fewer measurements will need to be collected for accurate recovery, and thus lowering N_{ct} and further decreasing data acquisition time. The same $\mathbf{\Psi}$ is used for the calculation of the coherence as was used to compute the values in Table 1.

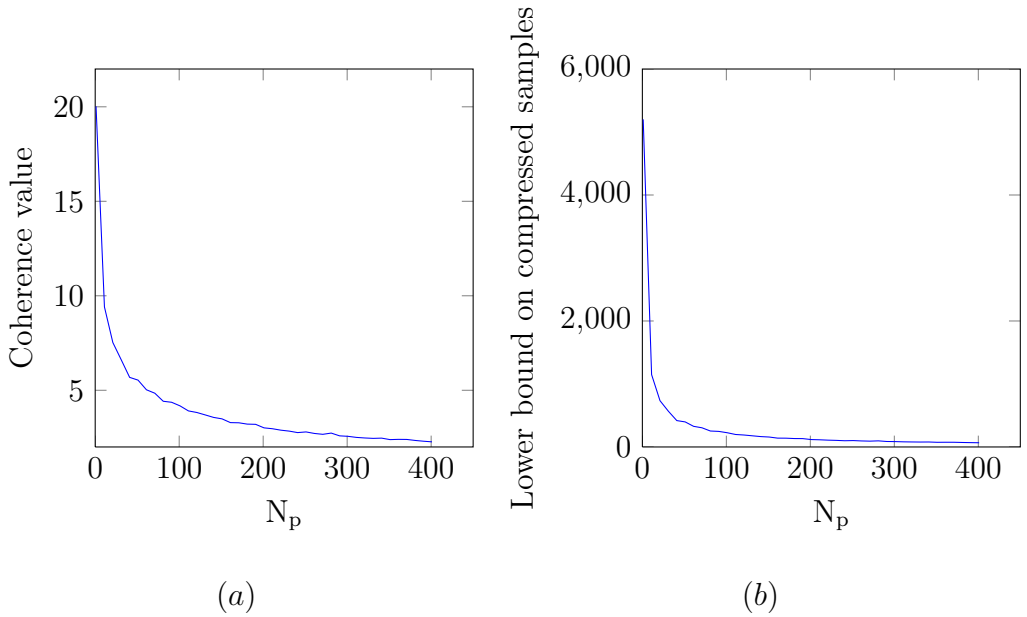


Figure 24: Calculations to show the (a) coherence and (b) lower bound minimum samples for CS accuracy using different length $p(t)$ pulses.

Once the response has been randomly sampled, looking at the resulting subsampled response can also show why longer pulses can be better for detection accuracy. Figure 25 shows four separate subsampled responses when different length pulses are used. Figure 25(a) and Figure 25(b) show the subsampled response when a shorter, length 1 and length 10 respectively, $p(t)$, is used. There are many rows, time samples, that have little to no useful data which mean the samples are worthless. Figure 25(c) and Figure 25(d) show the subsampled response when a longer, length 100 and length 400 respectively, $p(t)$, is used. The subsampled responses acquired from using a longer $p(t)$ have a much lower probability of including "empty" samples.

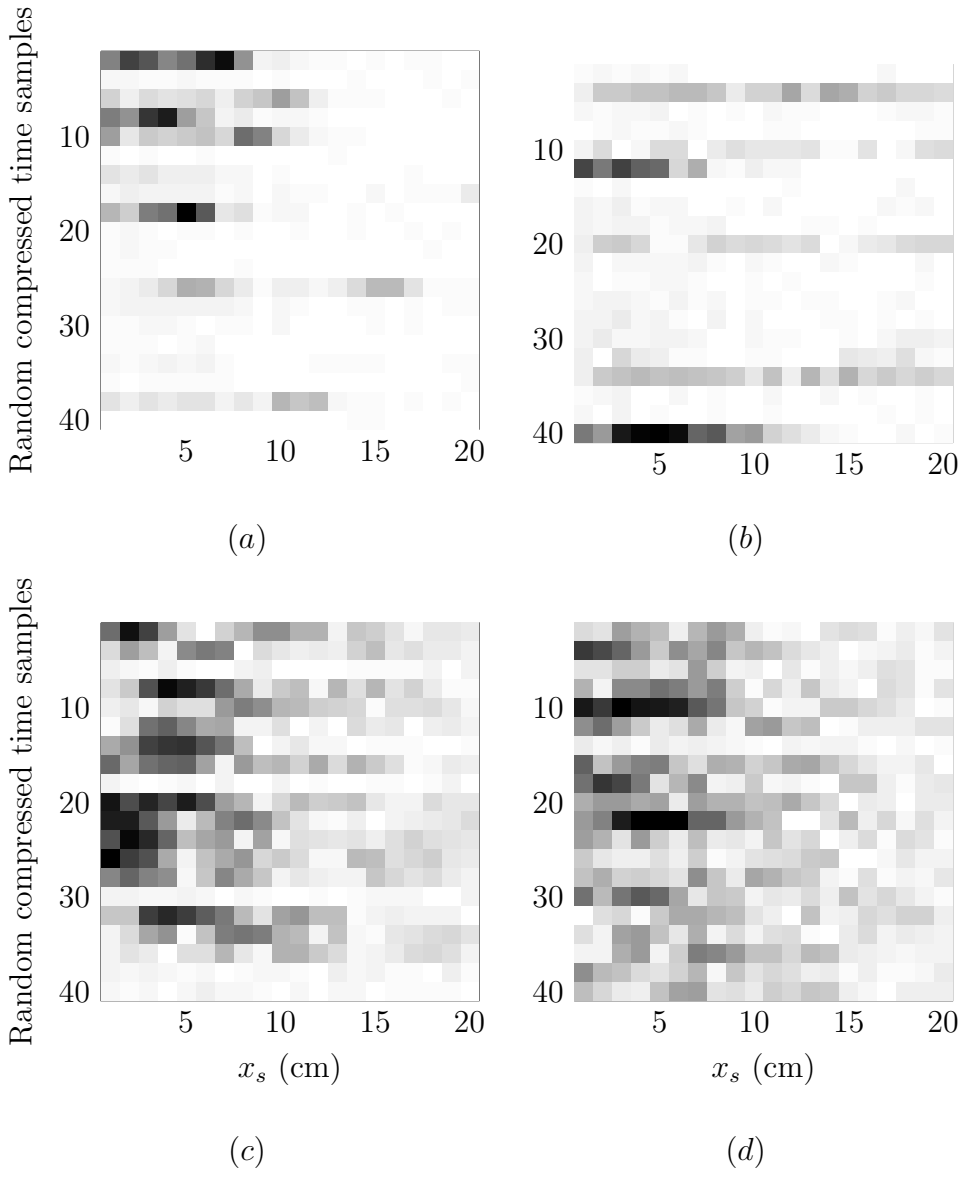


Figure 25: Showing randomly sampled measurements with different $p(t)$ lengths. (a) $N_p = 1$, (b) $N_p = 10$, (c) $N_p = 100$, and (d) $N_p = 400$.

2.5.2 2D simulations

A few simulations were run in order to check the validity of the theory in simple 2D problems, since the expansion to 3D is trivial and 2D is easier to work with and much more efficient for collecting large amounts of data. The simulation was to image a small, three target environment in 15dB SNR. The inversion algorithms are performed

in almost the exact same way as the SFGPR examples in the previous sections. The shift invariance is exploited by making sure that the same random time samples are taken at each scan position $\boldsymbol{l}_s$. All that as changed is that the response is in the time domain, instead of the frequency domain, but the properties to apply $g_{\boldsymbol{\Theta}}$ and $g_{\boldsymbol{\Theta}}^H$ still apply.

The compression algorithms are examined, CS and COMP. For the example solution images, $p(t)$ is length $N_p = 100$ and there are $N_{ct} = 25$ random, compressed measurements taken in the time domain. The solution images are seen in Figure 26 for both COMP and CS, and perfect reconstruction is accomplished in both. To check the importance of the length of $p(t)$ with respect to the number of compressed measurements, a large number of simulations were run to gather statistics. The EMD is once again used to determine accuracy. Figure 27 shows plots based on the number of compressed measurements. It is fairly obvious from Figure 27 that the EMD drops significantly as the number of compressed measurements increases. Also, something worth noticing, once $N_{ct} = 60$, both algorithms, CS and COMP, perform relatively equivalently. Since COMP is much more time efficient, there are certain situations when COMP would be preferred over CS for this reason.

The same data can be looked at slightly differently in Figure 28, where the N_p is held constant for each plot, and the number of compressed measurements is analyzed. Again, increasing the length of $p(t)$ reduces the number of samples required to get accurate recovery.

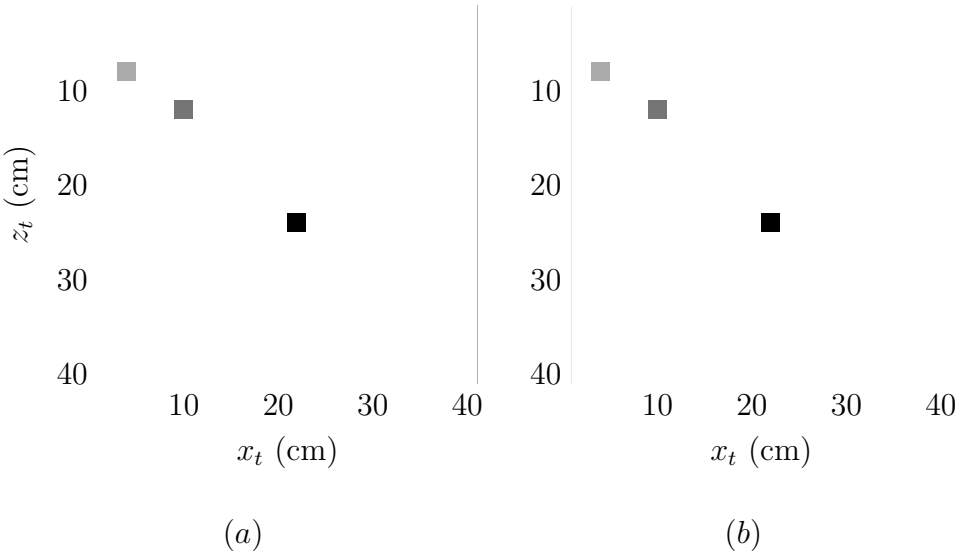


Figure 26: Solution images with two compression based algorithms (a) COMP and (b) CS.

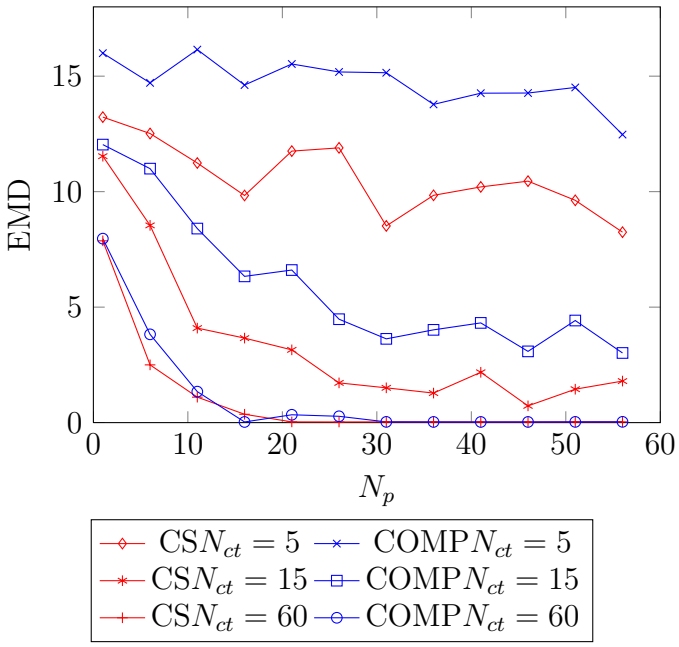


Figure 27: Comparing the required length of $p(t)$ based on the number of random samples desired.

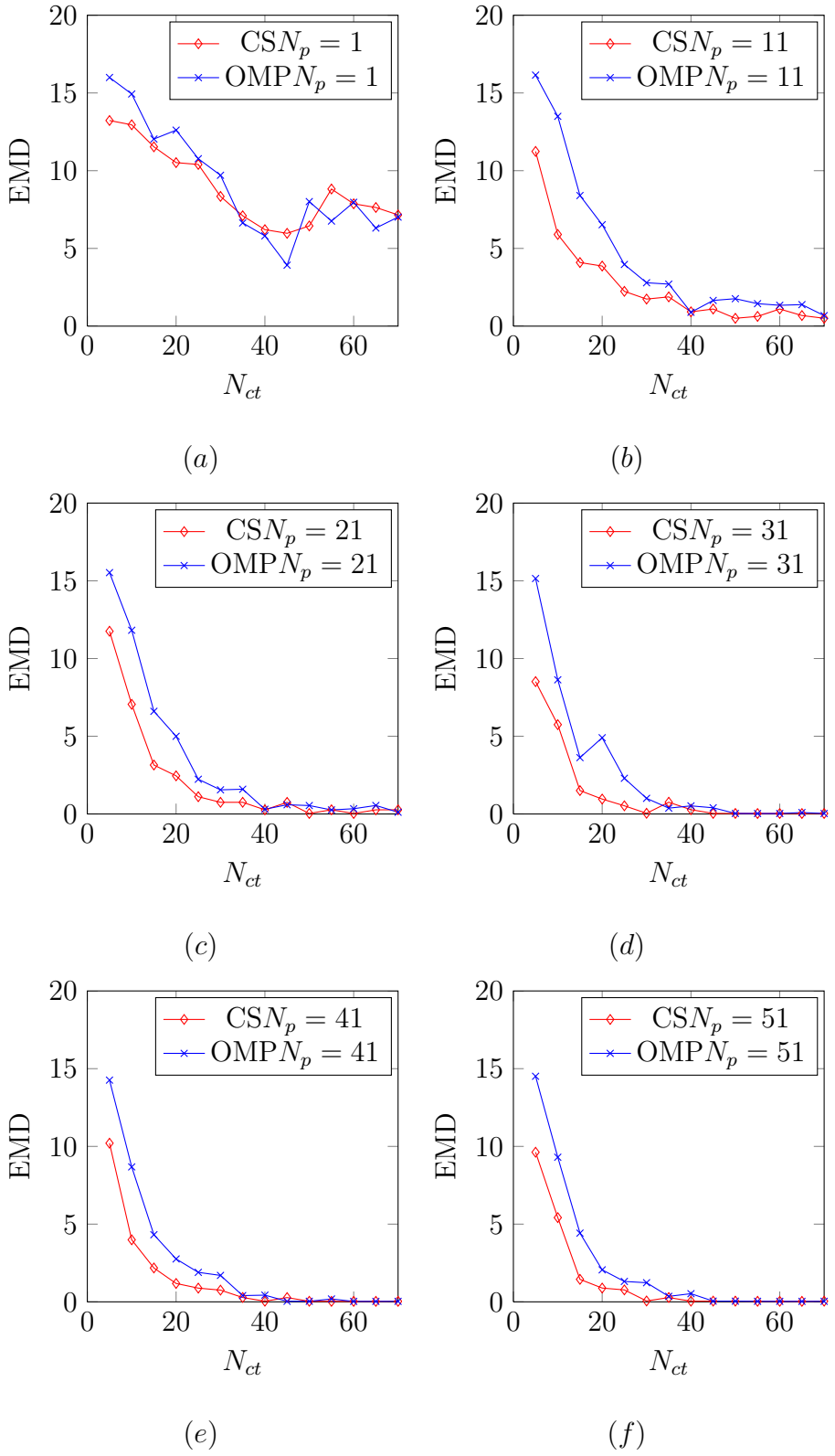


Figure 28: Comparing the required number of random time samples taken with the given $p(t)$ length.

2.6 Conclusions

An efficient restructuring of the storage and application of the dictionary used in GPR inversion algorithms has been discussed. The dramatic reduction in computation times and storage constraints allows for previously unsolvable problems to be solved. The computational reduction also increases the ability for these algorithms to be used on small, mobile devices which are very common in landmine detection.

CHAPTER III

INCREASING EFFICIENCY FOR THE EMI PROBLEM

The model used in the EMI system is more complicated than the point-target model used in the GPR. With the ability of the model to extract more parameters than just the location of a point reflector, the dimensionality of the EMI model scales up quickly, so methods for dealing with the dimensionality need to be developed for the imaging to be practical. With the scalability in mind, a change is proposed to the fundamentals of the dictionary setup. As discussed in previous sections of this thesis, each combination of parameters is stored in a single column of the dictionary $\boldsymbol{\psi}_t$. The amplitude of the response only represents the strength of the target and helps determine its presence. There has been some preliminary work where additional information can be stored in the amplitude of certain elements of a strategically designed dictionary to help obtain parameter information without requiring further parameter enumeration. Ekanadham et al. have shown a way to use a Taylor series or a cosine representation of a signal to interpolate small shifts in the signal without having to discretize the signal too finely [57]. Continuous parameter estimation is effective because not only does it reduce the storage requirements to represent a signal, but it can also reduce the modeling error. The modeling error reduction comes when the measured signal is not exactly equal to one of the discretizations in the parameter space, which is extremely common.

A new representation of the EMI data is proposed using a tensor representation, which can be thought of as a "tensor amplitude" [48, 58, 59]. The remainder of this chapter will be split into six sections. Section 3.1 shows how the "tensor amplitude" is introduced in the EMI response. Section 3.2 discusses the issues that can arise

with the data-acquisition setup. Section 3.3 explains how the "tensor amplitude" can be extracted using a semidefinite program. Section 3.4 and Section 3.5 show the results of some small simulations and controlled laboratory experiments respectively. Finally, a new sensing geometry is introduced in Section 3.6 and brief conclusions are given in Section 3.7.

3.1 Model Design

The model design is ultimately the most important part of the process because it relates the measurement process to the target parameters. The model needs to be able to be applied efficiently, while still maintaining a high level of accuracy. The basis for the model design in the EMI case can be found in [1, 4, 26, 60]. The data-acquisition system used is an essential part of model design. Once it is known how the measurements are going to be collected, it might still be necessary to approximate what the responses will be when certain parameters are set. In this research the measurements are acquired using one transmitter coil, and three receiver coils as discussed in Chapter 1, and seen in Figure 3. This section will be split into two subsections; subsection 3.1.1 will explain how the model can be converted from the frequency domain and into the discrete spectrum of relaxation frequencies (DSRF), and subsection 3.1.2 will explain how the dictionary using the DSRF can be dramatically simplified in a computational sense with the use of a "tensor amplitude."

3.1.1 Conversion to discrete spectrum of relaxation frequencies

The first step is to revisit the frequency domain response collected with the data-acquisition system described in Chapter 1. The sensor position, $\boldsymbol{l}_s$, only varies in x_s, and the three receive coils are aligned in the y dimension to help with extracting the 3D location of the target, $\boldsymbol{l}_t$. The response is recorded in the frequency, ω, domain. This acquisition leads to a system with measurements that depend on $\boldsymbol{l}_s$ and ω. The parameters to be extracted from the response are target location, $\boldsymbol{l}_t = (x_t, y_t, z_t)$;

target orientation, $\boldsymbol{o}_t = (\alpha_t, \beta_t, \gamma_t)$; and target type. For small targets, a simple dipole model can be used to represent the response of the system at coil c,

$$r_c(\omega, \boldsymbol{l}_s; \boldsymbol{l}_t, \boldsymbol{o}_t) = C\boldsymbol{g}_c^H(\boldsymbol{l}_s - \boldsymbol{l}_t)\boldsymbol{R}^H(\boldsymbol{o}_t)\boldsymbol{M}(\omega)\boldsymbol{R}(\boldsymbol{o}_t)\boldsymbol{f}(\boldsymbol{l}_s - \boldsymbol{l}_t). \tag{39}$$

where

$$\begin{aligned}\boldsymbol{M}(\omega) &= D_0\boldsymbol{\Lambda}_0 - \sum_{k=1}^{N_\zeta} D_k \left(\frac{j\omega/\zeta_k}{1 + j\omega/\zeta_k}\right)\boldsymbol{\Lambda}_k \\ &= \sum_{k=0}^{N_\zeta} D_k p(\omega, \zeta_k)\boldsymbol{\Lambda}_k.\end{aligned} \tag{40}$$

All of this work is using the measurement sensor described in Figure 3, so c will be 1, 2, or 3. The target type is embedded in the magnetic polarizability of the target, and it contains a unique response for every different possible object. This means creating a comprehensive dictionary is impossible, because there are an infinite number of possible targets. The dictionary would have to be created for only the highest probability targets. If a dictionary were to be created, as has been done previously in this thesis, with the frequency response, the dictionary would be,

$$\boldsymbol{\Psi}_c^\kappa(\omega, \boldsymbol{l}_s; \boldsymbol{l}_t, \boldsymbol{o}_t) = \boldsymbol{g}_c^H(\boldsymbol{l}_s - \boldsymbol{l}_t)\boldsymbol{R}^H(\boldsymbol{o}_t)\boldsymbol{M}(\omega)\boldsymbol{R}(\boldsymbol{o}_t)\boldsymbol{f}(\boldsymbol{l}_s - \boldsymbol{l}_t), \tag{41}$$

where κ would be the specific target type. The model complexity scales up at a dramatic rate determined by the number of measurements and parameters. It would be on the order of N^9 if each measurement and parameter is equally discretized where the target type is assumed to be one of the parameters.

A specific target can be decomposed into a sum of electromagnetic dipoles, and a graphical example of this decomposition can be seen in Figure 29. This is an important property because it allows for complex targets to be generalized into something common among all interesting targets. This is particularly interesting when using dictionary matching to extract target parameters. By generalizing all of the targets, a single dictionary for an electromagnetic dipole can be used, instead of needing to

enumerate every possible frequency response of each individual target type. A huge advancement in the representation of these types of targets is the transformation from the frequency domain into the discrete spectrum of relaxation frequencies (DSRF). Physically, the relaxation frequency is just the inverse of the relaxation time which is the time that it takes for an object induced with an electromagnetic field to return to steady state.

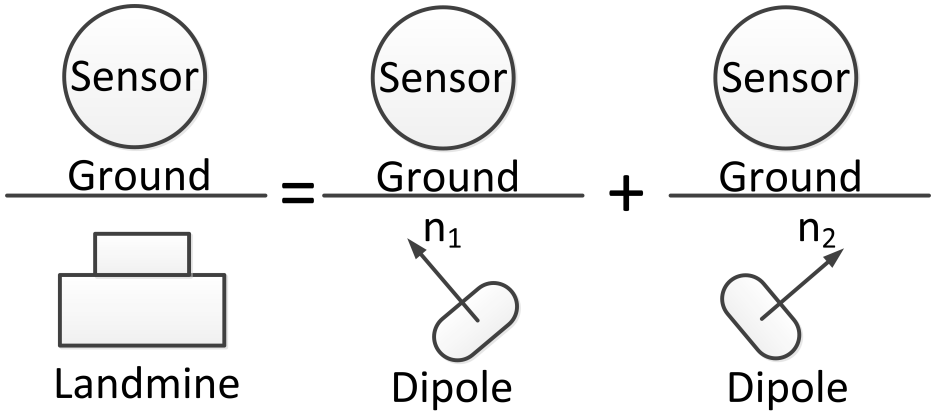


Figure 29: Landmine decomposed into electromagnetic dipoles.

