### STATISTICAL METHODS FOR UNEXPLODED ORDNANCE DISCRIMINATION

A Thesis Presented

by

#### Alireza Aliamiri

 $\operatorname{to}$ 

The Department of Electrical and Computer Engineering

In partial fulfillment of the requirements

for the degree of

#### Doctor of Philosophy

in

**Electrical Engineering** 

in the field of

Communication and Signal Processing

Northeastern University Boston, Massachusetts

December 2006

©Copyright by Alireza Aliamiri

All Rights Reserved

### NORTHEASTERN UNIVERSITY Graduate School of Engineering

Thesis Title: Statistical Methods for Unexploded Ordnance Discrimination

Author: Alireza Aliamiri

**Department**: Electrical Engineering

Approved for Thesis Requirement of Doctor of Philosophy Degree:

Thesis Advisor: Prof. Eric Miller Date Thesis Committee Member: Prof. Dana Brooks Date Thesis Committee Member: Prof. Jennifer Dy Date Thesis Committee Member: Dr. Jack Stalnaker Date Department Chair: Prof. Ali Abur Date Director of Graduate School: Prof. Yaman Yener Date Copy Deposited in Library: **Reference Librarian** Date To my parents

### Abstract

We propose statistical processing methods and performance analysis techniques for discrimination and localization of Unexploded Ordnance (UXOs) using EMI sensors based on nonparametrically defined prior probability density functions for target-relevant features. In the first part of this thesis, new sets of UXO discrimination methods using these nonparametric prior models are introduced where we use kernel density estimation (KDE) methods to build a priori probability density functions (PDFs) for the vector of features used to classify Unexploded Ordnance items given electromagnetic induction (EMI) sensor data. This a priori information is then used to develop a new suite of estimation and classification algorithms. As opposed to the commonly used maximum likelihood (ML) parameter estimation methods, here we employ a maximum a posteriori (MAP) estimation algorithm that makes use of the KDE-generated PDFs. Similarly, we use the KDE priors to develop new suite of classification schemes operating in both "feature" space as well as "signal/data" space. In terms of featurebased methods, we construct a support vector machine (SVM) classifier and its extension to support *M*-ary classification. The KDE PDFs are also used to synthesize a MAP feature-based classifier. To address numerical challenges associated with the optimal, data-space Bayesian classifier, we have constructed several approximations, including one based on a Laplacian approximation and a new hybrid approach that makes use of ML estimates of the parameters not essential to classification. Generalized likelihood ratio tests employing the priors are also considered. Using both simulations and real field data, we observe significant improvement in classification performance due to the use of the KDE-based prior models.

In second part of this thesis, we develop analytical performance analysis for the developed methods, where we derive analytical bounds on estimation and error performance including Cramer-Rao lower bounds (CRLB) and a Chernoff upper bound. The bounds are derived analytically and confirmed using Mont-Carlo simulations. Using the CRLBs the effect of unknown object geometry on estimation performance is analyzed.

In third part of this work analytical bounds for error performance are used to optimize

system performance with respect to sensor parameters in order to develop sensors that are tuned for best discrimination performance. To this end we have investigated the effect of sensor size and spatial sampling on error performance using the developed analytical bounds.

Finally we have considered the problem of sensor signal processing in the presence of sensor positional uncertainty where we have introduced a fusion of random walk and Markov chain models for modeling the correlated positional perturbation and developed low complexity Min-Max estimation routines using the Viterbi algorithm. The results using Min-Max approach show significant improvement compared to the case that sensor positional uncertainty is ignored.

### Acknowledgements

I would like to express my deep and sincere gratitude to my supervisor, Prof. Eric Miller, who not only shared his profound scientific knowledge with me but also taught me great lessons of life. His support, suggestions and encouragement gave me the drive and will to complete this work.

I wish to express my warm and sincere thanks to Dr. Jack Stalnaker for his collaboration to this research.

I am grateful to my parents for their love and support.

Finally, I want to thank Dr. Sara Ghassemi, who showed me the beauty of love and Farbod Farhanghmehr for his immortal friendship.