The model, (39) and (40), can be reorganized into a sum of dipole form. The target classification can be done separately once the ζ_k are estimated [21]. After doing this reorganization,

$$\begin{aligned} r_c(\omega, \boldsymbol{l}_s; \boldsymbol{l}_t, \boldsymbol{o}_t) &= \sum_{k=0}^{N_\zeta} CD_k p(\omega, \zeta_k) \boldsymbol{g}_c^H(\boldsymbol{l}_s - \boldsymbol{l}_t) \boldsymbol{R}^H(\boldsymbol{o}_t) \boldsymbol{\Lambda}_k \boldsymbol{R}(\boldsymbol{o}_t) \boldsymbol{f}(\boldsymbol{l}_s - \boldsymbol{l}_t) \\ &= \sum_{k=0}^{N_\zeta} CD_k p(\omega, \zeta_k) a_c(\boldsymbol{l}_s; \boldsymbol{l}_t, \boldsymbol{o}_t, \boldsymbol{\Lambda}_k) \\ &= \sum_{k=0}^{N_\zeta} \upsilon_c^k(\boldsymbol{l}_s; \boldsymbol{l}_t, \boldsymbol{o}_t, \boldsymbol{\Lambda}_k) p_k(\omega, \zeta_k), \end{aligned} \tag{42}$$

the ω component is separated from the location and orientation components, and $p(\omega, \zeta_0) = 1$. $\boldsymbol{\Lambda}_k$ is a 3×3 real, diagonal, positive-semidefinite matrix. The values of υ_c^k become the coefficients of the dipole expansion and it is important to notice that υ_c^k

is not a function of ζ or ω. Since υ_c^k is not a function of frequency, this representation can be used to image each relaxation frequency in the sum, independently, and the dictionary does not need to be enumerated for each relaxation or combination of relaxation frequencies.

The estimation of the DSRF is not part of this particular research, but an explanation of how it is calculated is important because it is essential, and nontrivial. The N_ζ, $(\upsilon_c^k(\boldsymbol{l}_s;\boldsymbol{l}_t,\boldsymbol{o}_t,\boldsymbol{\Lambda}_k),\zeta_k)$ pairs are computed using a method proposed by Wei et. al. [21, 27]. The application that Wei et. al. addressed is determining the support of the relaxation frequencies because it can help to classify a target. The classification would most likely be combined and used with this research to locate, find the orientation of, and classify a target. However, for this thesis, the concern is with locating and finding the orientation of the target, not classifying. First, (42) can be rewritten as a matrix vector multiplication, for simplicity the non essential arguments to the functions are going to be suppressed from the notation, this includes everything except for, ω and ζ,

$$
\begin{aligned}
r_c(\omega) &= \sum_{k=0}^{N_\zeta} \upsilon_c^k p_k(\omega,\zeta_k) \\
\begin{bmatrix} r_c(\omega_1) \\ r_c(\omega_2) \\ \vdots \\ r_c(\omega_{N_\omega}) \end{bmatrix} &= \begin{bmatrix} 1 & p(\omega_1,\zeta_1) & p(\omega_1,\zeta_2) & \cdots & p(\omega_1,\zeta_{N_\zeta}) \\ 1 & p(\omega_2,\zeta_1) & p(\omega_2,\zeta_2) & \cdots & p(\omega_2,\zeta_{N_\zeta}) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & p(\omega_{N_\omega},\zeta_1) & p(\omega_{N_\omega},\zeta_2) & \cdots & p(\omega_{N_\omega},\zeta_{N_\zeta}) \end{bmatrix} \begin{bmatrix} \upsilon_c^0 \\ \upsilon_c^1 \\ \vdots \\ \upsilon_c^{N_\zeta} \end{bmatrix} \\
\boldsymbol{r}_c &= \boldsymbol{P}\boldsymbol{\upsilon}_c.
\end{aligned}
\tag{43}
$$

Since the model order of ζ is not generally known, the ζ space can be discretized to cover the range of possible ζ values, and an over complete dictionary can be used to find the support of ζ_k and the coefficients υ_c^k. For a value $M \gg N_\zeta$, a new linear

problem is set up,

$$\begin{bmatrix} r_c(\omega_1) \\ r_c(\omega_2) \\ \vdots \\ r_c(\omega_{N_\omega}) \end{bmatrix} = \begin{bmatrix} 1 & p(\omega_1,\zeta_1) & p(\omega_1,\zeta_2) & \cdots & p(\omega_1,\zeta_M) \\ 1 & p(\omega_2,\zeta_1) & p(\omega_2,\zeta_2) & \cdots & p(\omega_2,\zeta_M) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & p(\omega_{N_\omega},\zeta_1) & p(\omega_{N_\omega},\zeta_2) & \cdots & p(\omega_{N_\omega},\zeta_M) \end{bmatrix} \begin{bmatrix} \tilde{\upsilon}_c^0 \\ \tilde{\upsilon}_c^1 \\ \vdots \\ \tilde{\upsilon}_c^M \end{bmatrix} \tag{44}$$

$$\boldsymbol{r}_c = \tilde{\boldsymbol{P}}\tilde{\boldsymbol{\upsilon}}_c.$$

In the new setup shown in (44), $\boldsymbol{r}_c$ and $\tilde{\boldsymbol{P}}$ are known, and $\tilde{\boldsymbol{\upsilon}}_c$ needs to be solved for. The support of the N_ζ non-zero values of $\tilde{\boldsymbol{\upsilon}}_c$ correspond to the column space of $\tilde{\boldsymbol{P}}$ that is equivalent to $\boldsymbol{P}$ and it gives the support of ζ. The amplitudes of the non-zero support of $\tilde{\boldsymbol{\upsilon}}_c$ give the values for $\boldsymbol{\upsilon}$. The inversion process that attempts to extract these parameters is done by imposing a non-negative constraint on $\tilde{\boldsymbol{\upsilon}}_c$. The optimization for the inverse becomes,

$$\arg\min_{\tilde{\boldsymbol{\upsilon}}_c} \quad \left\| \begin{bmatrix} \Re\{\boldsymbol{r}_c\} \\ \Im\{\boldsymbol{r}_c\} \end{bmatrix} - \begin{bmatrix} \Re\{\tilde{\boldsymbol{P}}\} \\ \Im\{\tilde{\boldsymbol{P}}\} \end{bmatrix} \tilde{\boldsymbol{\upsilon}}_c \right\|$$

$$\text{s.t.} \quad \tilde{\boldsymbol{\upsilon}}_c \geq 0.$$

The final step in the inversion process is to interpolate. As with any dictionary matching technique where a continuous space is discretized, there is going to be some off-grid error caused by the fact that the actual ζ is most likely not identically on one of the grid points. The off-grid error generally presents itself by splitting the amplitude of the actual relaxation frequency between its two nearest neighbors. The grid for ζ is done on a log scale, which leads to an interpolation for the actual values as,

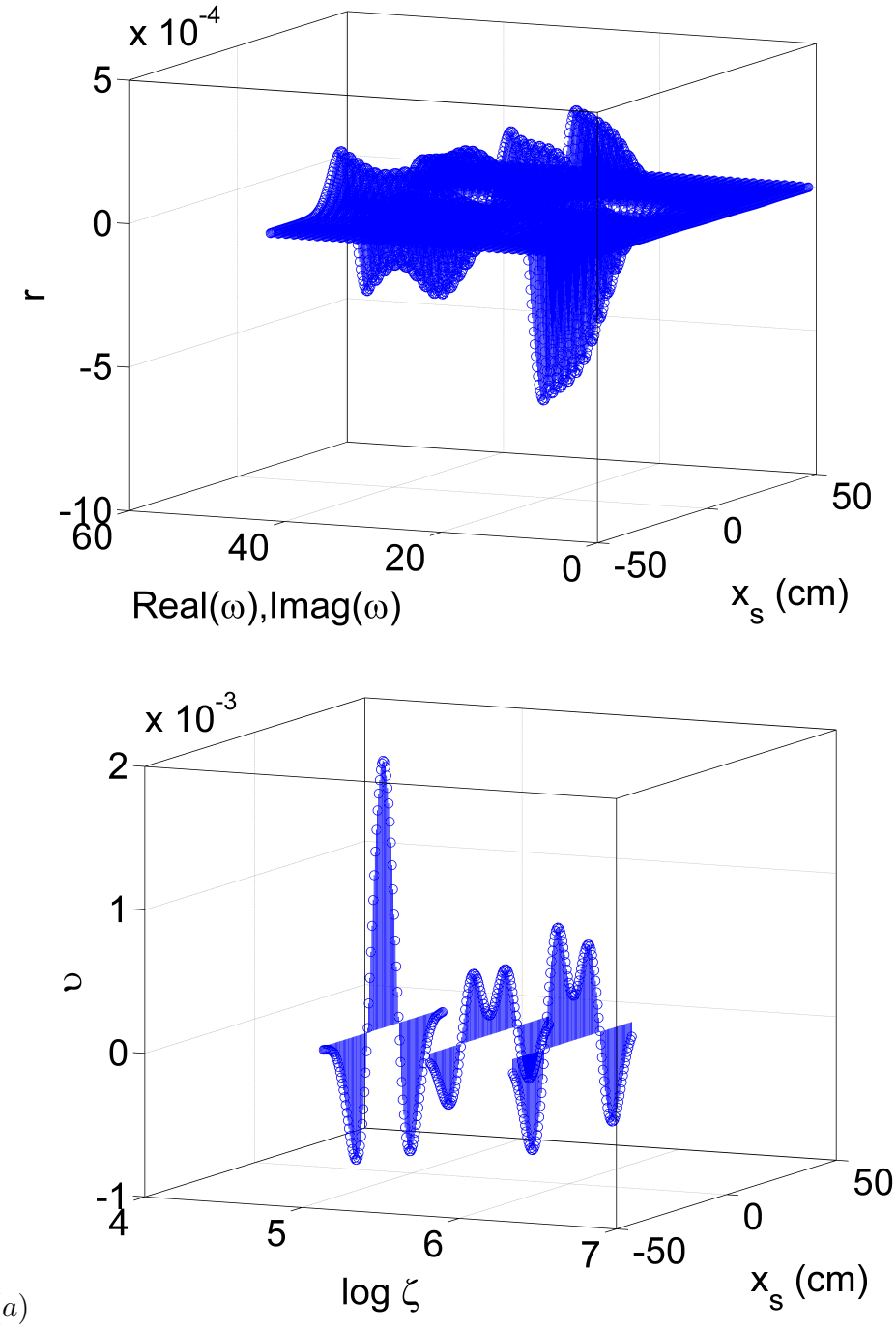
$$\tilde{\upsilon}_c^{intp} = \tilde{\upsilon}_c^a + \tilde{\upsilon}_c^b \tag{45}$$

$$\log(\zeta_{intp}) = \log(\zeta_a) + \frac{\tilde{\upsilon}_c^b}{\tilde{\upsilon}_c^a + \tilde{\upsilon}_c^b} \log\left(\frac{\zeta_b}{\zeta_a}\right). \tag{46}$$

Now that the measurements can be transformed into the DSRF, the coefficients $\upsilon_c^k(\boldsymbol{l}_s;\boldsymbol{l}_t,\boldsymbol{o}_t,\boldsymbol{\Lambda}_k)$ for each ζ_k, can be imaged to obtain the location and orientation parameters of the target. The response model to image the coefficients of each relaxation frequency is now,

$$\upsilon_c^k(\boldsymbol{l}_s;\boldsymbol{l}_t,\boldsymbol{o}_t,\boldsymbol{\Lambda}_k) = Da_c(\boldsymbol{l}_s;\boldsymbol{l}_t,\boldsymbol{o}_t,\boldsymbol{\Lambda}_k). \tag{47}$$

The DSRF coefficients are no longer a function of ω or target type, and if each υ_c^k is imaged independently, then the storage order is reduced to N^7. The representation used in (47) has been inverted using an explicit dictionary [60]. The sample reduction by going from frequency domain to DSRF of a three relaxation target can be seen in Figure 30.



(*a*)

(*b*)

Figure 30: Measurements of a three relaxation target in (a) frequency domain and (b) DSRF.

3.1.2 Tensor amplitude

The proposed changes to modeling and inverting $\upsilon_c^k(\boldsymbol{l}_s;\boldsymbol{l}_t,\boldsymbol{o}_t,\boldsymbol{\Lambda}_k)$ are done with respect to the rotation matrices and $\boldsymbol{\Lambda}_k$. The rotation matrices and $\boldsymbol{\Lambda}_k$ can be rewritten as a tensor

$$\boldsymbol{T}(\boldsymbol{o}_t,\boldsymbol{\Lambda}_k)=\boldsymbol{R}^H(\boldsymbol{o}_t)\boldsymbol{\Lambda}_k\boldsymbol{R}(\boldsymbol{o}_t)=\begin{bmatrix} t_1 & t_4 & t_6 \\ t_4 & t_2 & t_5 \\ t_6 & t_5 & t_3 \end{bmatrix}. \tag{48}$$

The substitution in (48) makes

$$a_c(\boldsymbol{l}_s;\boldsymbol{l}_t,\boldsymbol{o}_t,\boldsymbol{\Lambda}_k)=\boldsymbol{g}_c^H(\boldsymbol{l}_s-\boldsymbol{l}_t)\boldsymbol{T}(\boldsymbol{o}_t,\boldsymbol{\Lambda}_k)\boldsymbol{f}(\boldsymbol{l}_s-\boldsymbol{l}_t). \tag{49}$$

The significance of this substitution is that if $\boldsymbol{T}(\boldsymbol{o}_t,\boldsymbol{\Lambda}_k)$ can be solved for directly, there is no need to enumerate the orientation parameters, saving three dimensions of discretization. $\boldsymbol{T}(\boldsymbol{o}_t,\boldsymbol{\Lambda}_k)$ is also real and positive semidefinite (PSD) which is important when designing the inversion. Linear algebra can be done to reorganize (49) into a vector-matrix multiplication. Since $\boldsymbol{f}=[f_x,f_y,f_z]^H$ and $\boldsymbol{g}=[g_x,g_y,g_z]^H$ are 3D vectors, and since $\boldsymbol{T}(\boldsymbol{o}_t,\boldsymbol{\Lambda}_k)$ is symmetric, a_c can be rewritten as

$$\begin{aligned} a_c(\boldsymbol{l}_s;\boldsymbol{l}_t,\boldsymbol{o}_t,\boldsymbol{\Lambda}_k) &= [g_xf_x,g_yf_y,g_zf_z,g_xf_y+g_yf_x,g_yf_z+g_zf_y,g_xf_z+g_zf_x]\begin{bmatrix} t_1 \\ t_2 \\ t_3 \\ t_4 \\ t_5 \\ t_6 \end{bmatrix} \\ &= \boldsymbol{\psi}_c^H(\boldsymbol{l}_s;\boldsymbol{l}_t)\boldsymbol{t}. \end{aligned}$$

The next step is to vectorize the measurement variable, $\boldsymbol{l}_s$. This is done simply by stacking the $\boldsymbol{\psi}_c^H$ for every $\boldsymbol{l}_s$,

$$\boldsymbol{\Psi}_c(\boldsymbol{l}_t) = \begin{bmatrix} \boldsymbol{\psi}_c^H(\boldsymbol{l}_s{}^1; \boldsymbol{l}_t) \\ \boldsymbol{\psi}_c^H(\boldsymbol{l}_s{}^2; \boldsymbol{l}_t) \\ \vdots \\ \boldsymbol{\psi}_c^H(\boldsymbol{l}_s{}^{N_{l_s}}; \boldsymbol{l}_t) \end{bmatrix}. \tag{50}$$

The steps taken to create (50) must be done for all three different receive coils,

$$\boldsymbol{\Psi}(\boldsymbol{l}_t) = \begin{bmatrix} \boldsymbol{\Psi}_1(\boldsymbol{l}_t) \\ \boldsymbol{\Psi}_2(\boldsymbol{l}_t) \\ \boldsymbol{\Psi}_3(\boldsymbol{l}_t) \end{bmatrix}, \tag{51}$$

for each target location parameter. Therefor, $\boldsymbol{\Psi}(\boldsymbol{l}_t)$ is size $3N_{\boldsymbol{l}_s} \times 6$, and $N_{\boldsymbol{l}_t}$ of them must be stored for a total storage burden of $18N_{\boldsymbol{l}_s}N_{\boldsymbol{l}_t}$. This is much smaller than $3N_{\boldsymbol{l}_s}N_{\boldsymbol{l}_t}N_{\boldsymbol{o}_t}N_\Lambda$, which is the storage requirements if the tensor representation is not used and full enumeration of the orientation parameter is needed.

Similar steps need to be taken with the measurements to create a measurement vector that matches the rows of $\boldsymbol{\Psi}(\boldsymbol{l}_t)$. This is simple to do by vectorizing all of the measurements for the three coils. If $\boldsymbol{t}$ is now a length $6N_{\boldsymbol{l}_t}$ vector, the corresponding response model equation becomes

$$\boldsymbol{\upsilon}^k = \boldsymbol{\Psi}\boldsymbol{t}^k, \tag{52}$$

for the k^{th} relaxation frequency, where

$$\boldsymbol{\Psi} = \left[\begin{array}{c|c|c|c} \boldsymbol{\Psi}(1) & \boldsymbol{\Psi}(2) & \cdots & \boldsymbol{\Psi}(N_{\boldsymbol{l}_t}) \end{array} \right]. \tag{53}$$

A key takeaway from (52) is that $\boldsymbol{\Psi}$ does not depend on the specific relaxation frequency indexed by k, which allows for each measured relaxation frequency to be measured using the same dictionary. The dictionary does not need to be enumerated

for multiple relaxation frequencies. However, each relaxation frequency must be imaged independently and the information combined somehow to help select the target parameters.

3.2 Data-acquisition Deficiencies

There are a few issues with this particular dictionary that make it rather difficult to invert. The creation of the dictionary highlights some of the deficiencies with the data acquisition and suggests some improvements that need to be made to more accurately image the targets. Also, the target response for a target at different positions changes smoothly, so the response from similar targets within a close proximity will be highly correlated, which can cause issues in the detection process. However, when using the DSRF representation of a target, the times when there are two closely located targets is limited, because it is rare for the targets to share relaxation frequencies, and be similarly oriented.

3.2.1 Measurement offsets

The EMI measurement system measures an offset that must be removed before processing the data. The offset is due to fixed hardware offset issues, drift, and from the response of the soil. The offset can be mostly removed by convolving the data with a zero-mean filter with respect to $\boldsymbol{l}_s$. The filter can be constructed using an SVD process. First, the response at a receive coil must be enumerated for all possible $\boldsymbol{l}_t$,

$$\boldsymbol{\Psi}_c = \left[\; \boldsymbol{\Psi}_c(1) \; \middle| \; \boldsymbol{\Psi}_c(2) \; \middle| \cdots \middle| \; \boldsymbol{\Psi}_c(N_{\boldsymbol{l}_t}). \; \right]. \tag{54}$$

Then the singular value decomposition (SVD) of $\boldsymbol{\Psi}_c$ is taken,

$$\boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^H = SVD(\boldsymbol{\Psi}_c). \tag{55}$$

The first column of U, $\boldsymbol{u}_1$, corresponding to the largest singular value of $\boldsymbol{\Psi}_c$, is used as the filter coefficients. Before $\boldsymbol{u}_1$ can be used as a finite impulse response (FIR)

filter, its mean must be removed, so that when it is used as a downrange filter during data collection, the DC offset is eliminated. An example of the $\boldsymbol{u}_1$ that is used with this application can be seen in Figure 31.

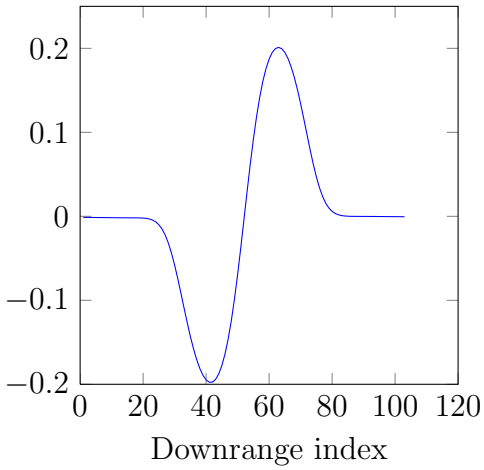


Figure 31: Downrange filter $\boldsymbol{u}_1$ used to eliminate the DC offset in collected EMI measurements.

The linear equation from (52) can be rewritten with the filter applied as a convolution,

$$\boldsymbol{u}_1 * \boldsymbol{v}^k = \boldsymbol{\Psi}_{u_1} \boldsymbol{t}^k, \tag{56}$$

for the k^{th} relaxation frequency, where

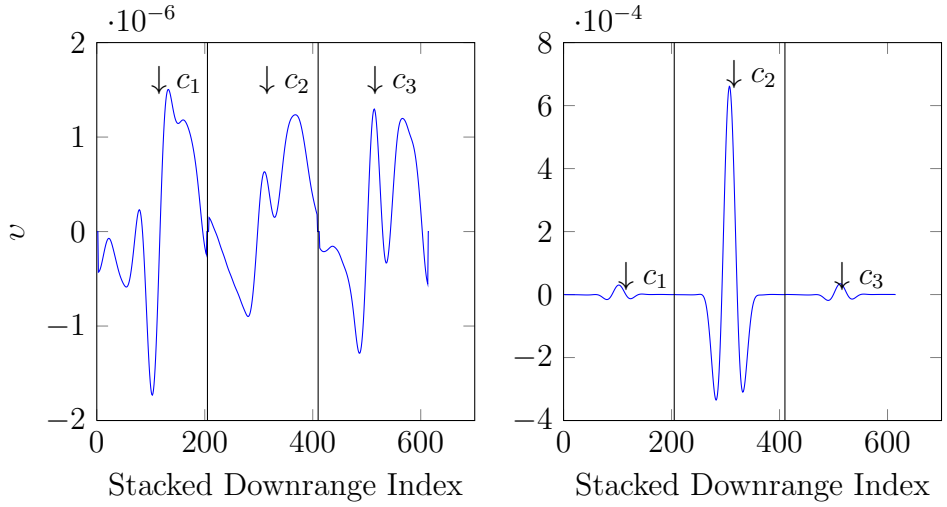
$$\boldsymbol{\Psi}_{u_1} = \left[\boldsymbol{u}_1 * \boldsymbol{\Psi}(1) \,\middle|\, \boldsymbol{u}_1 * \boldsymbol{\Psi}(2) \,\middle|\, \cdots \,\middle|\, \boldsymbol{u}_1 * \boldsymbol{\Psi}(N_{\boldsymbol{l}_t}) \right]. \tag{57}$$

In other words, the downrange filter, $\boldsymbol{u}_1$, is being applied to each enumerated column of the dictionary as the final step in dictionary creation.

3.2.2 Problems with co-linear receivers

The acquisition system does a very poor job of sensing targets that have tensor components perpendicular to the magnetic field. This can easily be explained by an example. The simplest EMI target is a loop of wire which has a tensor that can be

written as the outer product of the unit vector normal to the plane of loop: $\mathbf{\Lambda} = \hat{\boldsymbol{n}}\hat{\boldsymbol{n}}^H$. When this unit vector is perpendicular to the magnetic field for the transmitter and/or receiver, the sensor will be essentially "blind" to the loop. In the case of the current acquisition system, this corresponds to targets with electromagnetic dipoles directed along the y-axis. An example of the measured response from a target rotated so that its dipole equivalent is y and z directed can be seen in Figure 32. The plots are the stacked downrange responses for each of the three receive coils. The stacked responses can be seen clearly in Figure 32 (b) where the response from the first receive coil comprises the indices from 1 to 201, the second coil comprises the indices from 202 to 402, and the third coil comprises the indices from 403 to 603.



(a) (b)

Figure 32: Stacked receive coil measurements for (a) an $\alpha_t = 90°$ and $\beta_t = 90°$ oriented target, y directed, and (b) an $\alpha_t = 0°$ and $\beta_t = 0°$ oriented target, z directed. The response in (a) is two orders of magnitude smaller and is essentially receiver noise.

The y directed target in Figure 32(a) is essentially small measurement noise without any real structure, and Figure 32(b) has very well defined structure and the target can be seen in all three coils. A solution to this problem would be to augment the

system with another receiver that would have a coil directed perpendicular to the current coils so that the y direction is not almost invisible. A second possible solution would be to take a second pass over the target, changing the orientation of the receiver coils after each pass to achieve a similar result to adding additional coils but without having to make hardware modifications.