Boston, Massachusetts December 31, 2006 Alireza Aliamiri

## Table of Contents

Ał	bstract	iv											
Ac	Acknowledgements												
Ta	Table of Contents												
Li	ist of Tables	ix											
Li	ist of Figures	x											
1	Introduction	1											
2	Physical Model and Prior Work         2.1       Dipole Model         2.2       Basic Processing	<b>10</b> 10 13											
3	Feature Prior Model         3.1       Kernel Density Estimation         3.1.1       Whitening Transformation	<b>17</b> 17 18											
4	Estimation and Classification Methods         4.1       Estimation Methods         4.2       Classification Methods         4.2.1       Optimal classification         4.2.2       Signal Space Classifiers         4.2.3       Feature Space Classifiers         4.3       Classification Results	<ul> <li>23</li> <li>23</li> <li>24</li> <li>24</li> <li>25</li> <li>27</li> <li>29</li> </ul>											
5	Performance Analysis         5.1       Estimation And Classification Methods         5.1.1       Estimation         5.1.2       Optimal Classification         5.2       Estimation Performance Analysis         5.2.1       Cramer-Rao Lower Bounds	<b>41</b> 41 42 43 43											
	5.3 Error Probability Analysis	51											

		5.3.1	Optimum Data Space Classifier	51
		5.3.2	Feature Based Classifier	59
6	Tow	vards S	Sensor Optimization	65
	6.1	Upper	Bound on Error Probability	65
	6.2	Minim	nization of Error Upper Bound	68
		6.2.1	Optimal Sensor Dimension	68
		6.2.2	Optimal Dimension of Sensing Area	70
		6.2.3	Optimal Spatial Sampling	80
7	Sen	sor Po	sitional Uncertainty	83
	7.1	Proble	em Formulation	83
	7.2	Marko	ov Chain/ Random Walk Model	85
	7.3	Min-M	fax Estimation Using Viterbi Algorithm	89
	7.4	Nume	rical Results	90
8	Cor	nclusio	n and Future Work	92
А	$\mathbf{E}\mathbf{M}$	T Mod	els	95
	A.1	The D	Dipole Model	95
	A.2	Sensor	r-Specific Calibration of the Physical Model	98
		A.2.1	Magnetic Field of a Rectangular Coil	98
		A.2.2	Magnetic Field of a Circular Coil	99
в	Dat	a Surv	/ev	101
_	B.1	Data 1	Preprocessing	101
	B.2	Model	Parameter Estimation	102
		B.2.1	Parameter Space Search	103
		B.2.2	Particle Swarm Optimization (PSO)	104
		B.2.3	Model Parameter Estimation from Field Data	106
С	Hid	den M	[arkov Mode]	111
U	$C_{1}$	Hidder	n Markov Model Structure	111
	$C_{2}$	Viterb	a Algorithm	112
	C.3	Traini	ng the Trellis	113
	C.4	MATI	LAB code	114
D	hlia	monher		116
$\mathbf{D}$	ເນເມດຊ	grapny		110

## List of Tables

4.1	Summary of classification and estimation methods	30
4.2	Confusion matrices for simulated targets at SNR of 30 DB $\ldots$	34
4.3	Confusion matries summarizing the classification of several UXO targets from	
	the ERDC test stand data set. Classification was performed using a leave-one-	
	out method	35
7.1	Confusion matrices in presence of sensor position uncertainty at SNR of 30 dB $$	91
7.2	Confusion matrices in presence of sensor position uncertainty at SNR of 30 dB $$	
	using Min-Max approach	91
B.1	A demonstration of the efficiency of the particle swarm optimization algorithm	
	as compared to a simple parameter space search $\hdots$	106
B.2	Estimated poles for 105 mm, 81 mm, 60 mm, and 40 mm UXO items at various	
	orientations. The poles are estimated from EM63 data collected on a grid	
	centered on the target	109
B.3	Mean pole values estimated from data collected at the ERDC test stand	109

## List of Figures

2.1	Sensor configuration	14
2.2	Pole library for a cylindrical steel object	15
2.3	Pole library for a 81mm mortar shell	16
3.1	Effect of whitening transformation on density estimation	19
3.2	Feature space for cylindrical steel object	20
3.3	Contours corresponding to slice cuts across $x$ -axis poles for pole cloud in Fig. 3.2	21
3.4	Feature space for 81mm mortar shell object	22
3.5	Contours corresponding to slices across the $45^{\circ}$ plane of x and y-axis poles for	
	pole cloud in Fig. 3.4	22
4.1	Pole distributions for S1 (blue), S2 (red), A1 (black) and A2(green) objects.	
	See text for a physical description of targets	31
4.2	Probability of error as a function of SNR for feature based classifiers	32
4.3	Probability of error as a function of SNR for signal based classifiers $\ldots$ .	33
4.4	Poles distributions for BDU-26 (×), BDU-28 (o), 90mm (diamond) and 155mm	
	(star) UXO items	36
4.5	Noise histogram estimated using spatial and temporal residuals $\ldots \ldots \ldots$	37
4.6	Noise covariance matrix estimated using residual snapshots over spatial domain	38
4.7	spectral structure of estimated noise covariance matrix $\ldots \ldots \ldots \ldots \ldots$	38
4.8	Effect of bandwidth on performance of KDE classifier for ERDC data sets	40
5.1		44
5.2	Effect of unknown geometry on CRLB as a function of sampling and noise	
	variance for $x$ axis pole $\ldots$	47
5.3	Effect of unknown geometry on CRLB as a function of sampling and noise	
	variance for $y$ axis pole $\ldots$	48

5.4	Effect of unknown geometry on CRLB as a function of sampling and noise	
	variance for $z$ axis pole	49
5.5		50
5.6	Error probability upper bound as a function of signal to noise ratio for distin-	
	guishing between a luminum and a steel objects using GEM3 sensor $\ . \ . \ . \ .$	54
5.7	Error probability upper bound for distinguishing between $81 \text{ mm}$ and a $60 \text{ mm}$	
	object using EM63 sensor on a $5 \times 5$ sampling grid $\ldots \ldots \ldots \ldots \ldots \ldots$	55
5.8	Error probability upper bound for distinguishing between $81 \text{ mm}$ and a $60 \text{ mm}$	
	object using EM63 sensor on a $15 \times 15$ sampling grid $\ldots \ldots \ldots \ldots \ldots \ldots$	55
5.9	Error probability upper bound and SNR for distinguishing between $105 \text{ mm}, 81$	
	mm, 60 mm and 40 mm objects using EM63 sensor as a function of number of	
	spatial samples	56
5.10	Error probability upper bound and SNR for distinguishing between $81 \text{ mm}$ and	
	a 60 mm object using EM63 sensor as a function of number of spatial samples	57
5.11	Pole distributions for S1 and S2 steel objects at different SNR $\ldots$	60
5.12	Error probability upper bound for feature space classifier $\ldots \ldots \ldots \ldots$	61
5.13	Pole clouds of combined UXO targets and uniform clutter distribution bounds	63
5.14	Error probability upper bound for UXO vs. clutter for uniform clutter PDF $$ .	64
6.1	Sensor spatial sampling configuration	70
6.2	Collected data as the sensor size changes for 105mm mortar shell $\ldots \ldots \ldots$	71
6.3	Collected data as the sensor size changes for 81mm mortar shell $\ldots \ldots \ldots$	72
6.4	Collected data as the sensor size changes for 60mm mortar shell $\ldots \ldots \ldots$	73
6.5	Collected data as the sensor size changes for 40mm mortar shell $\ldots \ldots \ldots$	74
6.6	Error upper bound as the sensor size increases	75
6.7	Error upper bound as a function of sensor dimensions for four mortar shell objects	75
6.8	Error upper bound as a function of sensor dimensions $\ldots \ldots \ldots \ldots \ldots \ldots$	76
6.9	Performance upper bound as a function of sensor dimensions considering sensor	
	cost	76
6.10	Error upper bound as a sensing dimension	77
6.11	Error upper bound as a sensing dimension	77
6.12	Signal view as the sensor sampling space grows larger	79
6.13	Mortar shell objects configuration	80
6.14	Optimum sampling scheme for system configuration in Figure 6.13	81