3.2.3 Problems with coherence

The coherence of the dictionary with respect to closely spaced targets is a serious problem. When high spatial resolution is required, a high SNR is required to distinguish one location from the other. The high correlation factor can be seen directly from the dictionary elements. The concern is distinguishing between two simulated responses of the same target at the same orientation, and different locations. A small toy problem was created to evaluate this issue and a dictionary was created using (56) with a fixed tensor value

$$\begin{aligned}\boldsymbol{T}([0, 22.5^\circ, 0], \operatorname{diag}(0.5, 0, 1)) &= \boldsymbol{R}^H([0, 22.5^\circ, 0]) \operatorname{diag}(0.5, 0, 1) \boldsymbol{R}([0, 22.5^\circ, 0]) \\ &= \begin{bmatrix} 0.38 & 0.92 \\ 0 & 0 \\ 0.92 & -0.38 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 0.5 \end{bmatrix} \begin{bmatrix} 0.38 & 0.92 \\ 0 & 0 \\ 0.92 & -0.38 \end{bmatrix}^H \\ &= \begin{bmatrix} 0.57 & 0 & 0.18 \\ 0 & 0 & 0 \\ 0.18 & 0 & 0.93 \end{bmatrix},\end{aligned}$$

and with simulated target locations between $\boldsymbol{l}_t = (0, -6, 4.5)$ and $\boldsymbol{l}_t = (0, 6, 11.5)$ using a 2 cm discretization in y_t and 1 cm discretization in z_t. A coherence matrix was created to check the euclidean distance between all of the dictionary columns. The minimum distance to each simulated location is shown in Table 4 where the values in the matrix are calculated by

$$\left\| \boldsymbol{\Psi}(\boldsymbol{l}_t{}^1) - \boldsymbol{\Psi}(\boldsymbol{l}_t{}^2) \right\|_2. \tag{58}$$

Table 5 shows the same data, but in percentage instead of absolute form,

$$\frac{\left\|\boldsymbol{\Psi}(\boldsymbol{l}_t{}^1) - \boldsymbol{\Psi}(\boldsymbol{l}_t{}^2)\right\|_2}{\left\|\boldsymbol{\Psi}(\boldsymbol{l}_t{}^1)\right\|_2}. \tag{59}$$

No matter how the dictionary is viewed, there are simulated target locations that look very similar to others, and if even a small amount of noise is added, differentiating between certain dictionary elements is almost impossible. This can be further visualized in Figure 33. Each plot shows a simulated target response at a fixed y_t position and changing z_t. Locating the targets is generally accomplished by analyzing the relative strength of the measured signal at each of the three coils. This is obvious when the target location shifts in y, as the ratio of the amplitudes change dramatically. However, when changing in depth, z_t, if there is enough noise present, then a scaled version of the same response at a different depth could be easily misidentified. This also becomes a problem when using inversion techniques like OMP or CS that require a user-specified error parameter. If this parameter is not set exactly right, the estimated locations can change quite dramatically.

Table 4: Absolute minimum distance between one dictionary location and the rest of the dictionary locations at a specific orientation. (10^{-3})

$\boldsymbol{z_t \backslash y_t}$	$\boldsymbol{-6\,cm}$	$\boldsymbol{-4\,cm}$	$\boldsymbol{-2\,cm}$	$\boldsymbol{0\,cm}$	$\boldsymbol{2\,cm}$	$\boldsymbol{4\,cm}$	$\boldsymbol{6\,cm}$
$\boldsymbol{4.5\,cm}$	1.01	0.50	0.15	0.15	0.15	0.50	1.01
$\boldsymbol{5.5\,cm}$	0.69	0.38	0.15	0.15	0.15	0.38	0.69
$\boldsymbol{6.5\,cm}$	0.47	0.28	0.13	0.13	0.13	0.28	0.47
$\boldsymbol{7.5\,cm}$	0.32	0.20	0.11	0.11	0.11	0.20	0.32
$\boldsymbol{8.5\,cm}$	0.22	0.15	0.09	0.09	0.09	0.15	0.22
$\boldsymbol{9.5\,cm}$	0.15	0.11	0.07	0.07	0.07	0.11	0.15
$\boldsymbol{10.5\,cm}$	0.11	0.08	0.05	0.05	0.05	0.08	0.11
$\boldsymbol{11.5\,cm}$	0.08	0.06	0.04	0.04	0.04	0.06	0.08

Table 5: Percentage of minimum distance between one dictionary location and the rest of the dictionary locations at a specific orientation.

$z_t \backslash y_t$	$-6\,cm$	$-4\,cm$	$-2\,cm$	$0\,cm$	$2\,cm$	$4\,cm$	$6\,cm$
$4.5\,cm$	30.92%	14.54%	3.98%	3.89%	4.00%	14.57%	30.91%
$5.5\,cm$	30.41%	16.14%	5.80%	5.68%	5.82%	16.18%	30.41%
$6.5\,cm$	29.53%	16.96%	7.42%	7.28%	7.43%	16.99%	29.52%
$7.5\,cm$	27.87%	17.24%	8.72%	8.57%	8.73%	17.27%	27.91%
$8.5\,cm$	26.08%	17.16%	9.71%	9.56%	9.72%	17.19%	26.11%
$9.5\,cm$	24.35%	16.85%	10.44%	10.30%	10.45%	16.88%	24.38%
$10.5\,cm$	22.71%	16.41%	10.94%	10.82%	10.95%	16.43%	22.74%
$11.5\,cm$	21.19%	15.89%	11.27%	11.17%	11.28%	15.91%	21.21%

3.3 Inversion Algorithm

The inversion algorithm is the next important step after model creation. There are two different ways that this type of problem can be inverted. The first is using a large, block-structured tensor problem to simultaneously extract the location and the tensor in a single optimization problem. An example of creating a large, block-structured convex optimization problem using semidefinite programming (SDP) can be found in [48]. The second inversion method is to split the problem into two parts, a location-estimation stage, followed by a tensor-estimation stage. The separability of this problem allows for a more time and storage efficient algorithm to be used to extract a rough estimate for the target location, and then the second stage refines the location estimate, and extracts the tensor.

3.3.1 Large-block SDP

The ideal optimization problem is one where all the parameters can be extracted in a single step. The large-block SDP takes this approach. This will allow for the

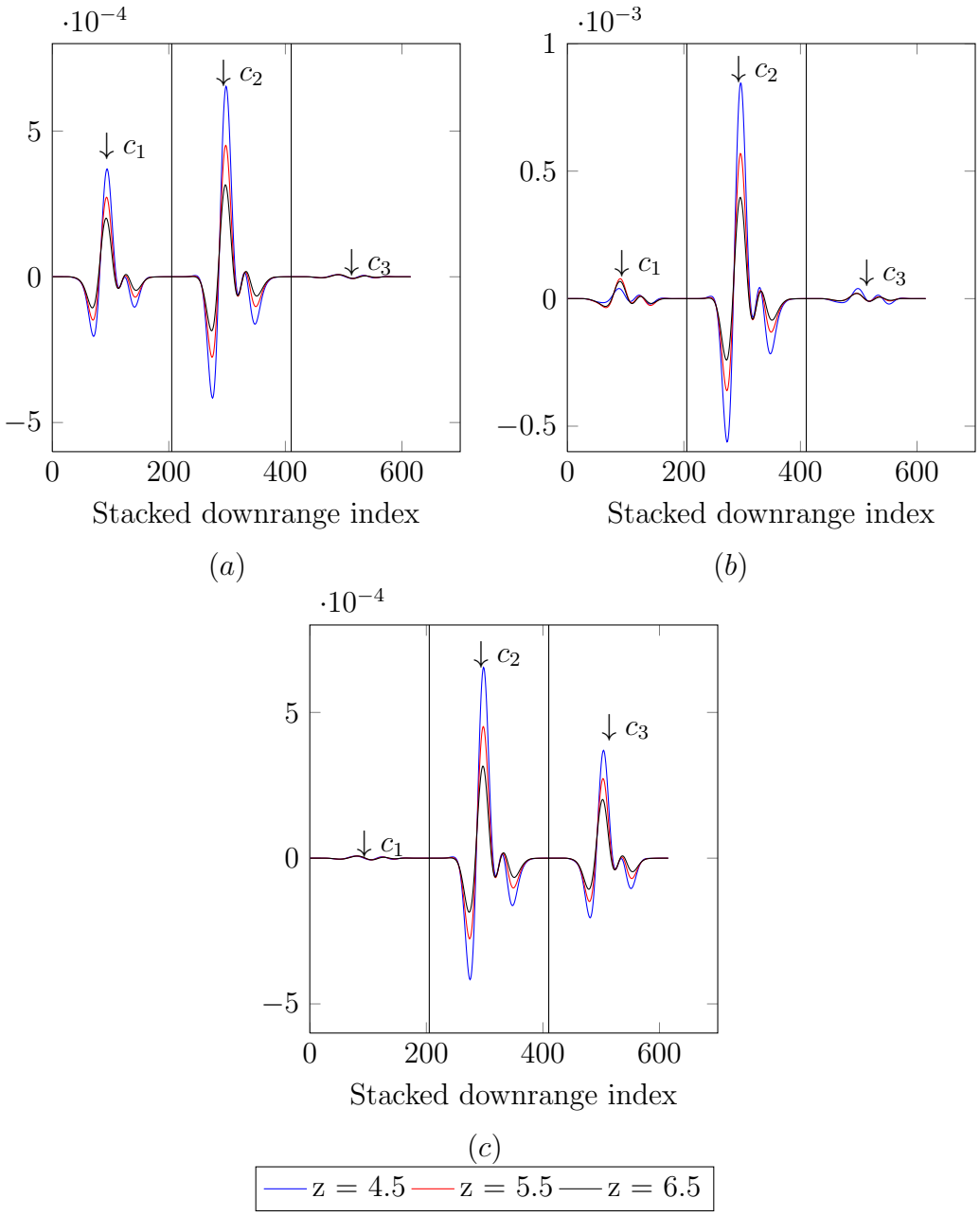


Figure 33: Target responses for (a) $y_t = -6\,\text{cm}$ (b) $y_t = 0\,\text{cm}$ (c) $y_t = 6\,\text{cm}$.

convex optimization problem to use the inherent sparsity in the number of targets to its advantage in finding the location. The vector $\mathbf{t}$ now becomes a length $6N_{\boldsymbol{l}_t}$ sparse vector, and

$$\boldsymbol{T} = \begin{bmatrix} \boldsymbol{T}_1(\boldsymbol{o}_t, \boldsymbol{\Lambda}_k) & 0 & 0 & 0 \\ 0 & \boldsymbol{T}_2(\boldsymbol{o}_t, \boldsymbol{\Lambda}_k) & 0 & 0 \\ 0 & 0 & \ddots & \vdots \\ 0 & 0 & \cdots & \boldsymbol{T}_{N_{\boldsymbol{l}_t}}(\boldsymbol{o}_t, \boldsymbol{\Lambda}_k) \end{bmatrix}. \tag{60}$$

All of the previous properties still apply, $\boldsymbol{T}$ is symmetric, $\boldsymbol{T}\succeq 0$, and can be constructed directly from the solution vector $\mathbf{t}$. The PSD property, $\boldsymbol{T}\succeq 0$, follows directly from the fact that for each sub matrix $\boldsymbol{T}_{\boldsymbol{l}_t}(\boldsymbol{o}_t, \boldsymbol{\Lambda}_k)\succeq 0$. It is obvious in the block-diagonal structure of $\boldsymbol{T}$, that the eigenvalues of $\boldsymbol{T}$ are the enumeration of all the eigenvalues from the individual $\boldsymbol{T}_{\boldsymbol{l}_t}$. It follows directly that if there exists a $\boldsymbol{T}_{\boldsymbol{l}_t}$ that has a negative eigenvalue, then $\boldsymbol{T}$ must also have the same negative eigenvalue. Also, if $\boldsymbol{T}$ contains at least one negative eigenvalue, then there must exist at least one $\boldsymbol{T}_{\boldsymbol{l}_t}$ that has the corresponding negative eigenvalue. This proves that $\boldsymbol{T}\succeq 0$ if and only if all $\boldsymbol{T}_{\boldsymbol{l}_t}\succeq 0$. To start the imaging process, the response model should be slightly updated to account for unknown noise. The response model in (56) will be slightly changed by including an additive noise vector, $\boldsymbol{\eta}$, to form the system measurements

$$\boldsymbol{m}^k = \boldsymbol{\Psi}\boldsymbol{t}^k + \boldsymbol{\eta}, \tag{61}$$

where $\boldsymbol{m}^k$ is now the measurement vector received at the k^{th} relaxation frequency, and $\boldsymbol{t}^k$ contains the tensor elements that need to be solved for. The subscript corresponding to the zero-mean filter is suppressed for now on to keep the notation cleaner. The optimization problem for the block-structured tensor is

$$\begin{aligned} \min \quad & \mathbf{tr}(\hat{\boldsymbol{T}}^k) \\ \text{s.t.} \quad & \hat{\boldsymbol{T}}^k \succeq 0 \\ & \| \boldsymbol{m}^k - \boldsymbol{\Psi}\hat{\boldsymbol{t}}^k \|_2 < \epsilon. \end{aligned} \tag{62}$$

Using the trace of $\hat{\boldsymbol{T}}^k$ is a relaxation of the rank minimization problem, which is desirable because we want a solution to $\hat{\boldsymbol{T}}^k$ that is going to be very low rank, generally between one and six [40]. There is some legitimate concern that this problem could become extremely large, since $\hat{\boldsymbol{T}}^k$ is of size $3N_xN_yN_z \times 3N_xN_yN_z$. The storage requirements for $\hat{\boldsymbol{T}}^k$ can be slightly alleviated since it is only a five diagonal, symmetric matrix, and the trace is only concerned with the main diagonal. The known structure of $\hat{\boldsymbol{T}}^k$ allows many storage-efficient techniques to be used to keep complexity down. The low-rank matrix approximation is enforcing a sparsity constraint on the number of targets. The sparsity constraint on the targets is enforced because the number of eigenvalues directly corresponds to the number of electromagnetic dipoles present in the specific relaxation frequency. It can be seen from Figure 29 that each target can be dissected directly into a linear combination of electromagnetic dipoles, and by imaging the dipoles, more complex targets can be extracted.

3.3.2 Two-stage inversion

The two main parameters that need to be extracted are the tensor and the location. To save computation, the location can be found first, and then the tensor would be extracted at that location to produce the final solution [59]. As was done in [60], each individual relaxation frequency of the measurement will be imaged independently.

The first stage of the inversion will be to estimate the location, $\boldsymbol{l}_t^k$, of the relaxation frequency. This is done by solving many small, least-squares problems for each possible target location. The least-squares solution vector, $\hat{\boldsymbol{t}}^k(\boldsymbol{l}_t)$, for each target location becomes

$$\hat{\boldsymbol{t}}^k(\boldsymbol{l}_t) = (\boldsymbol{\Psi}(\boldsymbol{l}_t)^H \boldsymbol{\Psi}(\boldsymbol{l}_t))^{-1} \boldsymbol{\Psi}(\boldsymbol{l}_t)^H \boldsymbol{m}^k. \tag{63}$$

Now that there is an approximation for $\hat{\boldsymbol{t}}^k$ for all $\boldsymbol{l}_t$, the particular $\boldsymbol{l}_t$ that minimizes the residual with $\boldsymbol{m}^k$ must be found,

$$\hat{\boldsymbol{l}}_t^k = \underset{\boldsymbol{l}_t}{\operatorname{argmin}} \|\boldsymbol{\Psi}(\boldsymbol{l}_t)\hat{\boldsymbol{t}}^k(\boldsymbol{l}_t) - \boldsymbol{m}^k\|_2. \tag{64}$$

The iterative least-squares process shown in (63) and (64) can be sped up by computing the least-squares matrices offline, and concatenating them into one large matrix, similar to $\boldsymbol{\Psi}$.

The least-squares solution will provide an estimate for the tensor as well, but the estimate will be poor since least-squares cannot constrain the tensor sufficiently. To get a better estimate, the tensor must be constrained to be positive-semidefinite (PSD). In addition, the sensor may be "blind" to some tensor components making the inversion ill conditioned. Constraining the eigenvalues of $\boldsymbol{T}$ can be used to reduce the influence of the "blind" tensor components. On the other hand, it would be preferable to have a new acquisition system design that would not have these deficiencies. Using a small SDP will constrain the tensor to being PSD, and minimizing the trace, a relaxation on rank minimization, will constrain the eigenvalues [40]. The optimization problem becomes,

$$\begin{aligned} \min \quad & tr(\hat{\boldsymbol{T}}^k_{\hat{\boldsymbol{l}}_t}(\boldsymbol{o}_t, \boldsymbol{\Lambda}_k)) \\ \text{s.t.} \quad & \hat{\boldsymbol{T}}^k_{\hat{\boldsymbol{l}}_t}(\boldsymbol{o}_t, \boldsymbol{\Lambda}_k) \succeq 0 \\ & \parallel \boldsymbol{m}^k - \boldsymbol{\Psi}(\hat{\boldsymbol{l}}_t^{\,k})\hat{\mathbf{t}}^k \parallel_2 < \epsilon, \end{aligned} \tag{65}$$

where $\hat{\mathbf{t}}^k$ is equal to the six independent values of $\hat{\boldsymbol{T}}^k_{\hat{\boldsymbol{l}}_t}(\boldsymbol{o}_t, \boldsymbol{\Lambda}_k)$, and $\hat{\boldsymbol{l}}^k_t$ is from (64). The selection of ϵ can be a difficult task. Since the optimization problem in (65) is relatively small, only six unknowns from the 3×3 symmetric $\hat{\boldsymbol{T}}^k_{\hat{\boldsymbol{l}}_t}(\boldsymbol{o}_t, \boldsymbol{\Lambda}_k)$, and can be computed quickly, ϵ is chosen using an L-curve analysis that was discussed in Chapter 1. An example for an L-curve used for this problem can be seen in Figure 34. The trace of the solution is used as the measure to make sure the chosen ϵ is not over fitting the noise. Once an approximation is made for the tensor, the orientation of the target, and its symmetry can be extracted from $\hat{\boldsymbol{T}}^k_{\hat{\boldsymbol{l}}_t}(\boldsymbol{o}_t, \boldsymbol{\Lambda}_k)$ by doing an eigendecomposition. The eigenvectors correspond to the direction of each axis of magnetic polarizability, and each eigenvalue corresponds to the strength of that axis. For example, a dipole

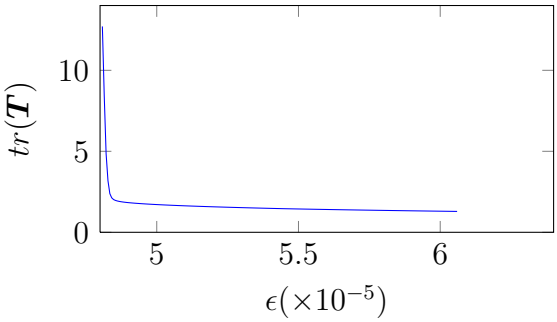


Figure 34: L-curve for ϵ selection.

will have a single eigenvalue, a sphere will have three equal eigenvalues, and a cylinder will have two equal eigenvalues and one non-equal eigenvalue.

The entire process from (63) to (65) is repeated for each individual relaxation frequency.

3.4 Simulations

Two simple experiments were run to test the method. The simulation treats a very small 2D problem, but can be expanded easily. Only one relaxation frequency is present, which will allow for the suppression of the k notation, and only two spatial dimensions are considered, $\boldsymbol{l}_t = (y_t, z_t)$, N_{y_t}=7 at 2cm spacing, and N_{z_t}=8 with 1cm spacing. Also, only two angles $\boldsymbol{o}_t = (\alpha_t, \beta_t)$ are used for orientation. The inversion algorithm used for these simulations is the large-block SDP. A consideration for setting up these problems is how to choose the ϵ value in (62). The SNR for each experiment is set at 35dB. For these experiments, a starting point for ϵ is selected to be the following simple estimate of the noise. It is reasonable to assume that most of the downrange measurements will contain no target, so the standard deviation is taken in this estimated target-less area to get the estimated noise amplitude. Then an L-curve, like the one in Figure 34, is made by varying ϵ below and above the starting ϵ. The value of ϵ is then selected at the knee of the curve.

The first experiment consists of a single target located at $\boldsymbol{l}_t = (0, 6.5)\,\text{cm}$, with

a 2-dimensional EM symmetry $\boldsymbol{\lambda} = (0.5, 0, 1)$, where $\boldsymbol{\Lambda} = \text{diag}(\boldsymbol{\lambda})$, oriented at $\boldsymbol{o}_t = (0°, 22.5°)$. The orientation for the target is based on the principle axis of the electromagnetic field for this experiment, which is the z axis. This target is represented by the tensor

$$\begin{aligned}\boldsymbol{T} &= \begin{bmatrix} 0.57 & 0.00 & 0.17 \\ 0.00 & 0.00 & 0.00 \\ 0.17 & 0.00 & 0.92 \end{bmatrix} \\ &= \begin{bmatrix} 0.92 & 0.38 \\ 0 & 0 \\ -0.38 & 0.92 \end{bmatrix} \begin{bmatrix} 0.5 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0.92 & 0.38 \\ 0 & 0 \\ -0.38 & 0.92 \end{bmatrix}^H\end{aligned}$$

when written as an eigenvalue expansion. After solving (62) the estimated block-tensor corresponded to the correct location, and the estimated tensor is

$$\hat{\boldsymbol{T}} = \begin{bmatrix} 0 & 0 & \cdots & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & \cdots & 0 \\ 0 & 0 & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \hat{\boldsymbol{T}}_{\boldsymbol{l}_t}(\boldsymbol{o}_t, \boldsymbol{\Lambda}) & \cdots & 0 \\ 0 & 0 & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & \cdots & 0 \end{bmatrix}.$$

There is only a single nonzero block in $\hat{T}$, the location of the block corresponds to the correct location $\boldsymbol{l}_t = (0, 6.5)\,\text{cm}$. The location estimation image can be seen in Figure 35, which is created by taking the trace of each block $\hat{\boldsymbol{T}}_{\boldsymbol{l}_t}(\boldsymbol{o}_t, \boldsymbol{\Lambda})$ in the

estimated block-tensor $\hat{\boldsymbol{T}}$. The estimated $\hat{\boldsymbol{T}}_{\boldsymbol{l}_t}(\boldsymbol{o}_t, \boldsymbol{\Lambda})$ has value of

$$\begin{aligned}\hat{\boldsymbol{T}}_{\boldsymbol{l}_t}(\boldsymbol{o}_t, \boldsymbol{\Lambda}) &= \begin{bmatrix} 0.57 & 0.01 & 0.20 \\ 0.01 & 0.00 & 0.00 \\ 0.20 & 0.00 & 0.89 \end{bmatrix} \\ &= \begin{bmatrix} 0.90 & 0.42 \\ 0.03 & 0.00 \\ -0.42 & 0.90 \end{bmatrix} \begin{bmatrix} 0.48 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0.90 & 0.42 \\ 0.03 & 0.00 \\ -0.42 & 0.90 \end{bmatrix}^H .\end{aligned}$$

The eigenvalues of $\boldsymbol{\Lambda}$ are normalized such that the largest eigenvalue is one. The plot of the actual electromagnetic field symmetry and the estimated electromagnetic field symmetry can be seen in Figure 36. The estimation of the tensor is quite accurate because $\lambda(1)$ has less than a 5% error, and $\lambda(2)$ is exact. The orientation of the target is determined by evaluating the direction of the eigenvectors of $\hat{\boldsymbol{T}}_{\boldsymbol{l}_t}(\boldsymbol{o}_t, \boldsymbol{\Lambda})$. The direction of the eigenvectors can be calculated using simple trigonometry. The first step is to calculate the yaw, α, which can be done by evaluating the x and y values of the eigenvectors. For the first eigenvector, $\hat{\boldsymbol{e}}_1$, the x and y coordinates are $(0.90, 0.03)$ and the yaw is calculated to be

$$\hat{\alpha}_{\boldsymbol{e}_1} = \frac{180}{\pi} \arctan(\frac{y}{x}) \tag{66}$$

$$= \frac{180}{\pi} \arctan(\frac{0.03}{0.90}) = 1.91^\circ. \tag{67}$$

For the second eigenvector, $\hat{\boldsymbol{e}}_2$, the x and y coordinates are $(0.42, 0.00)$ and the yaw is calculated to be

$$\hat{\alpha}_{\boldsymbol{e}_2} = \frac{180}{\pi} \arctan(\frac{0.00}{0.42}) = 0^\circ. \tag{68}$$

The next step is to calculate the pitch, β, which can be done by evaluating the z values and the projection onto the xy plane of the eigenvectors. The xy projection value is simply

$$xy = \sqrt{x^2 + y^2}. \tag{69}$$

For the first eigenvector, $\hat{\boldsymbol{e}}_1$, the (z, xy) coordinates are $(-0.42, 0.90)$ and the pitch is calculated to be

$$\hat{\beta}_{\boldsymbol{e}_1} = \frac{180}{\pi} \arctan(\frac{\sqrt{x^2+y^2}}{z}) \tag{70}$$

$$= \frac{180}{\pi} \arctan(\frac{0.90}{-0.42}) = -64.98°. \tag{71}$$

For the second eigenvector, $\hat{\boldsymbol{e}}_2$, the x and y coordinates are $(0.90, 0.42)$ and the pitch is calculated to be

$$\hat{\beta}_{\boldsymbol{e}_2} = \frac{180}{\pi} \arctan(\frac{0.42}{0.90}) = 25.02°. \tag{72}$$

Since $\hat{\boldsymbol{e}}_2$ is the principle vector, the estimated orientation of the target pitch is 25.05°. This is reinforced by evaluating the secondary eigenvector $\hat{\boldsymbol{e}}_1$, which is pitched $-64.98°$ off the z axis, equivalent to 125.02°, or pitched 25.02° off the xz-plane. The total angle error can be calculated based on the principle axis. The angle error between two vectors is

$$\boldsymbol{o_t}^{err} = \frac{180}{\pi} \arccos(\frac{\hat{\boldsymbol{e_1}}^H \boldsymbol{e_2}}{\|\hat{\boldsymbol{e_1}}\| \, \|\boldsymbol{e_2}\|}). \tag{73}$$

For this particular simulation the estimate matches within 2.5°, which is an acceptable margin.

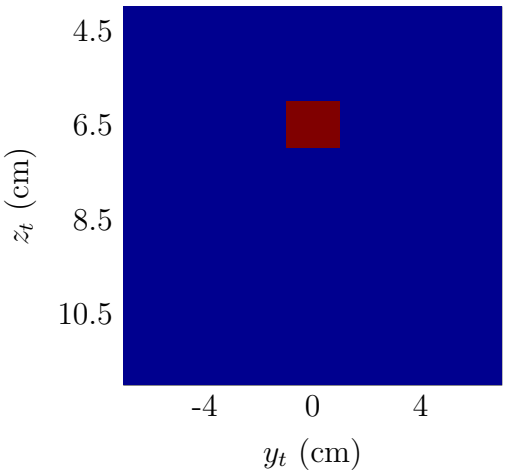


Figure 35: Location estimate for a single relaxation with 2D symmetry and true location of $\boldsymbol{l}_t = (0, 6.5)$ cm.

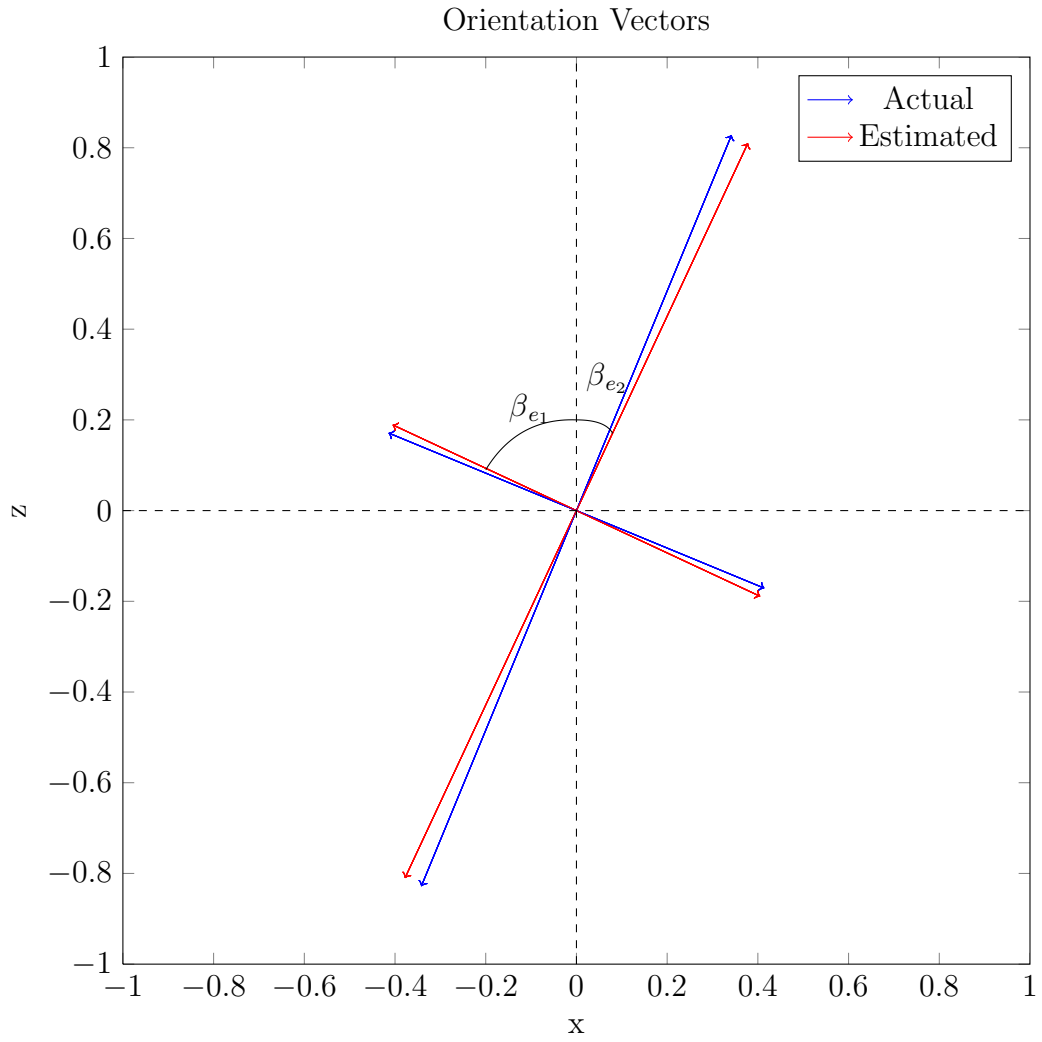


Figure 36: Orientation and symmetry of the electromagnetic fields of the estimated and actual target.

The second experiment consists of two targets located at $\boldsymbol{l_t}^1 = (0, 6.5)\,\text{cm}$ and $\boldsymbol{l_t}^2 = (-6, 7.5)\,\text{cm}$, with EM symmetries $\boldsymbol{\lambda}_1 = (0, 0, 1)$ and $\boldsymbol{\lambda}_2 = (1, 0, 0)$, both are

oriented at $\boldsymbol{o}_t = (45°, 22.5°)$, and represented with tensors

$$\begin{aligned}\boldsymbol{T}_1 &= \begin{bmatrix} 0.07 & -0.07 & 0.25 \\ -0.07 & 0.07 & -0.25 \\ 0.25 & -0.25 & 0.85 \end{bmatrix} \\ &= \begin{bmatrix} 0.27 \\ -0.27 \\ 0.92 \end{bmatrix} \begin{bmatrix} 1 \end{bmatrix} \begin{bmatrix} 0.27 \\ -0.27 \\ 0.92 \end{bmatrix}^H ,\end{aligned}$$

and

$$\begin{aligned}\boldsymbol{T}_2 &= \begin{bmatrix} 0.42 & -0.42 & -0.25 \\ -0.42 & 0.42 & 0.25 \\ -0.25 & 0.25 & 0.14 \end{bmatrix} \\ &= \begin{bmatrix} -0.65 \\ 0.65 \\ 0.38 \end{bmatrix} \begin{bmatrix} 1 \end{bmatrix} \begin{bmatrix} -0.65 \\ 0.65 \\ 0.38 \end{bmatrix}^H ,\end{aligned}$$

The estimated tensor corresponded to the correct locations again, the location estimation image can be seen in Figure 37, and the estimated tensors are

$$\begin{aligned}\hat{\boldsymbol{T}}_1 &= \begin{bmatrix} 0.41 & -0.42 & -0.25 \\ -0.42 & 0.42 & 0.25 \\ -0.25 & 0.25 & 0.15 \end{bmatrix} \\ &= \begin{bmatrix} 0.28 \\ -0.26 \\ 0.92 \end{bmatrix} \begin{bmatrix} 0.98 \end{bmatrix} \begin{bmatrix} 0.28 \\ -0.26 \\ 0.92 \end{bmatrix}^H ,\end{aligned}$$

and

$$\hat{\boldsymbol{T}}_2 = \begin{bmatrix} 0.08 & -0.07 & 0.26 \\ -0.07 & 0.07 & -0.24 \\ 0.26 & -0.24 & 0.85 \end{bmatrix}$$
$$= \begin{bmatrix} -0.64 \\ 0.65 \\ 0.39 \end{bmatrix} \begin{bmatrix} 1 \end{bmatrix} \begin{bmatrix} -0.64 \\ 0.65 \\ 0.39 \end{bmatrix}^H .$$

Again, the estimates of the tensors are quite accurate: $\boldsymbol{\lambda_1}$ has less than a 2% error, and $\boldsymbol{\lambda_1}$ is exact. The orientation estimate was even more accurate than the first simulation with both targets being oriented within 1° of the actual orientation.

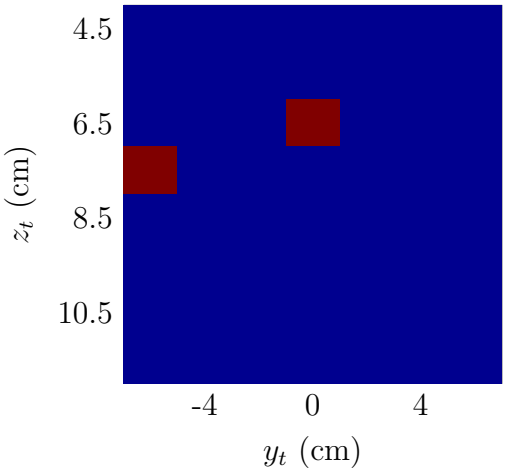
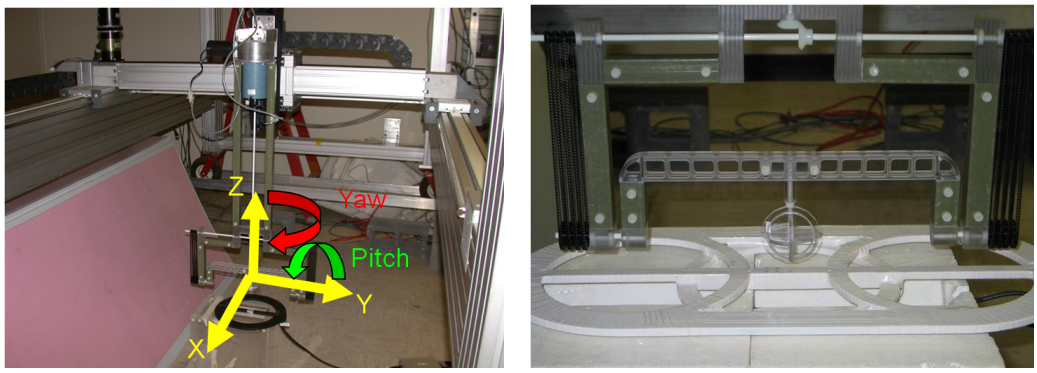


Figure 37: Location image for two targets at a single relaxation frequency.

3.5 Laboratory Experimental Results

A few different laboratory experiments have been run to show the validity of the method proposed in this chapter. The inversion technique that will be used is the two-stage inversion. The two-stage inversion is used because of its efficient computation, and the fact that it is known that only a single target is being imaged at a time. The experiments were performed in a lab, where a target is held above a sensor like the

one in Figure 38(b) at 21 different locations and 15 different orientations [1,61]. The orientation combinations are only varied in the α_t and β_t angles, and the location combinations are only in y_t and z_t, $x_t = 0\,\text{cm}$. The targets are then scanned at $N_{\boldsymbol{l}_s} = 201$ different downrange locations and data is collected at $N_\omega = 21$ wideband frequencies. The frequency response measurements are not used directly, but are transformed into the DSRF in the same steps that were done to create the dictionary in Section 3.1, and each relaxation frequency is imaged separately.



(a) (b)

Figure 38: Experiment setup. (a) Automated translational and rotational axes and (b) EMI sensor used with one transmitter coil and three receiver coils.

The dictionary is built to find targets between $\boldsymbol{l}_t = (-10, -30, 4.5)\,\text{cm}$ and $\boldsymbol{l}_t = (10, 30, 29.5)\,\text{cm}$ with a step size of $(0.5, 2, 1)\,\text{cm}$, making $N_{\boldsymbol{l}_t} = 16926$. This makes the full dictionary size, $3N_{\boldsymbol{l}_s} \times 6N_{\boldsymbol{l}_t} \approx 603 \times (1 \times 10^5)$, which is approximately 250 Mbytes. If the tensor representation is not used, and a 5° sampling is required in the discretization of the orientation, the size would be $3N_{\boldsymbol{l}_s} \times N_{\boldsymbol{o}_t} N_{\boldsymbol{l}_t} N_\Lambda \approx 603 \times (1 \times 10^9)$, or approximately 900 Gbytes.

There are four different targets used. The first is a target created for its simplicity and is just a metal loop, which creates a single-dipole response. The second is three metal loops organized orthogonal to each other to create a response that contains three dipoles all perpendicular to each other in 3D space. The third, target A, is

an actual landmine with two relatively strong dipoles oriented in the same direction. The final target, target B, is a landmine that has a very complicated response with six relaxation frequencies most having several components in the tensor. The response for target B is only known empirically.

3.5.1 Single-loop target

The single-loop target is used as a benchmark because its response is strong, clear, and simple. If the algorithm is not able to image the single-loop target, then the chances of getting accurate inversions with more complex targets are low. To do a baseline test, all 315 different parameter combinations are imaged. A few examples of location extraction for the single-loop target can be seen in Figure 39. With $\boldsymbol{o}_t = (45°, 45°)$ and the target at a relatively shallow depth of 7.5 cm, the locating of the relaxation frequency is perfect. When $\boldsymbol{o}_t = (90°, 90°)$, as is shown in Figure 39(d),(e), and (f), the locating of the relaxation frequency is much less accurate, and when $y_t = 0$ cm, the estimated target location is not close. The difficulty of finding targets in the y-direction is due to the co-linear receivers as discussed in subsection 3.2.2. The off-center targets are not perfect, but they are not nearly as bad as the centered target, meaning that even if only a little, some y directed information can be extracted in certain situations.

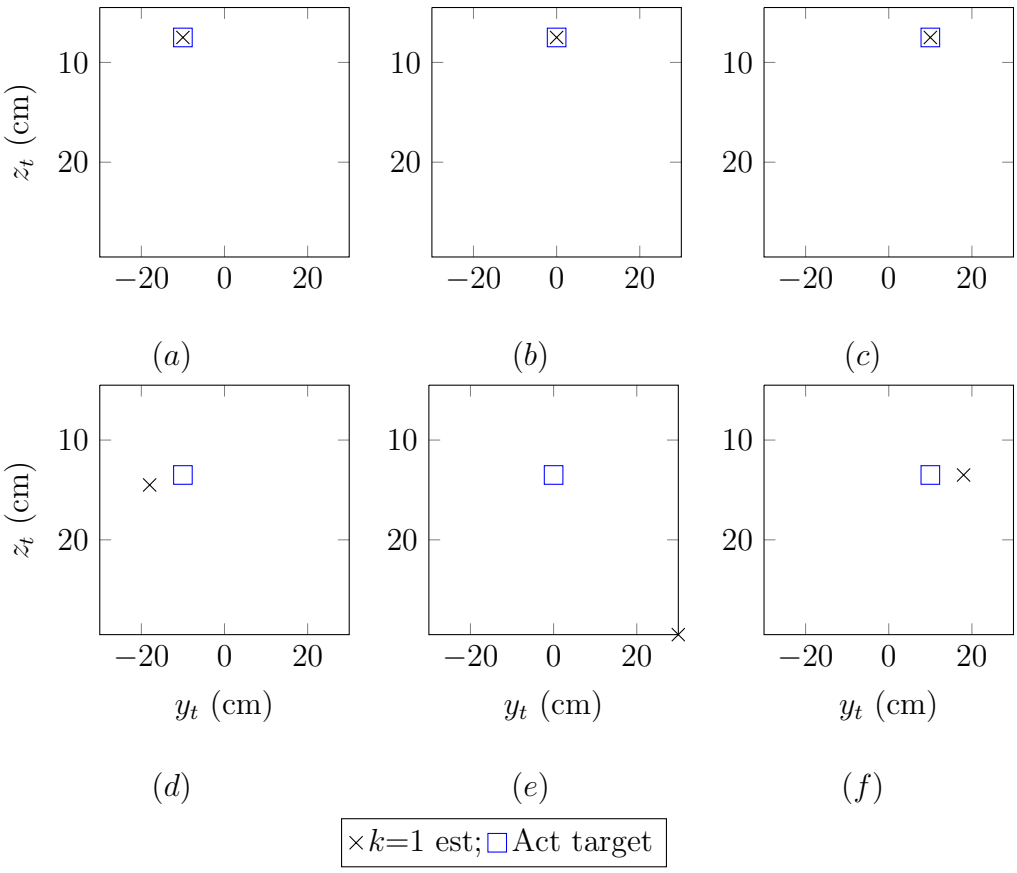


Figure 39: Location plots for single metal loop target with (a), (b), (c) $\alpha_t = 45°$ and $\beta_t = 45°$ and (d), (e), (f) $\alpha_t = 90°$ and $\beta_t = 90°$.

Some examples of extracted target orientation information for the single-loop target can be seen in Figure 40. The estimation is not perfect, but it is acceptably close for detection purposes. The estimated values are also not restrained to a discretization grid, outside of rounding decimal values. The fact that the discretization is avoided in the orientation space eliminates the "off-grid" modeling error, again outside of rounding the decimal values. The orientation information for the single relaxation frequency is also shown in a slightly different visualization in Figure 41, which draws out the 3D symmetry and orientation of the relaxation frequency at different parameter combinations. The orientation and symmetry plots are done by creating an ellipsoid with axes in the direction of the estimated tensors eigenvectors

and axes lengths set by the corresponding eigenvalues. The single-loop target will have a dipole symmetry, so a 3D line is what is expected in the plots in Figure 41. The first row of images, show the 2D projections of the 3D symmetry for a case where the estimated symmetry is extremely accurate. The second row, however, has the target at $y = 0\,\text{cm}$ with $\boldsymbol{o}_t = (90°, 90°)$, and no estimate could even be made for the tensor. However, the third row shows that when the target is moved off-center to $y = 10\,\text{cm}$ and kept at $\boldsymbol{o}_t = (90°, 90°)$, the symmetry is actually fairly accurately recovered.

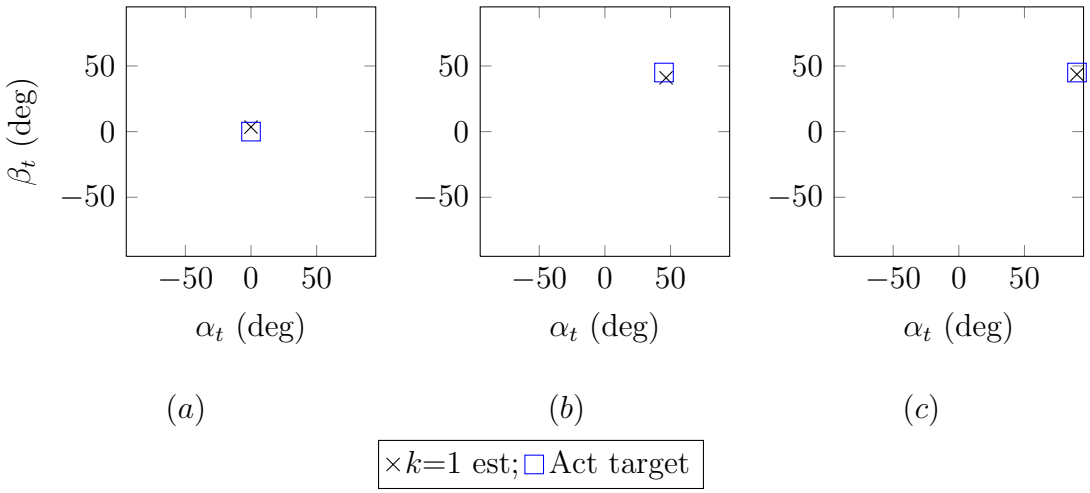


Figure 40: Angle plots for single metal loop target with $y_t = 0\,\text{cm}$ and $z_t = 8.5\,\text{cm}$.

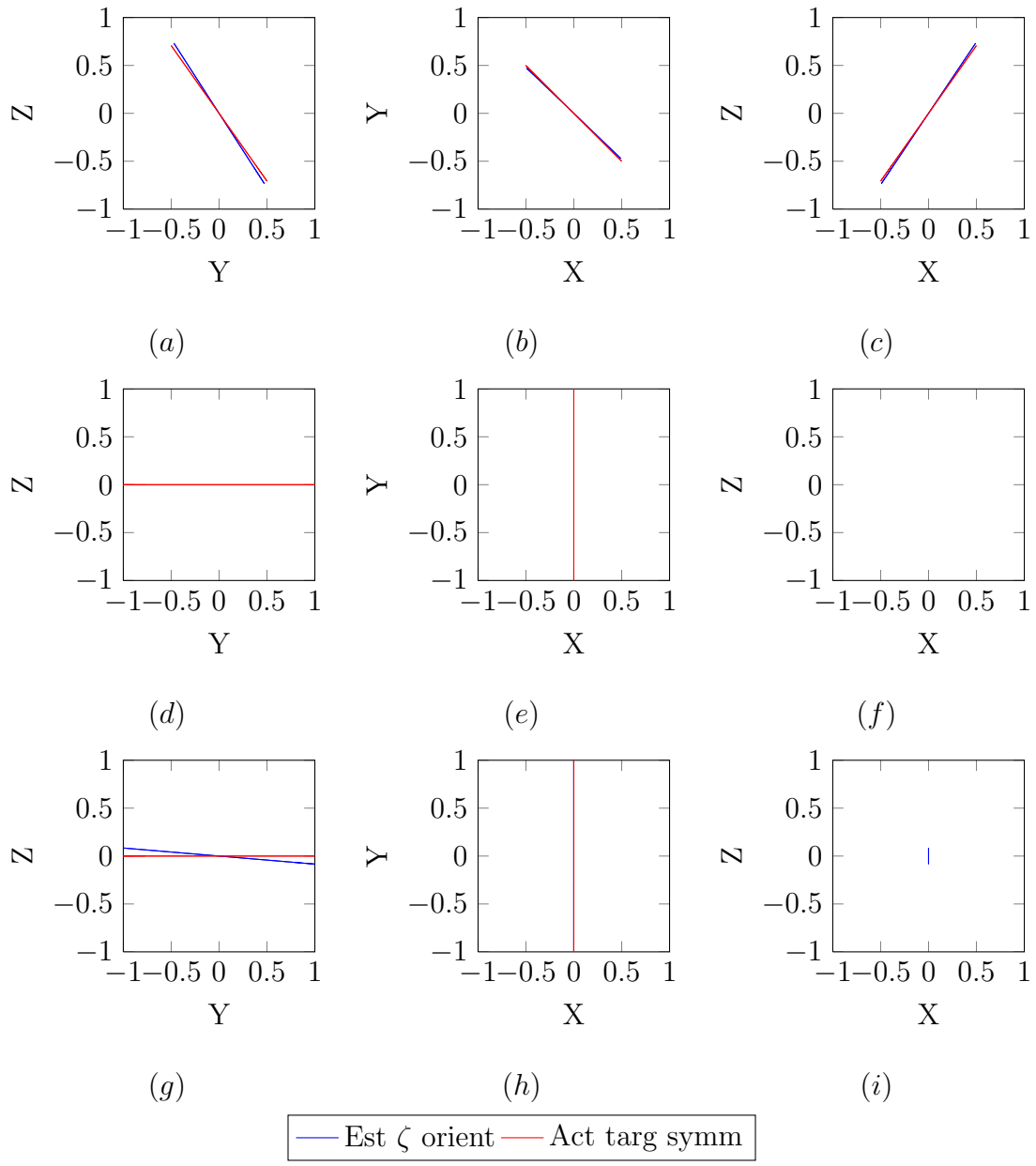


Figure 41: Orientation of the electromagnetic field of the single metal loop for (a), (b), (c) $y_t = 0\,\text{cm}$, $z_t = 7.5\,\text{cm}$, $\alpha_t = 45°$, and $\beta_t = 45°$; (d), (e), (f) $y_t = 0\,\text{cm}$, $z_t = 7.5\,\text{cm}$, $\alpha_t = 90°$, and $\beta_t = 90°$; (g), (h), (i) $y_t = 10\,\text{cm}$, $z_t = 7.5\,\text{cm}$, $\alpha_t = 90°$, and $\beta_t = 90°$.