6.15	Optimum sampling scheme for system configuration in Figure 6.13	82
7.1	Random walk/Markov chain model for sensor position uncertainty	86
7.2	Imposing neighborhood condition	87
7.3	2D random walk with equal probability neighborhood condition $\ldots \ldots \ldots$	87
7.4	2D random walk with high correlation with previous position	88
7.5	Nominal and perturbed sensor grid $\ldots$	88
7.6	Sensor sampling path	89
B.1	A comparison of a raw Geonics EM-63 dataset (left) collected over an 81mm	
	mortar shell at the ERDC UXO/Countermine test stand with a preprocessed	
	dataset (right).	103
B.2	Time domain data collected over a 40 mm mortar shell using the Geonics EM63	
	(dashed line) is compared to data simulated from estimated poles (solid line).	
	The curves are plotted for a profile from $y = 0m$ to $y = 4m$ , where $x = 0m$	
	and $z = -0.58m$ . Two target orientations are represented: (a) horizontal nose	
	north, and (b) vertical nose down	107
B.3	Time domain data collected over a 60 mm mortar shell using the Geonics EM63	
	(dashed line) is compared to data simulated from estimated poles (solid line).	
	The curves are plotted for a profile from $y = 0m$ to $y = 4m$ , where $x = 0m$	
	and $z = -0.58m$ . Two target orientations are represented: (a) horizontal nose	
	north, and (b) vertical nose up	107
B.4	Time domain data collected over a 81 mm mortar shell using the Geonics EM63	
	(dashed line) is compared to data simulated from estimated poles (solid line).	
	The curves are plotted for a profile from $y = 0m$ to $y = 4m$ , where $x = 0m$	
	and $z = -0.58m$ . Two target orientations are represented: (a) horizontal nose	
	north, and (b) vertical nose up	108
B.5	Time domain data collected over a 105 mm mortar shell using the Geonics EM63	
	(dashed line) is compared to data simulated from estimated (solid line) poles.	
	The curves are plotted for a profile from $y = 0m$ to $y = 4m$ , where $x = 0m$	
	and $z = -0.58m$ . Two target orientations are represented: (a) horizontal nose	
B.6	north, and (b) vertical nose down	108
~		

represents a different target orientation, target instance, or target burial depth. 110

C.1	Hidden Markov model	•	•		•	•	•	•		•	•	•	•	•	•	•	•			•	•		•	112
C.2	Viterbi algorithm		•					•							•									113

Where is the life we have lost in living? Where is the wisdom we have lost in knowledge? Where is the knowledge we have lost in information?

T.S. Eliot

# Chapter 1 Introduction

Unexploded ordnance (or UXOs/UXBs) are explosive weapons (bombs, shells, grenades, etc.) that did not explode when they were employed, and still pose a risk of detonation, decades after the battles in which they were used like American civil war, Vietnam war and Iraq war. According to US Environmental Protection Agency documents released in late 2002, UXO at 16,000 domestic inactive military ranges within the United States pose an "imminent and substantial" public health risk and could comprise the largest environmental cleanup ever, with a pricetag that begins at \$14 billion. Some individual ranges cover 500 square miles, and, taken together, the ranges comprise an area the size of Florida . In the Ardennes region of France, large-scale citizen evacuations were necessary during UXO removal operations in 2001<sup>1</sup>.

Considering these facts, UXO remediation poses a significant dilemma. In addition, costly, time consuming, and sometimes perilous excavations associated with false detection or classification of buried unexploded ordinance demand robust and powerful remote sensing processing algorithms. To this end, in the past decade UXO detection and classification methods have evolved significantly, both with the development of more sophisticated sensors as well as more accurate physical models and signal processing algorithms [1–3].