Now that a few experiments have been individually examined, more data is needed

to analyze the detection on a slightly more general scale. The error histogram described in subsection 1.4.2 will be used to analyze performance. For these images, 1 corresponds to 100% errors, 0 corresponds to no errors. In the case of the location, if the euclidean distance between the estimated location and the actual location is more than 5 cm, it is counted as an error. The number of errors is tabulated for all orientation combinations at each location and then converted to an error percentage. Next, the 2D error histogram is shown in Figure 42(a). The same error calculation is done with each orientation for all locations, using a threshold of 22.5°. The error histogram for the orientations is shown in Figure 42(b).

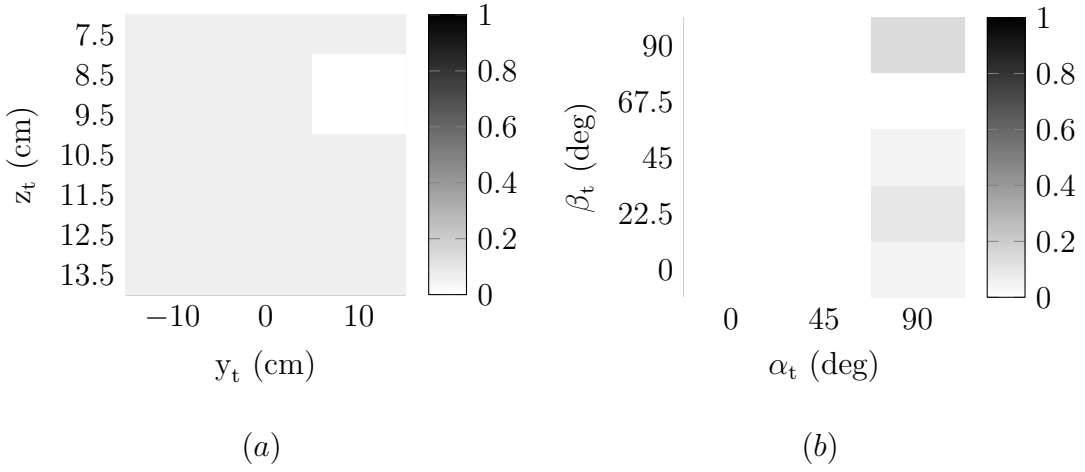


Figure 42: Error histograms for the single-loop target for (a) location and (b) angle.

The number of distance errors for these locations is extremely low using the 5 cm threshold. The angles however, start to become more difficult to image as they approach $\alpha_t = 90°$ and $\beta_t = 90°$ as has been anticipated throughout this chapter. Again, the detection difficulty comes from the way that the sensors are set up, and the way that measurements are taken. Dipoles that are directed only in the y direction produce extremely weak measurements that are nearly impossible for the sensor to distinguish over the noise. An example of a y-directed measurement was shown back in Figure 32(a), whereas a measurement with the target dipole directed perpendicular

to the sensor can be seen in Figure 32(b).

3.5.2 Triple-loop target

The triple-loop target is used to show how a single target, with multiple relaxations, would be imaged. The support of the DSRF for the triple-loop target can be seen in Figure 43. The three relaxation frequencies will be imaged separately in the same manner that the single relaxation frequency was imaged in the previous subsection. The triple-loop target is three single-loop targets placed together each in a direction of one of the main axes. For the notation, the subscript t will still correspond to the target as a whole, if the subscript is a number, for instance, α_1, the number corresponds to the relaxation frequency, $k = 1$. The locations should all be collocated for the targets in the experiment. In terms of the relaxation frequencies, $k = 1$ corresponds to the z-directed relaxation and the principle axis, the one that determines the overall orientation of the target. In other words, $\boldsymbol{o}_t = \boldsymbol{o}_1$. The relaxation $k = 2$ has a relative orientation with respect to $\boldsymbol{o}_t$ such that $\boldsymbol{o}_2 = (\alpha_t + 90°, 90°)$. The relaxation $k = 3$ has a relative orientation with respect to $\boldsymbol{o}_t$ such that $\boldsymbol{o}_2 = (\alpha_t, \beta_t + 90°)$.

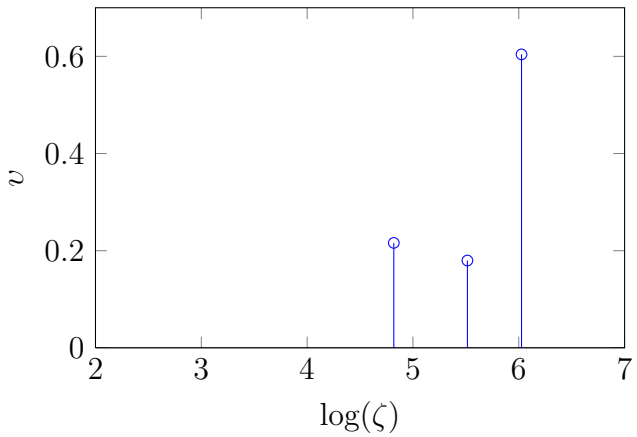


Figure 43: DSRF for triple-loop target, $k = 1, 2, 3$ from left to right.

A few individual experiments are run to show location estimation in Figure 44. The triple-loop target has three relaxation frequencies, so the different markings on

the plots correspond to the estimated location of each individual relaxation frequency. It is expected that all the relaxation frequencies would be located in the same place because they are parts of the same target. The first row of Figure 44 shows very accurate location information for all three relaxations, because none of the three are in $(90°, 90°)$. However, in the second row, k=1 happens to be at $(90°, 90°)$ and the location estimation is about as accurate as it was in the single-loop case. The advantage to having multiple relaxations in one target is that even if one component is difficult to image, the other components may not be, as can be seen by the imaging accuracy of k=2 and k=3 in the second row of Figure 44. While it is not being done here, there may be a way to intelligently cluster the location information of all the components to get a more accurate estimation of target location.

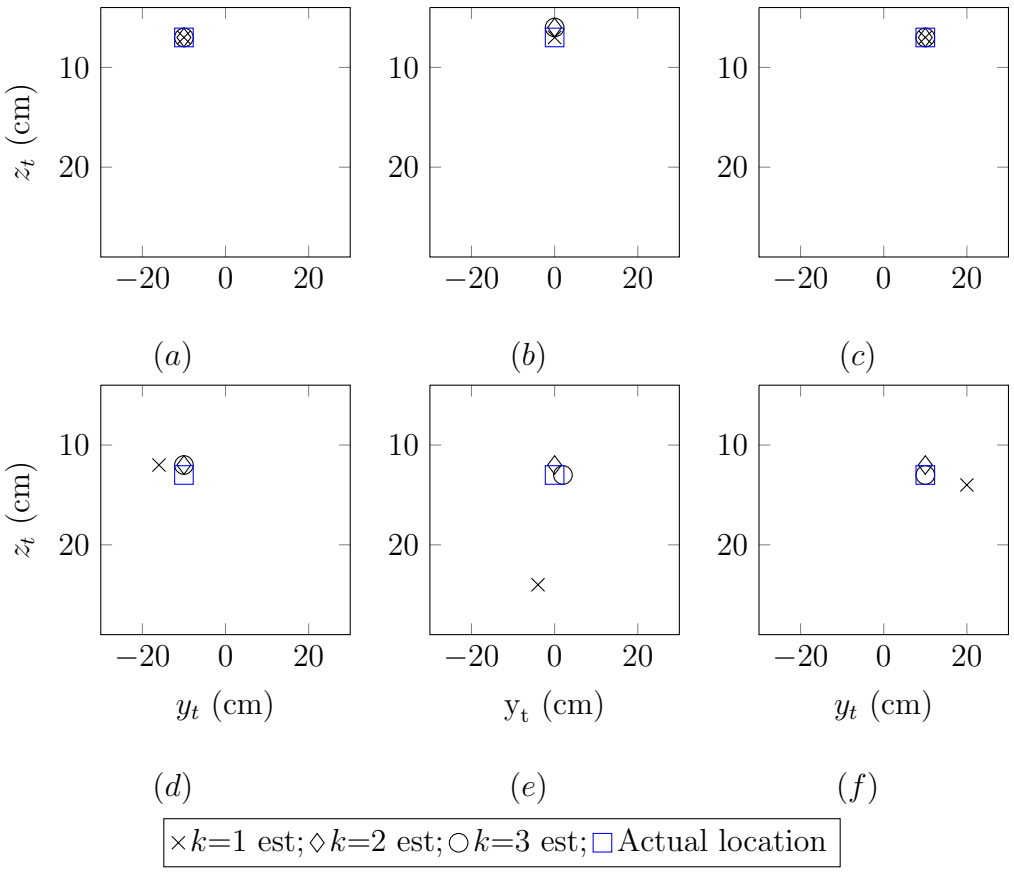


Figure 44: Location plots for triple-loop target with (a), (b), (c) $\alpha_t = 45°$ and $\beta_t = 45°$ and (d), (e), (f) $\alpha_t = 90°$ and $\beta_t = 90°$.

A few individual experiments are run to show angle estimation in Figure 45. The three separate relaxation frequencies are oriented orthogonal to each other in 3D. The angle plots can be slightly misleading if read incorrectly. For example, in Figure 45(a) the estimation for k=1 does not look visually close to the actual target orientation. However, with a β_1 error of only about 10°, having an error in α_1 will not contribute a large error in the 3D angle because it is still relatively close to $(0°, 0°)$. The estimation for $\boldsymbol{o}_2$ should be $(90°, 90°)$, but again that orientation is difficult to estimate correctly so an error there is expected. The estimation for $\boldsymbol{o}_3$ should be $(0°, 90°)$ and it is accurately estimated. In Figure 45(b) the target is oriented at a favorable $(45°, 45°)$, which means that none of the relaxations will be at $(90°, 90°)$, and the estimations

for all relaxations are acceptably close to the actual. Finally, in Figure 45(c) the estimation of a target at $(90°, 90°)$ is examined. The results should be very similar to the case in Figure 45(a), except the relaxations have all simply swapped orientations. Now that k=1 has an orientation of $(90°, 90°)$, it has similar error as to k=3 in Figure 45(a). The other two relaxations have been accurately estimated as expected.

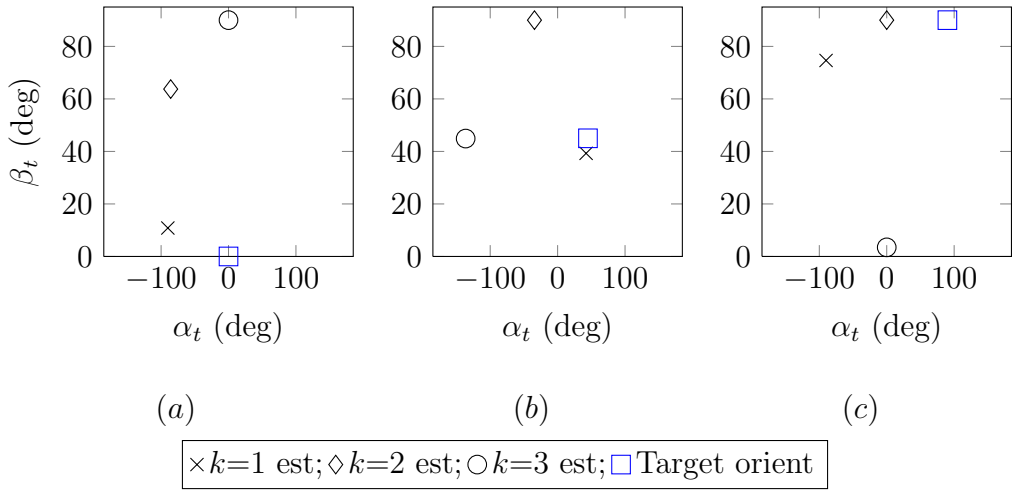


Figure 45: Angle plots for triple-loop target with $y_t = 0\,\text{cm}$ and $z_t = 8\,\text{cm}$.

The orientation information for the triple-loop target is shown in Figure 46 through plotting the symmetry and orientation just like was done in Figure 41. Again, the three relaxations are dipoles, and have one dimensional symmetries. Figure 46 shows a single experiment. The first row of Figure 46 corresponds to k=1, and the relaxation orientation is shown to match very closely to the target orientation. The second row of Figure 46 corresponds to k=2. The pitch, β_2, does not change by the rotations made in this experiment, and this is shown by having no z component in k=2. The yaw, α_2, which should have a value of $\alpha_2 = 90 + \alpha_t$, is shown to be close to perpendicular to the actually target symmetry in Figure 46(e), which is what is expected. Finally, the third row of Figure 46 corresponds to k=3. Remember that $\boldsymbol{o}_3 = (\alpha_t, \beta_t + 90)$, so the xy-plane should be the same for the estimate and the actual

target, but the estimated should be perpendicular to the actual for the zy-plane and the zx-plane, and that is what is observed.

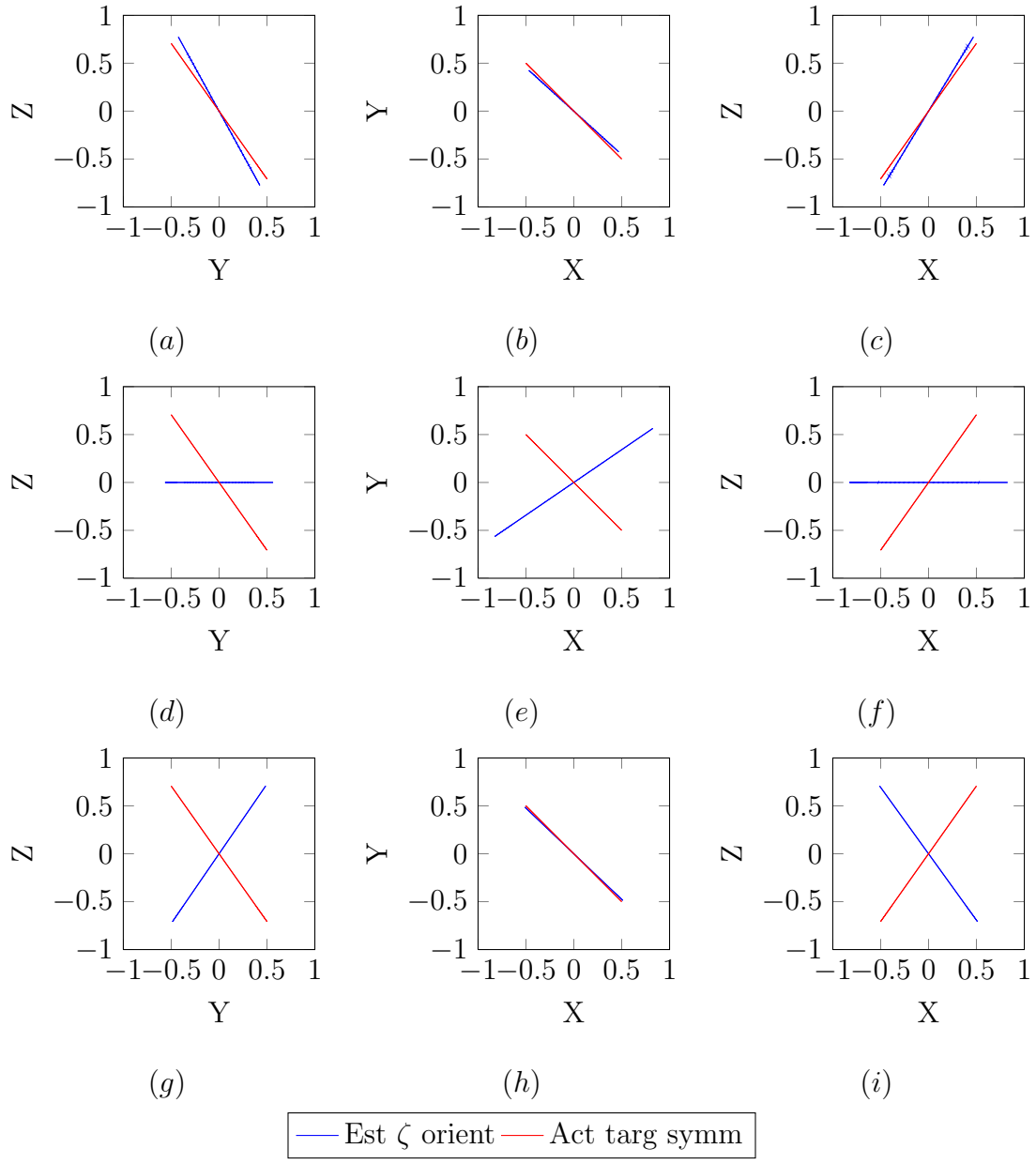


Figure 46: Orientation of the electromagnetic field of the triple-loop for $y_t = 0\,\text{cm}$, $z_t = 7.5\,\text{cm}$, $\alpha_t = 45°$, and $\beta_t = 45°$; (a), (b), (c) $k = 1$, (d), (e), (f) $k = 2$, and (g), (h), (i) $k = 3$.

The location and angle error histograms for the triple-loop target can be seen in

Figure 47. The locations have acceptably low probabilities of error, about equivalent to the ones calculated for the single-loop target. The angles on the histogram axes correspond to the orientation of the target as a whole, not each individual relaxation frequency. Changing the pitch of the target does not change the pitch of the dipole corresponding to the k=2 as was previously discussed, and this can be seen by the rather uniform error down the pitch axis of Figure 47(d). Also the target orientation $\alpha_t = 0$ corresponds to the heaviest amount of errors for k=2 and that is because again when $\alpha_t = 0$, $\alpha_2 = 90$.

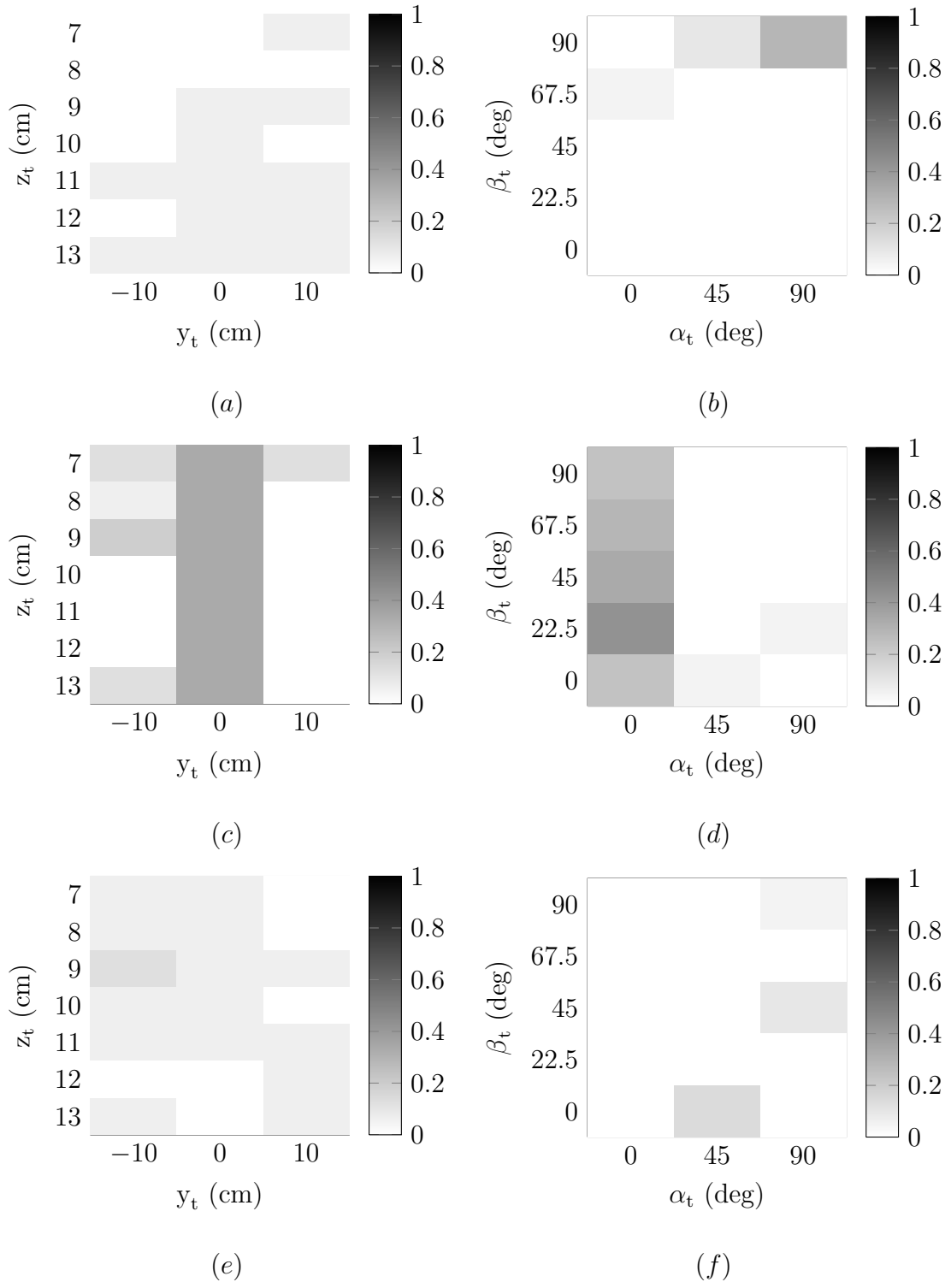


Figure 47: Error histograms for the triple-loop target for (a) location at $k = 1$, (b) angle at $k = 1$, (c) location at $k = 2$, (d) angle at $k = 2$,(e) location at $k = 3$, and (f) angle at $k = 3$.

3.5.3 Target A

Target A is an actual landmine that has a relatively strong response and two relaxations, both in the principle direction of the target. Having multiple relaxations again can give more opportunity to correctly estimate the target parameters. The DSRF for target A can be seen in Figure 48. Again, this target only has dipoles in its response, making the symmetries simple, and the ranks of the tensors 1.

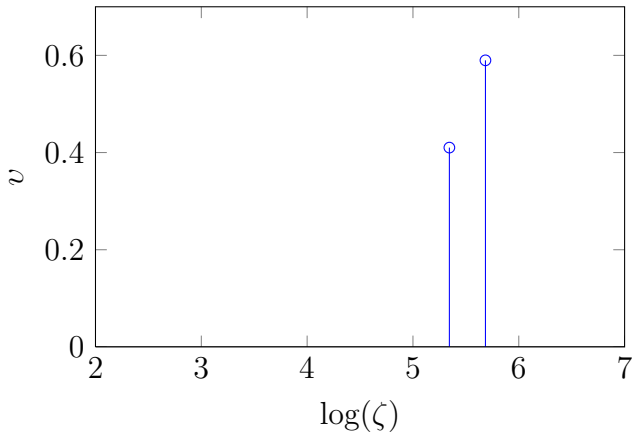


Figure 48: DSRF for target A, $k = 1, 2$ from left to right.