Because of metallic materials used in UXOs, most UXO discrimination problems are solved using model-based statistical processing methods employing electromagnetic induction (EMI) sensors. EMI sensors work based on scattering behavior of metallic materials when they are

<sup>&</sup>lt;sup>1</sup>"http://en.wikipedia.org/wiki/Unexploded\_ordinance"

excited by an electromagnetic field. The sensor in its simple form consists of a transmitter and receiver loop and generates an electromagnetic field by flowing current into transmitter loop and records the back scattered response of the object using a receiver loop. As explained before UXO remediation dilemma has resulted in more advanced sensors and processing algorithms. Indeed, current EMI sensors are capable of recording field response of targets with high spatial and temporal resolution. Physical models of the EMI sensors have been developed that enable us to model buried objects in terms of features that are relevant to classification. From these data and models, two classes of problems are typically considered: *Detection* and *Classification*. *Detection* of UXO can be considered as a binary hypothesis problem where the object is present or not, and was considered extensively in *e.g.* [4–6]. Our focus here is on the related *classification* problem in which we want to infer information about the type of UXO being investigated as well as *localization* of UXO. Localization gives spatial information about the buried UXO needed for safe excavation. In this thesis we will consider classification problem in conjunction with localization problem.

Most model-based methods for the classification of buried objects from electromagnetic induction sensor data are comprised of two steps. First, the data are used to estimate a set of parameters, some of which are of direct use in classification, and some of which are required to locate and orient the object relative to the sensor. These estimated parameters are then input to a classifier to determine the type of target under investigation. In many cases [2,7], underlying both of these processing steps is an implicit assumption that the objectspecific parameter values are independent of the orientation and location of the target. This assumption is in fact a consequence of using the so-called "dipole model" (or a derivative form of this model) as the basis for processing [8].

Under this model, the information concerning the target class is encoded in the  $3 \times 3$  magnetic polarizability tensor (MPT) whose three time-dependent eigenvalues consist of an infinite sum of decaying exponentials. The decay rates are the features of use in target classification. In theory, the veracity of this model requires illumination of an infinitesimally small target by a uniform field, a condition not generally met in practice as most real UXOs don't scatter

like dipoles. Moreover, given limited, noisy data, at most a single term in each of the three sums can be estimated reliably [8], leading some investigators to use forms of the model employing only two exponentials [8,9] on the assumption that targets of interest are spheroidal and therefore have two equal eigenvalues. These theoretical and practical limitations of the dipole model impart a position and orientation dependence on the object-specific parameter estimates required for accurate classification. Thus, analysis of multiple instances (i.e. positions, orientations, and target example) of a target class results in not a single parameter value for each object, but rather a cloud of values in parameter space.

To capture the dependency of the features on the location and orientation of the object, several methods have been proposed in the past. For example in [9], the authors used Laplaceplane poles located on the real-s axis for classification purposes. To account for the fact that the poles differed according to the orientation of the object in space, a single object was regarded as three different targets, one for each of three orthogonal orientations of the true UXO. Identification for arbitrary orientations as well as the issue of object localization was not considered in [9]. In [1], instead of a sum of decaying exponentials model, a frequency domain processing approach was considered. Here, the data at each frequency were used to estimate all nine elements of the MPT. An eigen-analysis of the estimated MPTs was used to determine a rotation matrix which implicitly described the orientation of the object in space as well as a collection of frequency-dependent features required for classification.

In [10,11], a processing algorithm was developed based on the fusion of the dipole scattering model and a parametric model for the eigenvalues of the MPT which are referred to here as *principle axis polarizability functions* (PAPFs). The model directly captures both the features for classification and the parameters required to infer the location and orientation of the buried object. The model uses a single pole per axis version of the PAPF expansion introduced in [8], which results in a three dimensional feature space; *i.e.* one pole per body axis. Because of this approximation the features are no longer independent of object orientation and location. This dependency is captured by building a library of poles for different orientations and locations. That is, each object in the library was represented in terms of a cloud of pole values. In terms of processing, a simple non-linear least squares method was used in [10, 11] to extract poles, location, and orientation information from the data. Using the estimated poles, classification was performed using the Mahalanobis distance metric. More specifically, for each object in the library this distance metric was calculated using the mean pole vector and covariance matrix for the associated pole cloud. In essence then, the work in [10, 11] was predicated on the implicit assumption that the pole cloud was Gaussian. As will be explained more in Chapter 2 this assumption is not true for our processing model and indeed we have found it to be true that such assumption is not in fact warranted in practice [7].

Here we consider new processing methods that account for this variability in ways that improve both the accuracy of the parameter estimates as well as the final classification. We interpret these pole clouds as prior probability density functions (PDF) over the object-specific parameters and use kernel density estimation techniques to determine these PDFs. The resulting prior models in parameter space are used to develop a new collection of estimation and classification algorithms. In terms of estimation, the priors we build allow for the use of Maximum *a posteriori* (MAP) methods for feature extraction rather than Maximum Likelihood techniques.

For classification, we consider schemes that function both in feature space (i.e. exclusively using the estimated parameters themselves) as well as data space (i.e. using these features along with a sensor model to simulate data and compute residuals). In the former case, we developed feature space MAP classifiers built around our KDE PDFs and a more generic support vector machine (SVM) classifier. For data-space classifiers, we have developed a pair of generalized likelihood ratio-type tests as well as approximations to the Bayesian optimal classifier. One of these methods makes use of the Laplacian approximation to the multidimensional integral encountered in the Bayes classifier. The second employs a new approach to reduce the dimensionality of the integral through judicious use of the estimated orientation and location parameters.

The performance of these methods is evaluated at different signal to noise ratios and compared in terms of complexity and probability of error. Results using both simulations and real field data indicate that the incorporation of these prior PDFs into the classification processing can substantially improve accuracy relative to previous mean-covariance classifiers.

Central to the continued success of these and related efforts is the issue of *performance* analysis; that is, understanding the limits of one's ability to successfully extract information from data provided by these sensors. The availability of tools for performance analysis can in principal impact the design and deployment of a sensor as well as provide us with a quantitative understanding of the impact of exogenous factors on our processing schemes. For example performance analysis methods can be used to provide for an understanding of the spatial sampling required to successfully classify buried objects. Alternatively, object classification is frequently complicated by the fact that the precise location of the buried target is not known and thus must be estimated as part of the processing. Performance analysis methods can be used to quantify the "information loss" associated with this uncertainty, thereby indicating e.g. whether one needs to deploy additional sensors that might be used to resolve these geometric parameters. One objective of this thesis is to develop a collection of analytical performance analysis methods and to use them to initially explore some of the issues raised in this paragraph.

In the UXO context as explained before, there are two relevant forms of performance analysis reflecting the fact that many algorithms require first the estimation of a collection of parameters and second the use of a subset of these parameters for purposes of classification and discrimination [12]. For example, the dipole scattering model [8] [13] and variants [14] are quite commonly employed in the UXO community as analytically tractable, phenomenological models that relate the observed data to properties of the buried object. These models are explicitly parameterized in terms of the location and orientation of the object in space as well as a collection of quantities (e.g., "poles,", "decay rates," "betas," etc. depending on the specific form of the model) that are in theory properties of the specific object under consideration and thus potentially useful for classification. Hence, when we discuss the issue of performance analysis for this class of processing schemes, we can think about metrics such as the accuracy with which we can estimate the parameters of the model or the rate of error in our classifiers. In this thesis we address both such possibilities.

The development and use of tools such as those we discuss here has a long history in the field of statistical signal processing in general and the UXO community in particular. Concentrating on the later, analysis of algorithm performance is commonly done in one of three ways. Using Monte-Carlo methods, data are simulated to explore the full range of randomness in the problem. Results are generated for each of the data sets and statistics are then collected to quantify e.g. the accuracy of parameter estimates, detection rates, false alarm probabilities, or error probabilities, depending on the study [7]. Such methods tend to be computationally intensive often requiring thousands (or more!) of simulations in order to obtain meaningful statistics on the performance metrics.