A few individual experiments were run to evaluate the estimation accuracy just like what was done with the metal-loop targets. The individual location estimation plots can be seen in Figure 49. When the target is relatively shallow and the orientation is not $(90°, 90°)$, as is the case in the first row of Figure 49, the location estimation is accurate. Once the target is turned to be at $(90°, 90°)$, the location estimation is not reasonable, as is seen in the second row of Figure 49. The angle estimation plots, Figure 50, show more of the same trends that were present in metal-loop experiments. The angle estimations are fairly accurate in the examples shown when $\boldsymbol{o}_t \neq (90°, 90°)$. When $\boldsymbol{o}_t = (90°, 90°)$, as is the case in Figure 50(c), the estimate is again inaccurate. The symmetries and orientations are again visualized in the plots in Figure 51. Both relaxation frequencies plotted show the dipole symmetry, and an accurate orientation

with respect to the target orientation, although the relaxation $k = 2$ is a more accurate than $k = 1$ for this particular experiment.

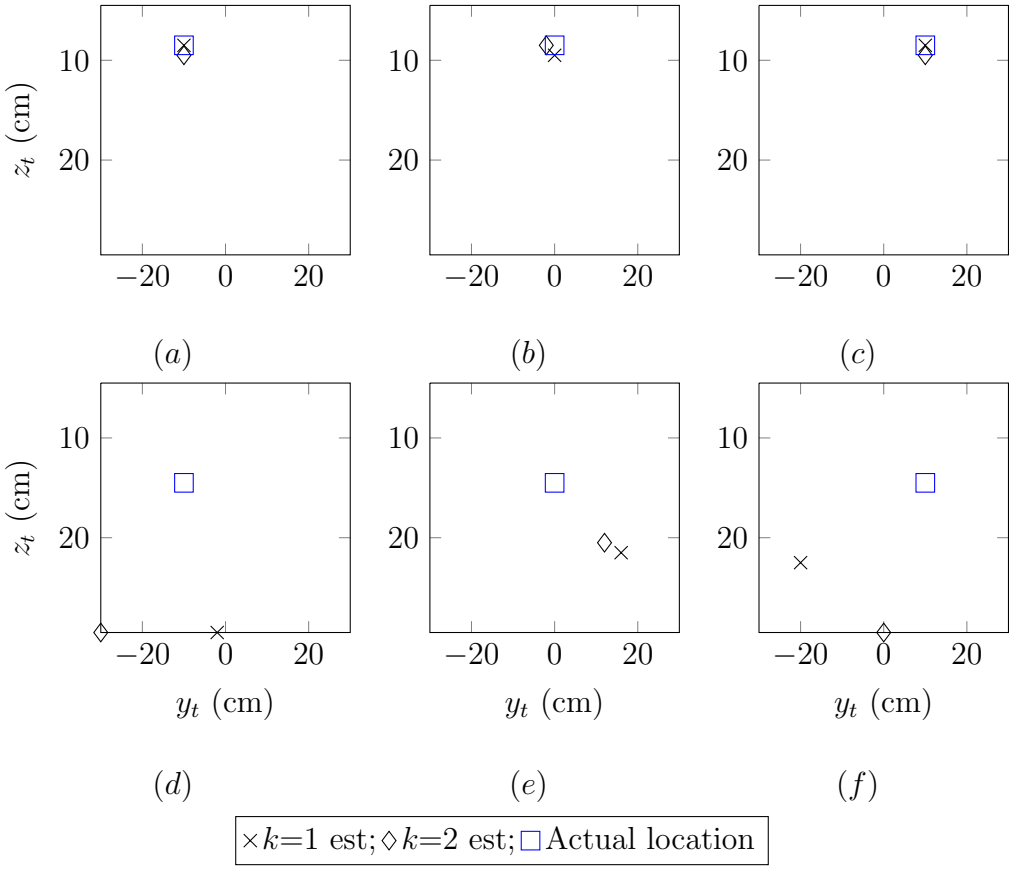


Figure 49: Location plots for a two-relaxation landmine target with (a), (b), (c) $\alpha_t = 45°$ and $\beta_t = 45°$ and (d), (e), (f) $\alpha_t = 90°$ and $\beta_t = 90°$.

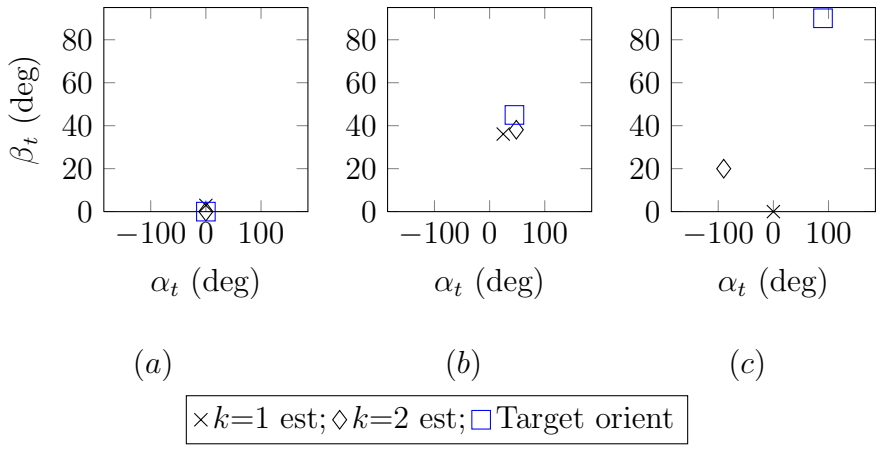


Figure 50: Angle plots for the two-relaxation landmine target with $y_t = 0\,\text{cm}$ and $z_t = 8.5\,\text{cm}$.

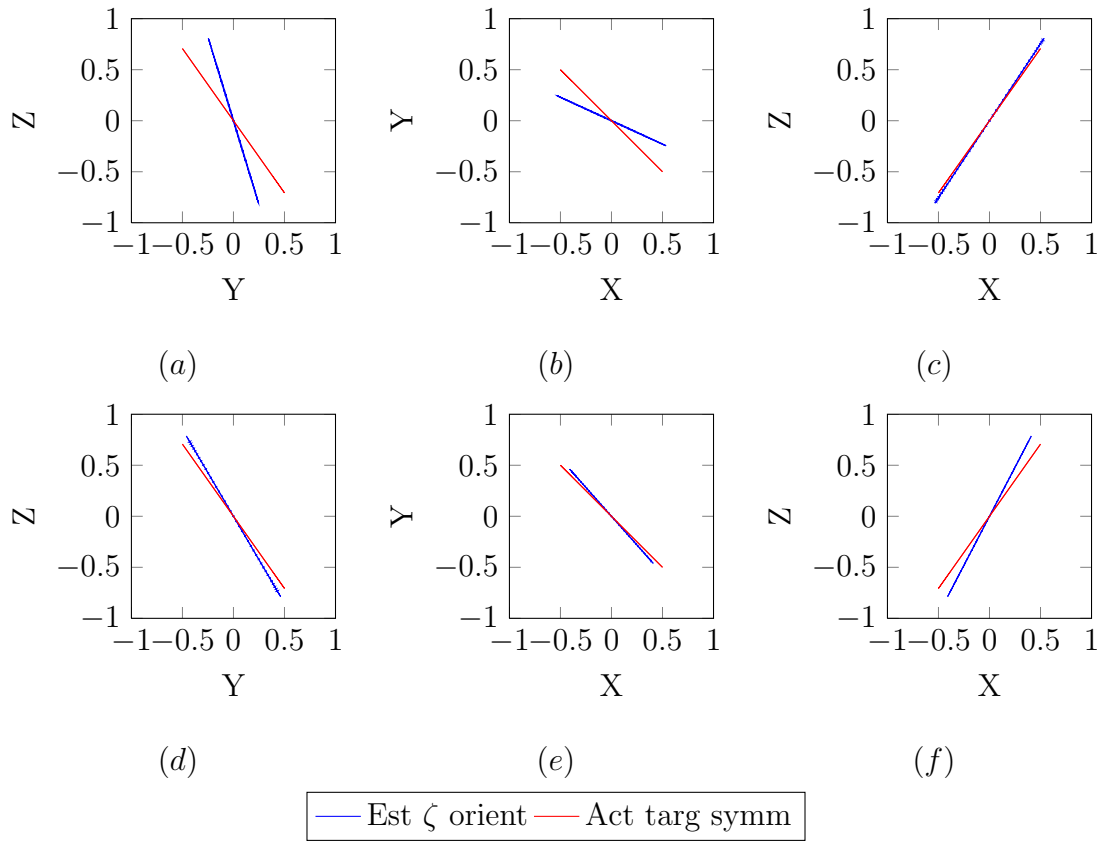


Figure 51: Orientation of the electromagnetic field of the two-relaxation landmine target for $y_t = 0\,\text{cm}$, $z_t = 8.5\,\text{cm}$, $\alpha_t = 45°$, and $\beta_t = 45°$; (a), (b), (c) $k = 1$, and (d), (e), (f) $k = 2$.

The error histograms for the two relaxations of target A can be seen in Figure 52. The probability of detection is lower for target A than it is for either of the metal-loop targets. This increased probability of error is most likely caused by the fact that the strength of target A is about an order of magnitude lower than the strength of the single-loop target. Target A is also hard for the sensor to see when the angle approaches $\alpha = 90°$ and $\beta = 90°$, like in the single-loop scenario.

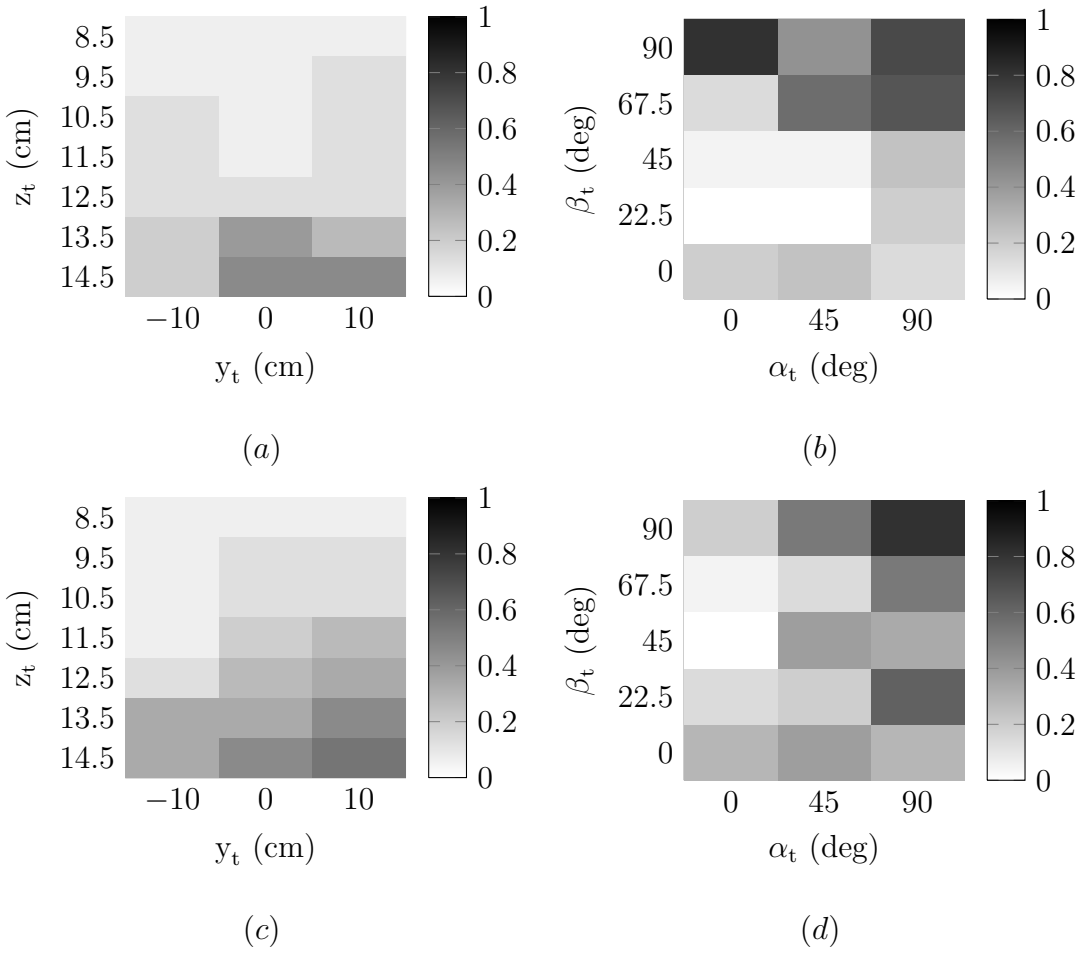


Figure 52: Error histograms for target A for (a) location at $k = 1$, (b) angle at $k = 1$, (c) location at $k = 2$, and (d) angle at $k = 2$.

3.5.4 Target B

Target B is another landmine, which has a fairly complex response with six relaxation frequencies. The DSRF and the magnetic polarizability of this target are only known experimentally. The DSRF of target B can be seen in Figure 53. The proposed method has been tested with targets having known parameters in the previous subsections, but the analysis of target B will be done somewhat blind, as it would be in practice. The target itself has been measured at the same parameter combinations as the other targets have, so the location is known, within the bounds of the mine

casing. A few of the experiments are individually examined in terms of locating the relaxation frequencies in Figure 54. Since target B has six relaxation frequencies, there are six different measurements that can help locate the target. The first row of Figure 54 shows a few experiments with the target at $z_t = 8.5\,\text{cm}$. The clustering of the relaxation frequencies around the target is promising for accurate detection. The second row of Figure 54 shows a few experiments with the target at $z_t = 11.5\,\text{cm}$. The location estimates for the different relaxation frequencies start to spread out, with a few relaxations, for example $k = 5$ in Figure 54(f), becoming extremely inaccurate. The third row of Figure 54 has targets located at $z_t = 14.5\,\text{cm}$, and has similar estimation accuracy to the second row. Knowing the location of the target for the experiments allows for the same location error histograms to be constructed for this target, and they can be seen in Figure 56. Other than the $k = 1$ relaxation, the probability of detecting the correct location is about as good as it is for target A. The relaxation at $k = 4$ seems to have the lowest probability of error, followed by $k = 5$ as long as the target is directly below the sensor.

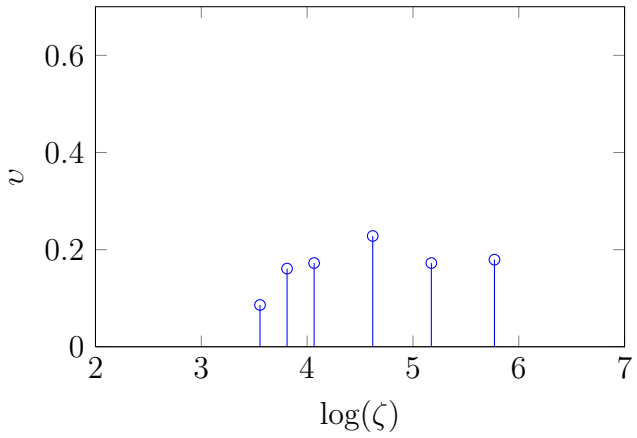


Figure 53: DSRF for target B, $k = 1, 2, 3, 4, 5, 6$ from left to right.

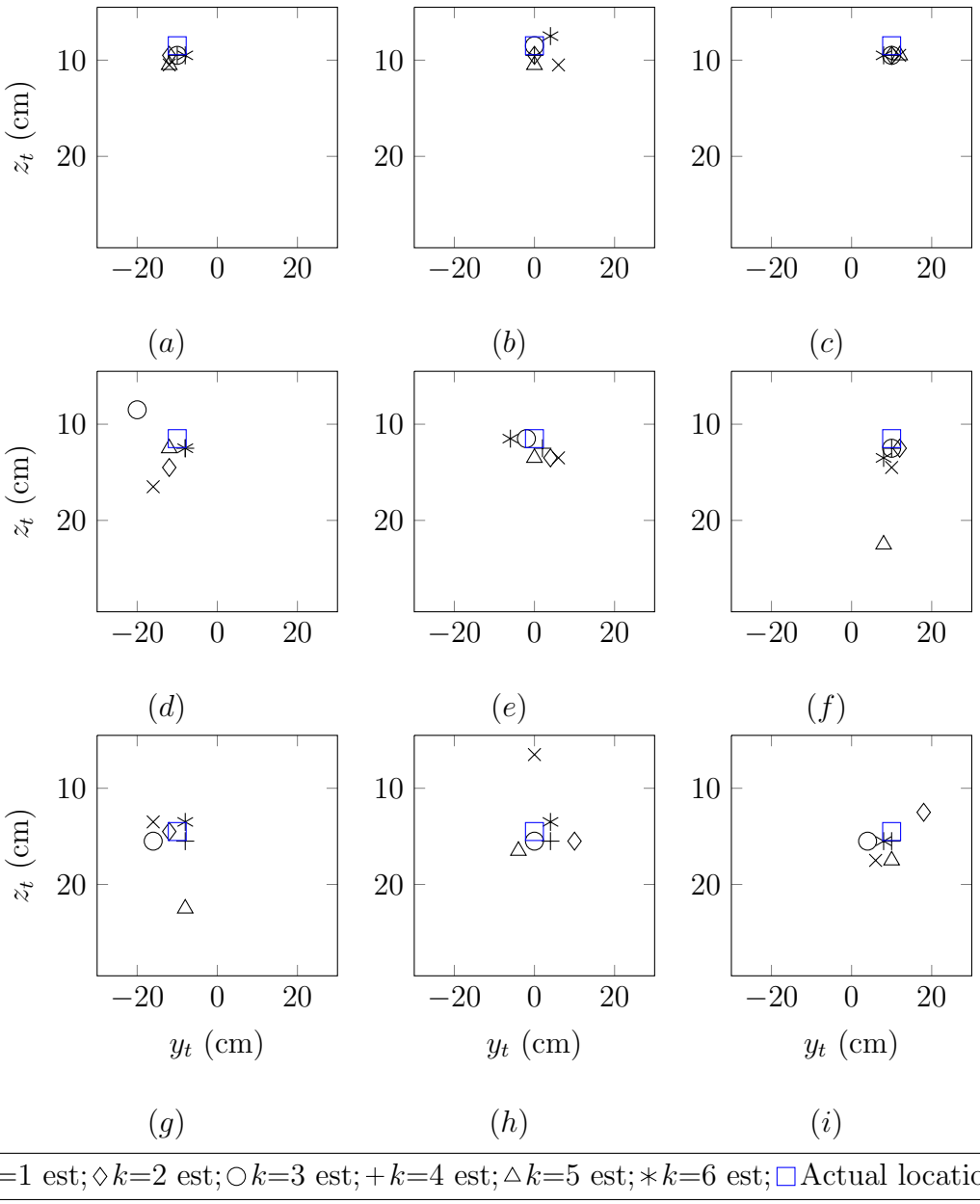


Figure 54: Location plots for a six-relaxation landmine target with $\alpha_t = 0°$ and $\beta_t = 0°$(a), (b), (c) $z_t = 8.5\,\text{cm}$, (d), (e), (f) $z_t = 11.5\,\text{cm}$, and (g), (h), (i) $z_t = 14.5\,\text{cm}$.

The tensor components of the relaxations for this target are not known theoretically, but they have been measured, see Figure 10 in [1], which is copied in this thesis as Figure 55 for easy access. From these measurements, each tensor is expected

(when the target is at the $\alpha_t = 0°$ and $\beta_t = 0°$ orientation) to consist of the superposition of 2 or 3 dipoles, with the x and y dipoles having roughly equal amplitudes (cylindrical symmetry) and the z-directed dipole being different. Since the tensors are complicated, and the sensors have trouble measuring the y component, an example evaluation will be done of the extracted tensors from a target B located 9.5 cm directly below the middle receive coil. Table 6 shows the six extracted tensors, and Figure 57 shows the symmetry of the $k = 5$ relaxation. The tensors are normalized so that the largest eigenvalue has a strength of 1. The extracted tensors only partially match the previous measurements. This discrepancy is not surprising since the sensor is insensitive to y-directed dipoles and the inversion was restricted to only a single measurement pass over the target. Consider relaxation 5 that consists of a x and z directed dipole. The symmetry, from the estimation, is a 2D ellipse with the major axis in the z direction and the minor axis in the x direction, which is shown in the first row of Figure 57. When the $k = 5$ relaxation is rotated in α, it would be expected that the ellipse would rotate around the z-axis. However, when the target is rotated by 45° in α, the resulting extracted tensor

$$\begin{bmatrix} .39 & .00 & -.02 \\ .00 & .00 & .00 \\ -.02 & .00 & 1.0 \end{bmatrix}$$

is almost identical to the pre-rotated tensor in Table 6. The extremely similar symmetries of the $k = 5$ relaxation rotated by 45° and 90° in α can be seen in the second and third rows of Figure 57. This would lead to the assumption that since the target symmetry is invariant to rotations in the α angle, that the symmetry in the xy plane is actually circular, and thus the symmetry of the $k = 5$ relaxation is cylindrical in 3D. The cylindrical symmetry more closely matches the empirical symmetry shown in Figure 55 at $k = 5$. The conclusion about the symmetry in $k = 5$ being cylindrical instead of elliptical would mean that the components in y are being suppressed by the

measurements system, which has been shown to be an issue in all of the previous laboratory experiments. So to get the most accurate measurement, the data-acquisition system should be enhanced to account for the suppressed y direction, or multiple passes over the target should be taken with the sensor itself oriented in different ways to build the most accurate measurement.

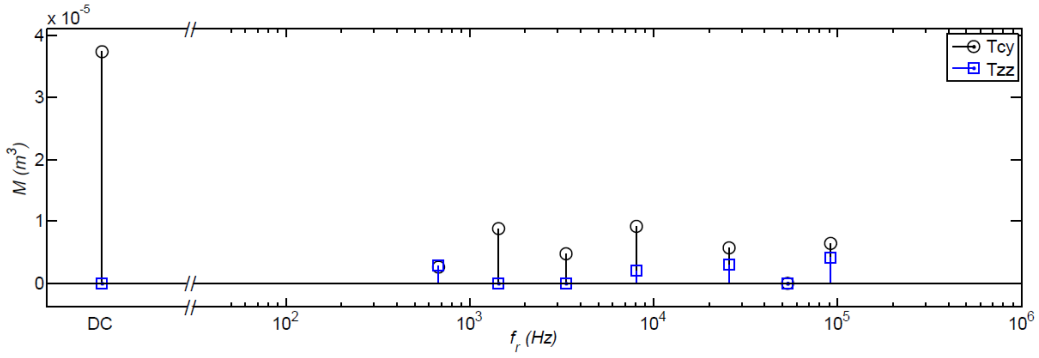


Figure 55: Empirical estimates of the magnetic polarizability of target B where $k = 0, 1, 2, 3, 4, 5, 6, 7$ from left to right, taken from [1].