Alternatively, a second form of performance analysis is based on a probabilistic formulation or interpretation of the processing problem. Determination of parameter values from noisy data is a statistical estimation problem. Discrimination and classification may be cast in terms of statistical decision theory. When viewed in this context, one can make use of analytical bounds [15] [16] [17] well known in the statistics literature to place limits on the performance of either estimation methods or classification techniques. Relative to Monte-Carlo methods, these bounds tend to be easier to evaluate computationally and can be used to explore all the design and analysis issues relevant to the UXO problem.

The use of bounds is not without its drawbacks. First, they *are* bounds and as such may be overly pessimistic or optimistic in their predictions. Hence in this thesis, we supplement our bound calculations with Monte-Carlo results to demonstrate that within the context of the problem at hand, the bounds do provide a usable level of accuracy. A second potential difficulty with the use of bounds for performance analysis is the fact that these quantities are all computed based on an underlying model relating the data to the unknown parameters. Thus, the accuracy of the predictions of the bound are clearly dependent on the accuracy of the model.

More relevant to this thesis is the fact that here we restrict our attention to the use of a variant of the dipole model which our group [12]- [10] as well as a wide range of other researchers [6] [9] have employed previously. Despite the acknowledged shortcomings of the dipole models we feel that its use here is appropriate because many of the bounding methods discussed here are new to the community and hence represent a contribution regardless of the exact model used in their evaluation. That is, we provide here a methodological framework into which other interested researchers and practitioners can "plug in" their favorite EMI sensing models. Second, the analytical simplicity of the model makes it well suited for use as an example for calculation of the bounds. Additionally, since this model *is* so widely used, we hope that better understanding the inherent performance characteristics of estimation and classification methods built using the model will make the analysis in Chapter 5 relevant and interesting.

As we discussed in greater depth in Chapter 2, we shall be exploring the performance characteristics of algorithms that employ not only the dipole model, but also prior statistical models for the classification features [12] [18] so this performance analysis will also serve as a complementary analysis to those methods. These prior models are designed to account for inaccuracies in the dipole model and hence here we hope improve the accuracy of our analysis in Chapter 5  $^{2}$ . The analysis methods we explore here are motivated by and build upon previous efforts of our group in the area of UXO classification from EMI data. It is important again to clearly distinguish this problem from the related one of detection. *Detection* of UXOs can be considered as a binary hypothesis problem where the object is present or not and was considered extensively [4–6]. As will be explained, by proposing classification methods using non-parametric prior models [12,18] we addressed the problem of *classification* and *localization* in which we want to infer information about the type of the object as well as the spatial position and orientation of the object. The idea behind this work was to explore methods in for which one might be willing to carefully collect data more densely sampled in space than is common in wide area surveillance in order to obtain finer grain information about a buried object suspected of being a UXO prior to removal.

<sup>&</sup>lt;sup>2</sup>Validating this assertion would require a Monte Carlo analysis that employed a computational model for the UXO problem based on e.g. a finite element solution to the underlying Maxwell's equations. Such a model is not at our disposal.

Considering performance analysis, we develop and analyze performance bounds for both the estimation as well as classification schemes associated with the above ideas. For estimation performance Cramer-Rao lower bounds (CRLB) on the variance of the estimated parameters are derived using both deterministic and Bayesian versions of CRLB. Monte-Carlo analysis is employed to verify the tightness of these bounds. For classification performance the Chernoff upper bound on the probability of error for an optimal Bayesian classifier is derived analytically using the Laplacian and asymptotic approximation in order to overcome difficulties with high dimensional integrals involved in the bound. The Chernoff upper bound is also evaluated for optimal feature based classifiers, which involves integration in three dimensional space, and is evaluated analytically and verified using Monte-Carlo simulation. These optimal classifiers were chosen because of the analytical versatility of these classifiers which enables us to derive analytical bounds in contrast to other ad-hoc methods.