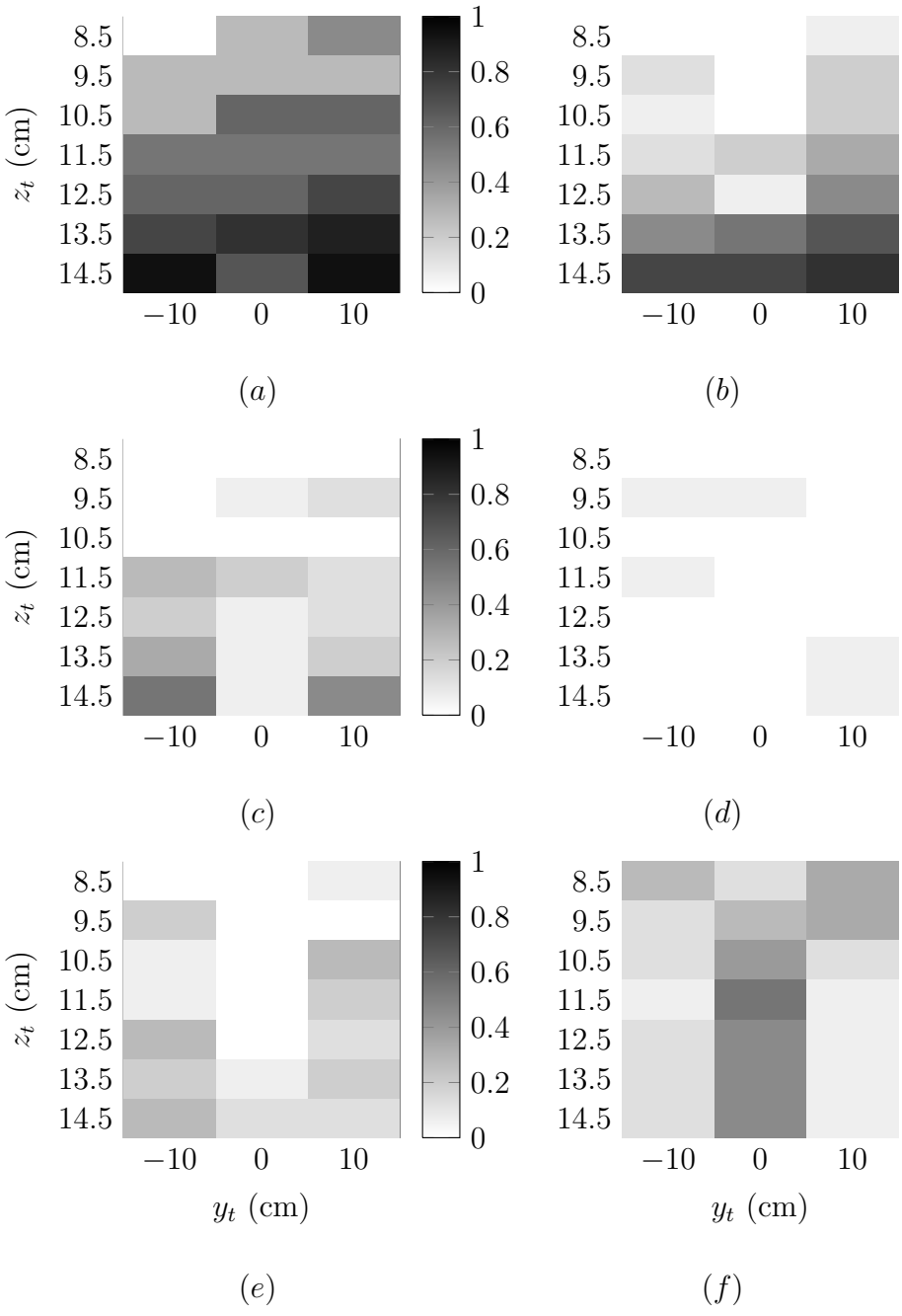


Figure 56: Location error histograms for target B at (a) $k = 1$, (b) $k = 2$, (c) $k = 3$, (d) $k = 4$, (e) $k = 5$, and (f) $k = 6$.

Table 6: Experimental target B tensors.

Relaxation #	Tensor	Nonzero Eigenvectors	Eigenvalues
1	$\begin{bmatrix} .01 & .03 & .08 \\ .03 & .10 & .30 \\ .08 & .30 & .89 \end{bmatrix}$	$\begin{bmatrix} .09 \\ .31 \\ .94 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
2	$\begin{bmatrix} .99 & -.09 & .00 \\ -.09 & .01 & .00 \\ .00 & .00 & .00 \end{bmatrix}$	$\begin{bmatrix} -.99 \\ .09 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
3	$\begin{bmatrix} 1.0 & .02 & .01 \\ .02 & .00 & .02 \\ .01 & .02 & .28 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & .28 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
4	$\begin{bmatrix} 1.0 & -.01 & .00 \\ -.01 & .00 & .00 \\ .00 & .00 & .79 \end{bmatrix}$	$\begin{bmatrix} 0 & -1 \\ 0 & 0 \\ 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & .79 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
5	$\begin{bmatrix} .49 & .00 & .00 \\ .00 & .00 & .00 \\ .00 & .00 & 1.0 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & .49 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
6	$\begin{bmatrix} .12 & .01 & -.01 \\ .01 & .37 & -.48 \\ -.01 & -.48 & .63 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 0 & -.61 \\ 0 & .79 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & .12 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

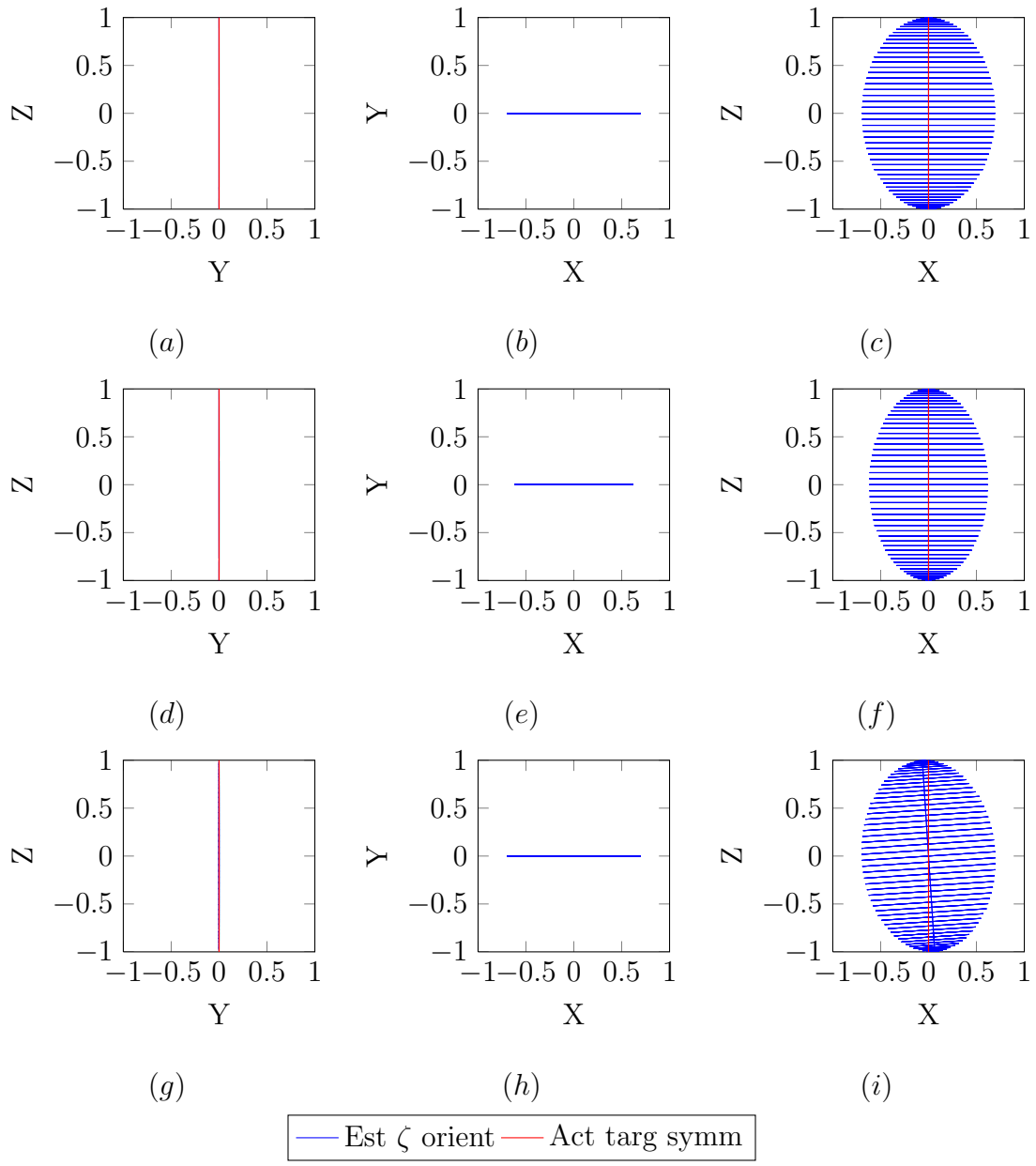


Figure 57: Orientation of the electromagnetic field of the six-relaxation landmine target for $y_t = 0\,\text{cm}$, $z_t = 9.5\,\text{cm}$, $k = 5$, (a), (b), (c) $\boldsymbol{o}_t = (0°, 0°)$, (d), (e), (f) $\boldsymbol{o}_t = (45°, 0°)$, and (g), (h), (i) $\boldsymbol{o}_t = (90°, 0°)$.

3.6 New Data-acquisition Techniques

The previous sections have discussed in detail the problems that arise with the current data-acquisition system. In an attempt to show that making changes to the system can improve, even if just slightly, the detection capability of these algorithms, three new data-acquisition systems are studied via simulation. The new systems incorporate adding new receivers, new transmitters, and/or taking multiple passes over a target with different sensor orientations.

The first new acquisition system, seen in Figure 58, consists of adding one new y-directed receive coil to the previous sensor geometry. Since no change has been made to the transmitter, even this new sensor will remain blind to targets directly beneath the middle of the transmitting coil and at $\boldsymbol{o}_t = (90°, 90°)$, but the additional receiver should help better detect targets that are close to this orientation and location. The response from the fourth receiver is stacked onto the response of the previously used receive coils to create the new response vector. A few examples of the new response vector can be seen in Figure 59. The fourth coil can be seen adding significant signal strength to targets with orientations in the y-direction. The increase in signal strength should garner a slight reduction in the error for the distance and the angle estimates.

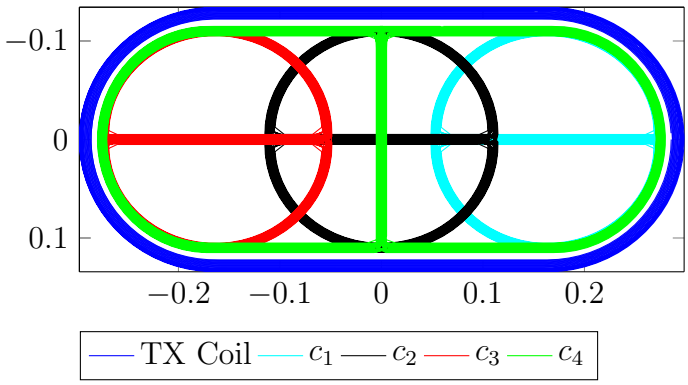
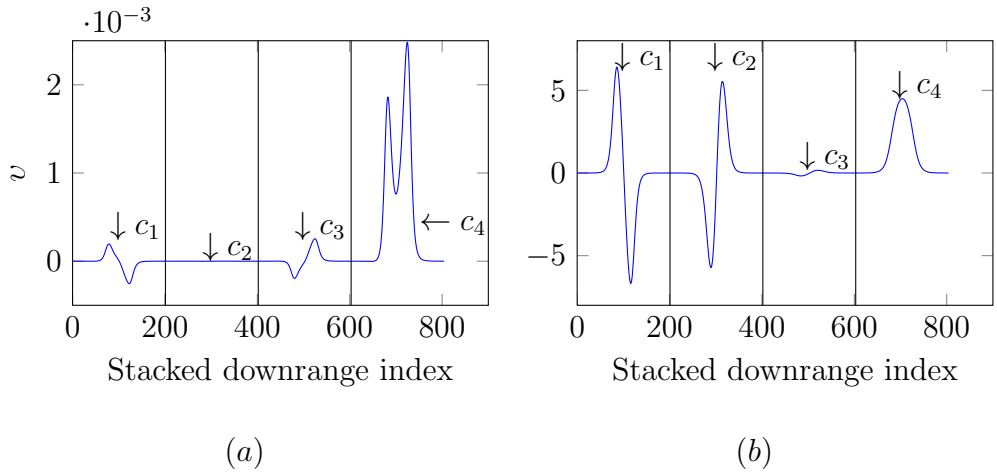


Figure 58: New sensor using a fourth receive coil that is added to the previous three.



(a) (b)

Figure 59: New measurements using the four receive coil sensor with $\boldsymbol{o}_t = (90°, 90°)$, (a) $\boldsymbol{l}_t = (0, 8.5)$ cm; and (b) $\boldsymbol{l}_t = (10, 8.5)$ cm.

The next system examined uses the same sensor setup as in Figure 58 but adds a new transmit-receive pair by transmitting and receiving on c_4. The addition of the new field transmitted in the y-direction should make the sensor much more robust to y-directed dipoles. The robustness in the y-direction can be seen by examining Figure 60. The new response stacked onto the measurements at index 805, c_4^{TX}, is done by taking the receive field on c_4 after transmitting on c_4. The new c_4^{TX} response can be seen adding a significant amount of signal power to the measurements of a y-directed dipole. However, adding multiple transmitters can make things difficult in a practical system.

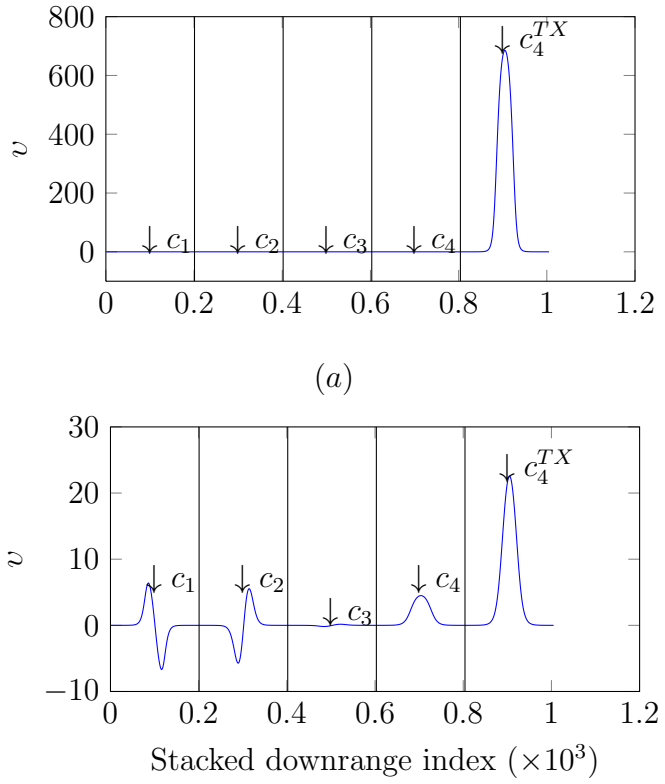


Figure 60: New measurements using the four receive coil sensor where the fourth receiver is also used as a transmitter with $\boldsymbol{o}_t = (90°, 90°)$, (a) $\boldsymbol{l}_t = (0, 8.5)\,\text{cm}$; and (b) $\boldsymbol{l}_t = (10, 8.5)\,\text{cm}$.

The final data-acquisition system examined is one that is created by taking two passes over the target while changing the orientation of the sensors. The sensor used in this simulation is the same sensor that was discussed in the previous sections of this thesis, and seen in Figure 61. However, the sensor is now oriented in $\alpha = -45°$ in the first pass, Figure 61(a), and rotated to $\alpha = 45°$ in the second pass, Figure 61(b). Using two orthogonal passes guarantees that the sensor will not be completely "blind" during both passes. Two example response vectors are shown in Figure 62. The measurements made from the two passes are concatenated together. When the target is directly below the center receive sensor, $y = 0\,\text{cm}$, the two passes look like reflections

of each other, although not exact reflections, because of the orthogonality of the passes, Figure 62(a). The two-pass system will likely not be quite as robust as the two-transmitter system, because there is no transmitted field in the y-direction, but the two-pass system uses less additional hardware and would be slightly more practical for use in a real environment.

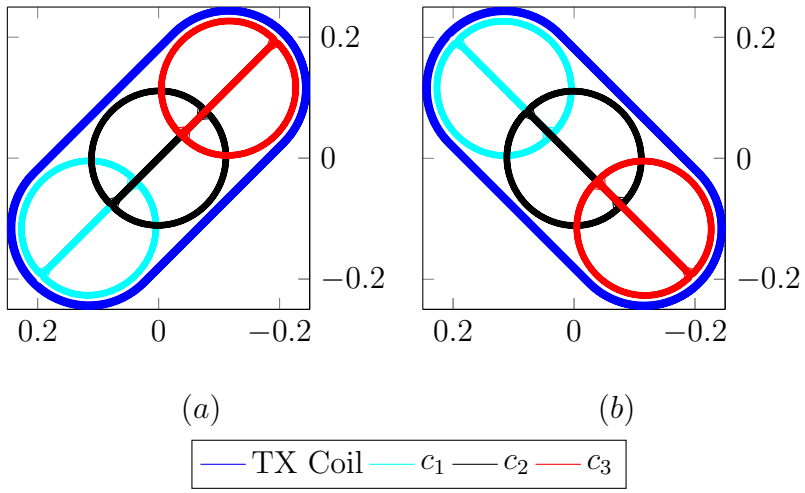


Figure 61: New measurement setup using only the three receive coils, but using two passes of the sensor with the sensor oriented at (a) $\alpha = -45°$ for pass one and (b) $\alpha = 45°$ for pass two.

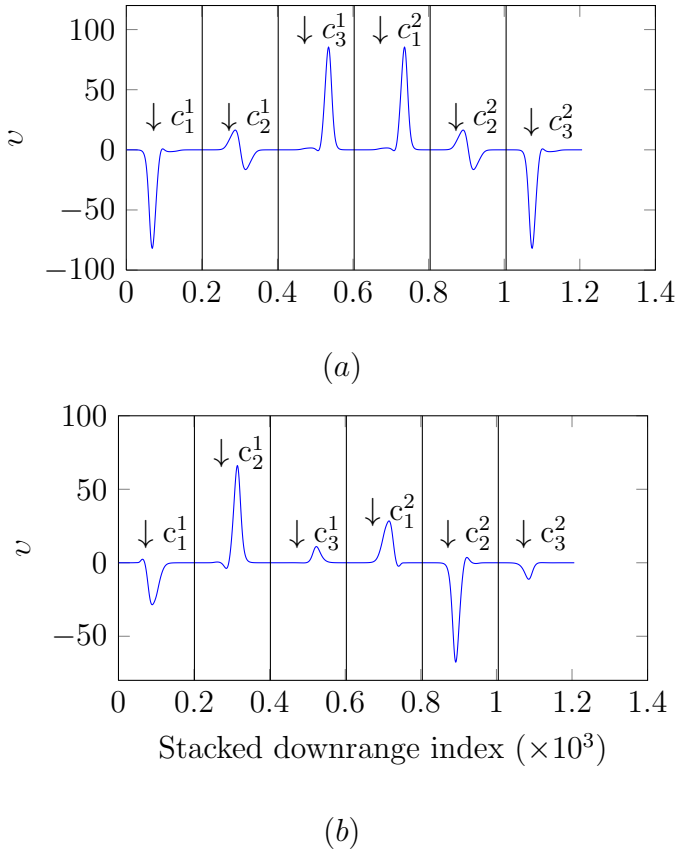


Figure 62: New measurements using the two-pass measurement setup with $\boldsymbol{o}_t = (90°, 90°)$, (a) $\boldsymbol{l}_t = (0, 8.5)\,\text{cm}$; and (b) $\boldsymbol{l}_t = (10, 8.5)\,\text{cm}$.

A small simulation was run, creating a new response dictionary using each of the new data-acquisition systems, adding noise to some of the entries of the dictionary to create the measurements, and trying to accurately estimate the parameters. The distance-error plots with respect to an increasing noise power can be seen in Figure 63 and the angle error plots can be seen in Figure 64. The rows of Figure 63 correspond to changing y_t from top, $y_t = -2\,\text{cm}$, middle, $y_t = 0\,\text{cm}$, and bottom, $y_t = 2\,\text{cm}$; and the columns correspond to changing the orientation from left, $\boldsymbol{o}_t = (90°, 90°)$, middle, $\boldsymbol{o}_t = (90°, 80°)$, and right, $\boldsymbol{o}_t = (90°, 70°)$. These orientations were chosen since they are problematic when using the original sensor. The acquisition adjustments should aid in the detection of other orientations as well. The errors for the target

at $y_t = 0\,\text{cm}$ and $\boldsymbol{o}_t = (90°, 90°)$, seen in Figure 63(d) and Figure 64(d), using the standard setup and the four-coil setup are never accurate for any noise power. The lack of accuracy in each of these setups is expected because the transmitter cannot induce a magnetic field on the target. However, adding the transmitter in the y-direction for the two-transmitter setup, or taking two orthogonal passes over the target, allow for the previously "invisible" target to be seen. In the general sense, adding c_4 without transmitting on it, allows for very little reduction in the location error, roughly 2 cm, or the angle error, 0° to 30°. The location estimation for targets with the specified parameters is much more accurate when the two-transmitter setup or the two-pass setup is used. The angle estimation for the two-transmitter setup in these simulations is much better than the rest for higher noise powers. However, when compared to the standard setup and the four-coil setup, the two-pass setup is much more accurate for lower noise powers, is not blind to y-directed targets, and is more consistent across the changing target parameters. The consistency of the two-pass setup can be seen in the similarity of the error plots for the two-pass setup with each different target location and orientation.

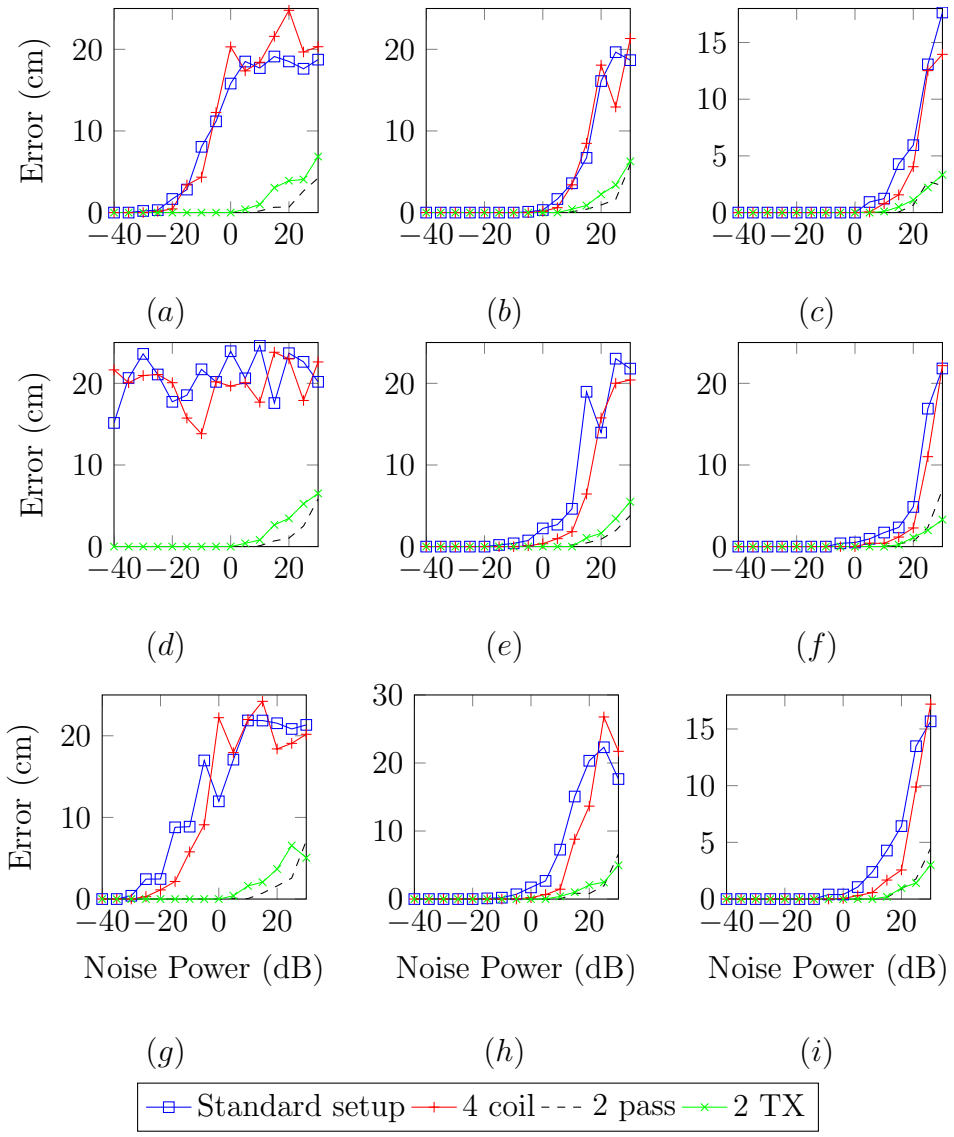


Figure 63: Location error comparison for the four different geometries examined for (a) $\boldsymbol{o}_t = (90°, 90°)$ and $\boldsymbol{l}_t = (-2, 8.5)\,\text{cm}$; (b) $\boldsymbol{o}_t = (90°, 80°)$ and $\boldsymbol{l}_t = (-2, 8.5)\,\text{cm}$; (c) $\boldsymbol{o}_t = (90°, 70°)$ and $\boldsymbol{l}_t = (-2, 8.5)\,\text{cm}$; (d) $\boldsymbol{o}_t = (90°, 90°)$ and $\boldsymbol{l}_t = (0, 8.5)\,\text{cm}$; (e) $\boldsymbol{o}_t = (90°, 80°)$ and $\boldsymbol{l}_t = (0, 8.5)\,\text{cm}$; (f) $\boldsymbol{o}_t = (90°, 70°)$ and $\boldsymbol{l}_t = (0, 8.5)\,\text{cm}$; (g) $\boldsymbol{o}_t = (90°, 90°)$ and $\boldsymbol{l}_t = (2, 8.5)\,\text{cm}$; (h) $\boldsymbol{o}_t = (90°, 80°)$ and $\boldsymbol{l}_t = (2, 8.5)\,\text{cm}$; (i) $\boldsymbol{o}_t = (90°, 70°)$ and $\boldsymbol{l}_t = (2, 8.5)\,\text{cm}$.

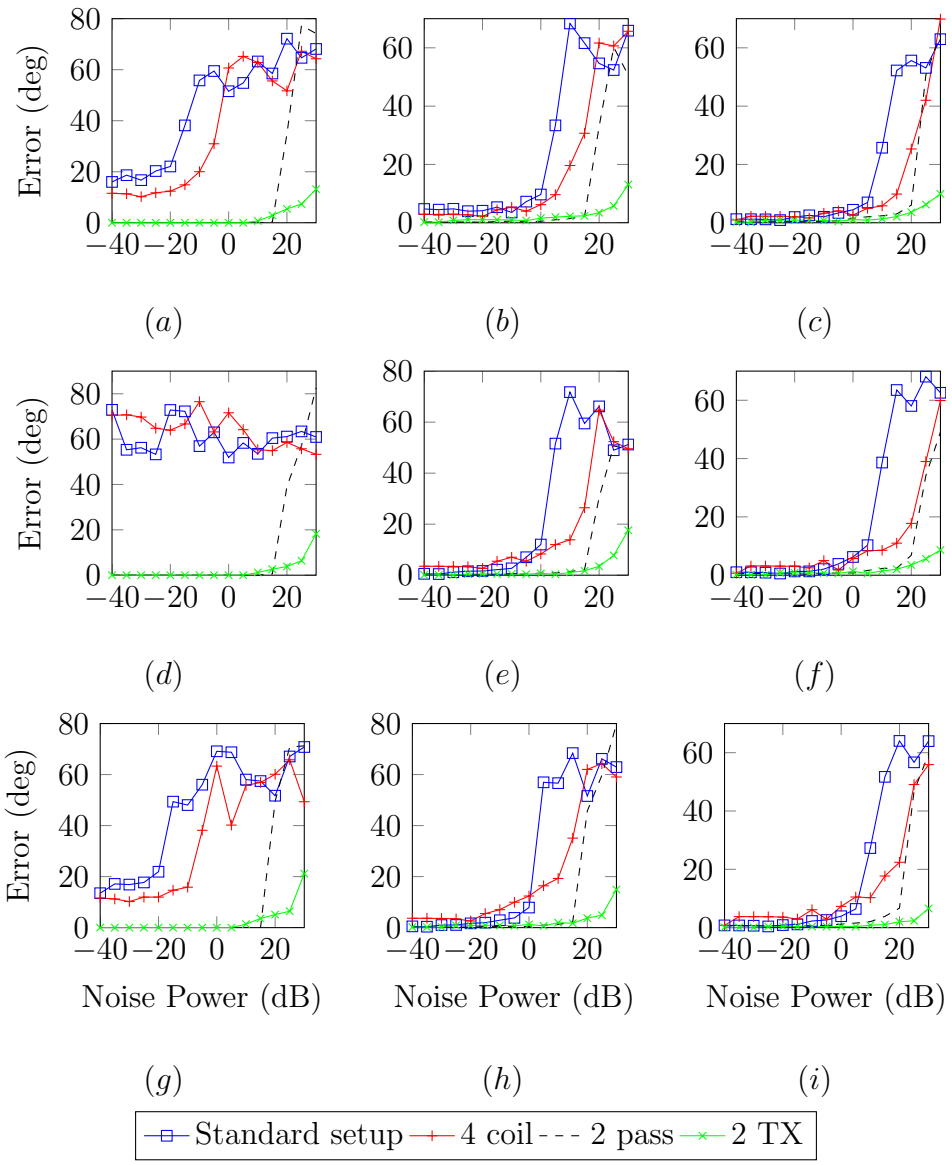


Figure 64: Angle error comparison for the four different geometries examined for (a) $\boldsymbol{o}_t = (90°, 90°)$ and $\boldsymbol{l}_t = (-2, 8.5)\,\text{cm}$; (b) $\boldsymbol{o}_t = (90°, 80°)$ and $\boldsymbol{l}_t = (-2, 8.5)\,\text{cm}$; (c) $\boldsymbol{o}_t = (90°, 70°)$ and $\boldsymbol{l}_t = (-2, 8.5)\,\text{cm}$; (d) $\boldsymbol{o}_t = (90°, 90°)$ and $\boldsymbol{l}_t = (0, 8.5)\,\text{cm}$; (e) $\boldsymbol{o}_t = (90°, 80°)$ and $\boldsymbol{l}_t = (0, 8.5)\,\text{cm}$; (f) $\boldsymbol{o}_t = (90°, 70°)$ and $\boldsymbol{l}_t = (0, 8.5)\,\text{cm}$; (g) $\boldsymbol{o}_t = (90°, 90°)$ and $\boldsymbol{l}_t = (2, 8.5)\,\text{cm}$; (h) $\boldsymbol{o}_t = (90°, 80°)$ and $\boldsymbol{l}_t = (2, 8.5)\,\text{cm}$; (i) $\boldsymbol{o}_t = (90°, 70°)$ and $\boldsymbol{l}_t = (2, 8.5)\,\text{cm}$.

3.7 Conclusion

A new model representation and inversion strategy have been introduced for the landmine parameter-estimation problem using EMI sensors. The ability to directly extract the tensors reduces the scalability of the dictionary storage by $\mathcal{O}(N^3)$. Using the tensor extraction with the dipole model reduces the storage from an impractical $\mathcal{O}(N^9)$ to a much more practical $\mathcal{O}(N^4)$. Also, new data-acquisition systems are analyzed to help make suggestions on how the hardware system could be changed to account for some deficiencies.

CHAPTER IV

CONCLUSIONS

In this thesis, two separate structures are acknowledged and exploited to help improve the algorithmic efficiency of two different EM models for the purpose of landmine imaging. Each structure reduces the computational complexity of their particular algorithms to such a degree that the problems went from completely impractical in real-world applications to practical. Each method reduced the storage order of the associated algorithms by atleast $\mathcal{O}(N^2)$, which for the particular applications could practically be four orders of magnitude. An outline of the reduction techniques and their consequences on each different system are written out in detail in Figure 65. Although they have not been exploited in this particular research, compressive algorithms and shift invariance could very well be exploited in the sum of dipoles model to garner additional computational reductions if the inversions are examined and altered appropriately.

The main contribution of this research was identifying and exploiting the shift-invariance property in the GPR point-target model, and the "tensor amplitude" formulation of the EMI sum of dipoles model. These structural changes are not specific to the landmine problem, but would be present in any problem where there is some sort of invariance present in the underlying model, or if the object can be represented by a multidimensional field, in the case of the "tensor amplitude."

Chapter 1 discussed the motivation and the background for the GPR and EMI problems. The point-target model and the dipole model are introduced along with the associated imaging algorithms typically used. The inherent drawbacks to using the algorithms, BP, OMP, and CS, are in the scalability of the dictionary that is created

by enumerating a nonlinear model into a linear matrix. Each one of the algorithms is explained in short detail, and a brief description of the metrics used for analyzing accuracy are discussed.

Chapter 2 goes on to introduce the shift-invariance property in the point-target model associated with the GPR problem. A detailed explanation of how to exploit the shift-invariance property in BP, OMP, and CS to garner a dramatic reduction in required storage and computation time is discussed. Tests were run using synthetic data on multiple dimensions to determine the validity of the improvements and finally a test was run on previously imaged laboratory data to compare the new method to previous methods. In addition to the improved model structure, a framework was laid out to show that with current technology, a relatively effective compressive sensing GPR could be developed and used fairly easily.

Chapter 3 introduces the "tensor amplitude" which is used in the sum of dipoles model in the EMI problem. Exploiting this property not only reduces storage constraints but it also eliminates the "off-grid" model error associated with linearizing a nonlinear model through enumeration in the orientation space. Many different real targets are analyzed and imaged from controlled laboratory experiments. Three new data-acquisition systems are introduced to show that changing the hardware setup can increase the accuracy of the inversions.

Further work has been identified in both the advancement of the GPR and EMI systems. For the GPR system, uniform sampling is not going to be a guarantee in real systems, as some consist of a hand-held wand that is scanned over the ground in a non-uniform manner. There should be a method created that can easily allow nonuniform sampling but still be able to have the shift-invariant property. For the EMI system, inherent issues with the data-acquisition system were confirmed and suggestions have been made on how to address the issues for someone in hardware development. These suggestions include adding additional coils so the sensors are not

blind to certain target parameters, or adding a systematic way to achieve multiple looks at the target in an efficient manner. There are still computational constraints associated with solving the large SDP problem, which could be addressed in the future. The algorithms should be tested on field data and an algorithm should be created that can be more robust to multiple targets while exploiting the inherent shift invariance of this model. Finally, the algorithms data-acquisition process should be examined to see if a CS algorithm could be used to enhance the data-acquisition or inversion processes.

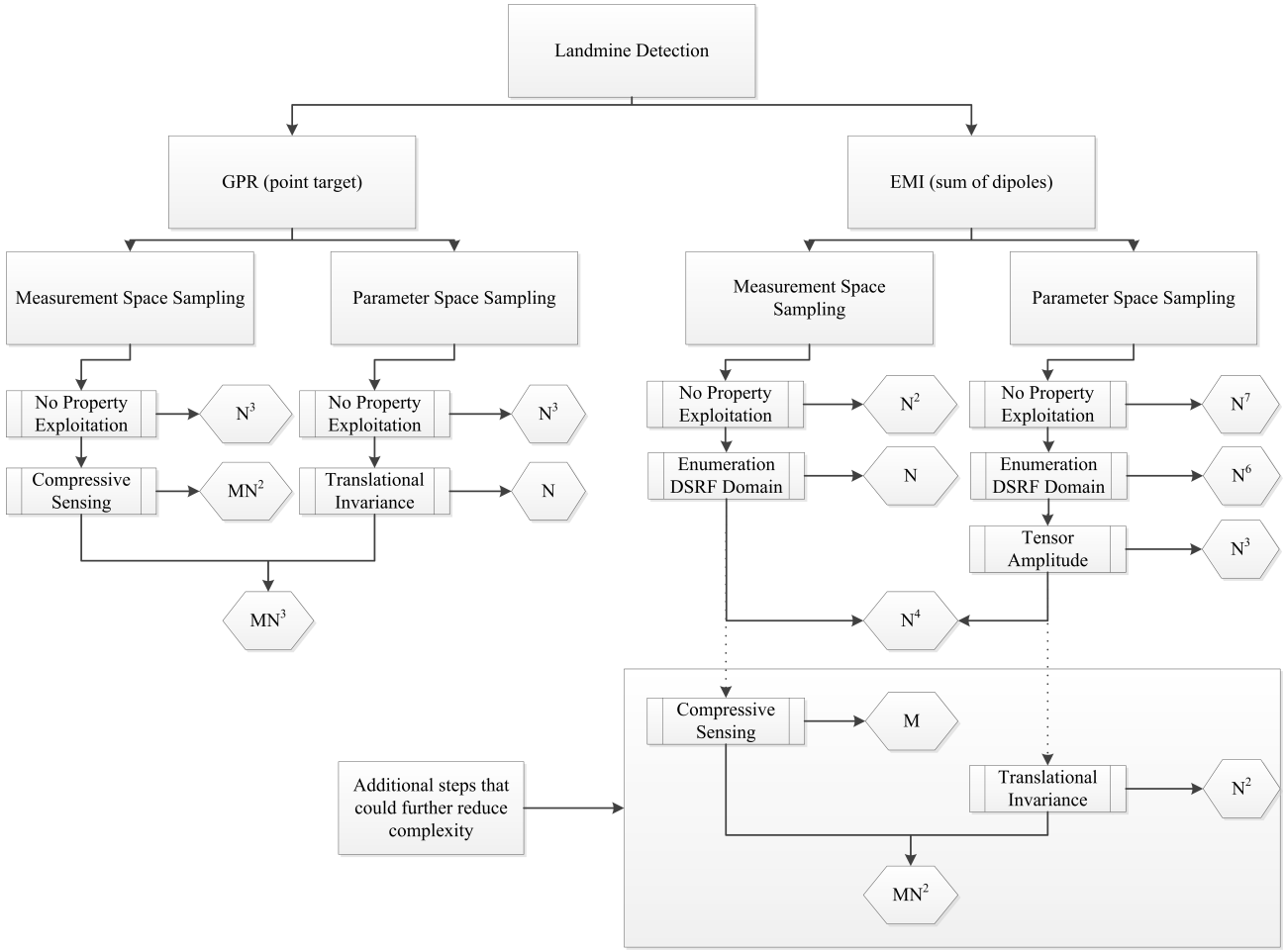


Figure 65: Summary chart of reduction techniques for landmine detection.

REFERENCES

[1] W. R. Scott, Jr. and G. D. Larson, "Measured dipole expansion of discrete relaxations to represent the electromagnetic induction response of buried metal targets," *Proc. SPIE*, vol. 7664, 2010.

[2] International Compaign to Ban Landmines, *Landmine Monitor 2012*, 2012.

[3] T. Counts, A. C. Gurbuz, W. R. Scott, Jr., J. H. McClellan, and K. Kim, "Multistatic ground-penetrating radar experiments," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 8, pp. 2544–2553, Aug. 2007.

[4] A. C. Gurbuz, W. R. Scott, Jr., and J. H. McClellan, "Location estimation using a broadband electromagnetic induction array," *Proc. SPIE*, vol. 7303, 2009.

[5] S. R. J. Axelsson, "Analysis of random step frequency radar and comparison with experiments," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 4, pp. 890–904, 2007.

[6] L. P. Peters, Jr., J. J. Daniels, and J. D. Young, "Ground penetrating radar as a subsurface environmental sensing tool," *Proc. IEEE*, vol. 82, no. 12, pp. 1802–1822, 1994.

[7] D. J. Daniels, "Surface-penetrating radar," *Electronics & communication engineering journal*, vol. 8, no. Aug., pp. 165–182, 1996.

[8] R. M. Lemer, "Ground radar system," Patent U.S. Patent 3 831 173, 1974.

[9] J. D. Young, L. Peters, Jr., and C. Chen, "Characteristic resonance identification techniques for buried targets seen by ground penetrating radar," in *Detection and identification of visually oscured targets*, 1999, pp. 103–162.

[10] P. Millot and A. Berges, "Ground based SAR imaging tool for the design of buried mine detectors," *Detection of abandoned land mines*, no. 431, pp. 7–9, 1996.

[11] G. F. Stickley, "Synthetic aperture radar for the detection of shallow buried objects," *Detection of abandoned land mines*, no. 431, pp. 7–9, 1996.

[12] A. C. Gurbuz, J. H. McClellan, and W. R. Scott, Jr., "Compressive sensing for subsurface imaging using ground penetrating radar," *Signal Processing*, vol. 89, no. 10, pp. 1959–1972, Oct. 2009.

[13] ——, "A compressive sensing data acquisition and imaging method for stepped frequency GPRs," *IEEE Transactions on Signal Processing*, vol. 57, no. 7, pp. 2640–2650, Jul. 2009.

[14] L. J. Cutrona and E. N. Leith, "On the application of coherent optical processing techniques to synthetic-aperture radar," *Proc. IEEE*, vol. 54, no. 8, pp. 1026–1032, 1966.

[15] W. M. Brown and L. J. Porcello, "An introduction to synthetic-aperture radar," *IEEE Spectrum*, no. Sep., 1969.

[16] J. C. Kirk, "A discussion of digital processing in synthetic aperture radar," *IEEE Transactions on aerospace and electronic systems*, no. 3, 1975.

[17] Geneva International Centre for Humanitarian Demining, *Detectors and Personal Protective Equipment Catalogue*, 2009.

[18] D. A. Keiswetter, I. J. Wona, J. Miller, T. Bell, E. Cespedes, and K. O'Neill, "Discriminating capabilities of multifrequency EMI data," *International Geoscience and Remote Sensing Symposium*, pp. 1415–1417, 2000.

[19] P. Gao, L. Collins, P. M. Garber, N. Geng, and L. Carin, "Classification of landmine-like metal targets using wideband electromagnetic induction," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 38, no. 3, pp. 1352–1361, 2000.

[20] E. B. Fails, P. A. Torrione, W. R. Scott, Jr., and L. M. Collins, "Performance of a four parameter model for modeling landmine signatures in frequency domain wideband electromagnetic induction detection systems," *Proc. SPIE*, vol. 6553, 2007.

[21] M. Wei, W. R. Scott, Jr., and J. H. McClellan, "Robust Estimation of the Discrete Spectrum of Relaxations for Electromagnetic Induction Responses," *IEEE Transactions on Geoscience and Remote Sensing*, pp. 1–11, 2009.

[22] M. Wei, J. H. McClellan, and W. R. Scott, Jr., "Estimation of the discrete spectrum of relaxations for electromagnetic induction responses using ℓp-regularized least squares for $0 \leq \mathrm{p} \leq 1$," *IEEE Geoscience and Remote Sensing Letters*, pp. 233–237, 2010.

[23] J. T. Miller, T. H. Bell, J. Soukup, and D. Keiswetter, "Simple phenomenological models for wideband frequency-domain electromagnetic induction," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 39, no. 6, pp. 1294–1298, Jun. 2001.

[24] G. Arfken, *Mathematical Methods for Physicists*, 2nd ed. San Diego, CA: Academic Press, 1985.

[25] C. E. Baum, "Detection and identification of mines from natural magnetic and electromagnetic resonances," in *Detection and identification of visually oscured targets*, 1999, pp. 163–218.

[26] M. Özdemir, E. L. Miller, and S. Norton, "Localization and characterization of buried objects from multi-frequency, array inductive data," *Proc. SPIE*, Apr. 1999.

[27] M. Wei, W. R. Scott, Jr., and J. H. McClellan, "Estimation of the discrete spectrum of relaxation frequencies using multiple measurements," *International Geoscience and Remote Sensing Symposium*, pp. 586–589, 2012.

[28] M. Richards, *Fundamentals of Radar Signal Processing.* New York, NY: McGraw-Hill, 2005.

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

[30] A. C. Gurbuz, "Sparsity enhanced fast subsurface imaging with GPR," in *International Conference on Ground Penetrating Radar (GPR)*, Jun. 2010, pp. 1–5.

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

[32] R. G. Baraniuk, "Compressive sensing [lecture notes]," *IEEE Signal Processing Magazine*, vol. 24, no. July, pp. 118–121, 2007.

[33] E. J. Candès and J. Romberg, "Sparsity and incoherence in compressive sampling," *Inverse problems*, no. 3, pp. 1–20, 2007.

[34] E. J. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on Information Theory*, pp. 1–41, 2006.

[35] E. J. Candès and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Processing Magazine*, vol. 25, no. Mar., pp. 21–30, 2008.

[36] J. Romberg, "Compressive sensing by random convolution," *SIAM Journal on Imaging Science*, Dec. 2009.

[37] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Journal on Scientific Computing*, vol. 43, no. 1, pp. 129–159, 2001.

[38] E. J. Candès and T. Tao, "The Dantzig selector: Statistical estimation when p is much larger than n," *The Annals of Statistics*, vol. 40698, pp. 1–37, 2007.

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

[40] C. Beck and R. D'Andrea, "Computational study and comparisons of lft reducibility methods," in *American Control Conference, 1998. Proceedings of the 1998*, vol. 2, jun 1998, pp. 1013 –1017 vol.2.

[41] P. Boufounos, M. F. Duarte, and R. G. Baraniuk, "Sparse signal reconstruction from noisy compressive measurements using cross validation," in *Workshop on Statistical Signal Processing*, 2007, pp. 299–303.

[42] Y. Rubner, C. Tomasi, and L. J. Guibas, "A Metric for Distributions with Applications to Image Databases," *International Conference on Computer Vision*, 1998.

[43] O. Pele and M. Werman, "Fast and robust earth mover's distances," in *ICCV*, 2009.

[44] E. M. Johansson and J. E. Mast, "Three-dimensional ground-penetrating radar imaging using synthetic aperture time-domain focusing," *Proc. SPIE*, vol. 2275, pp. 205–214, Sep. 1994.

[45] M. Tuncer and A. Gurbuz, "Ground reflection removal in compressive sensing ground penetrating radars," *Geoscience and Remote Sensing Letters, IEEE*, vol. 9, no. 1, pp. 23–27, 2012.

[46] K. R. Krueger, J. H. McClellan, and W. R. Scott, Jr., "Dictionary reduction technique for 3-D stepped-frequency GPR imaging using compressive sensing and the FFT," *Proc. SPIE*, Apr. 2012.

[47] ——, "3-D imaging for ground penetrating radar using compressive sensing with block-toeplitz structures," in *Sensor Array and Multichannel Signal Processing Workshop (SAM), 2012 IEEE 7th*, June 2012, pp. 229 –232.

[48] ——, "Sampling techniques for improved algorithmic efficiency in electromagnetic sensing," in *International Conference on Sampling Theory and Applications*, July 2013.

[49] W. U. Bajwa, J. D. Haupt, G. M. Raz, S. J. Wright, and R. D. Nowak, "Toeplitz-structured compressed sensing matrices," in *Workshop on Statistical Signal Processing*, Aug. 2007, pp. 294–298.

[50] E. van den Berg and M. P. Friedlander, "SPGL1: A solver for large-scale sparse reconstruction," June 2007, http://www.cs.ubc.ca/labs/scl/spgl1.

[51] E. Candès and J. Romberg, "ℓ1-magic," Oct. 2005.

[52] M. Grant and S. Boyd, "Cvx: Matlab software for disciplined convex programming," 2008, http://cvxr.com/cvx/.

[53] J. Sachs, "M-sequence ultra-wideband-radar: state of development and applications," in *Radar Conference, 2003. Proceedings of the International*, 2003, pp. 224–229.

[54] M. Kahrs, "50 years of RF and microwave sampling," *Microwave Theory and Techniques, IEEE Transactions on*, vol. 51, no. 6, pp. 1787–1805, 2003.

[55] G. J. Frye and N. S. Nahman, "Random sampling oscillography," *Instrumentation and Measurement, IEEE Transactions on*, vol. 13, no. 1, pp. 8–13, 1964.

[56] Agilent Technologies, *What is the difference between an equivalent time sampling oscilloscope and a real-time oscilloscope?*, ser. Application Note 1608, 2008.

[57] C. Ekanadham, D. Tranchina, and E. P. Simoncelli, "Recovery of sparse translation-invariant signals with continuous basis pursuit," *IEEE Transactions on Signal Processing*, pp. 4735–4744, Oct. 2011.

[58] K. R. Krueger, J. H. McClellan, and W. R. Scott, "Tensor amplitude extraction in sensor array processing," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, 2013, pp. 3895–3899.

[59] K. R. Krueger, W. R. Scott, Jr., and J. H. McClellan, "Location and continuous orientation estimation of buried targets using tensor extraction," *Proc. SPIE*, Jun. 2013.

[60] ——, "Location and orientation estimation of buried targets using electromagnetic induction sensors," *Proc. SPIE*, Apr. 2012.

[61] W. R. Scott, Jr. and G. D. Larson, "Modeling the measured EM induction response of targets as a sum of dipole terms each with a discrete relaxation frequency," July 2010, pp. 4188–4191.