As explained before, these performance tools can be used both in analytical description of the sensor system as well as synthetic interpretation, depending which is of use to the system analyst or developer. In this work we concentrate more on analytical description of the system and show paths toward the design of systems that are tuned for best performance in terms of these bounds using the theory of design of experiments [19]. Here again we assume that an initial wide-area assessment of a region of interest has identified a few regions in which one suspects with high probability the presence of a UXO item. The cued identification problem then involves the deployment of sensors (in our case EMI) to collect data on a fine scale spatial grid for purposes of classifying the buried object and perhaps characterizing, with greater precision than could be obtained from the wide area assessment, the location and orientation of the buried object. Using the Cramer-Rao and Chernoff methods discussed previously we examine for a hypothetical but realistic sensing problem the impact of noise and spatial sampling density on both the accuracy of our decay estimates as well as the error rates of optimal classifiers. We are also able to examine quantitatively the impact of unknown location and orientation of the object on our decay rate estimates, thereby providing an indication as to the value of deploying independent sensors such as magnetometers capable

of resolving this geometric information [2].

The last part of this work deals with developing classification algorithms when there is uncertainty in sensor position, which is a relevant problem in field data processing where the sensor position is controlled by a imprecise instrument. To this end we have developed an uncertainty position model using a Markov chain/ random walk model which captures spatial dependency of error locations, and developed new robust estimation algorithms with low computational cost using a min-max approach.

The thesis is organized as follows: In Chapter 2 physical models and basic processing models are introduced. In Chapter 3 prior models are discussed. Estimation and classification algorithms are introduced in Chapter 4. In Chapter 5 performance analysis for estimation and classification methods are discussed. In Section 6 we will introduce methods for sensor system optimization utilizing the tools developed in Chapter 4. In Section 7 sensor processing in the context of position uncertainty is investigated. Chapter 8 contains our conclusions and suggestions for future work.

# Chapter 2 Physical Model and Prior Work

In this chapter a summary of the physical model used in our processing algorithms and in previous processing methods, and its drawback are discussed. As explained in the introduction the dipole model is used to model EMI scattering behavior of UXOs for frequency and time domain sensors. Section 2.1 deals with the dipole physical model and Section 2.2 reviews basic processing methods employed in [10]. A more accurate description of the physical models used in our processing is provided in Appendix A.

#### 2.1 Dipole Model

The processing model used here is based on [10], which is a truncated version of the EMI physical model in [8] and [13]. Using this model, the scattered signal collected at M time gates or frequencies (depending on the sensor) at each of L locations in space can be written as:

$$y_{l,m} = H_{R,l}^{T}(r_{0})R^{T}(\alpha)\Lambda_{m}(p,c,\delta)R(\alpha)H_{X,l}(r_{0}) + n_{l,m}$$

$$l = 1...L, m = 1...M$$
(2.1.1)

where:

- .  $y_{l,m}$  is the received signal at location l and frequency or time sample m.
- .  $H_{X,l}$  and  $H_{R,l}$  are  $3 \times 1$  vectors holding the x, y, and z components of the transmitted field and the hypothetical field of the receiver [20]. Both of these vectors are functions of the  $3 \times 1$  vector  $r_0$  containing the x, y and z locations of the UXO item.

- .  $n_{l,m}$  is additive Gaussian noise at location l and frequency/time m.
- .  $R(\alpha)$  is 3 × 3 rotation matrix orienting the object in space and dependent on a 3 × 1 vector,  $\alpha$ , of Euler angles.
- . p is  $3 \times 1$  pole vector for frequency domain sensors or time constants for time domain sensors, coefficient factors and DC value (for frequency domain sensor).
- . c is  $3\times 1$  coefficient vector.
- .  $\delta$  is DC value for frequency domain sensors.

In general the matrix  $\Lambda_m(p,c,\delta)$  is a 3 × 3 diagonal matrix taking the form

$$\Lambda_m(p,c,\delta) = \begin{pmatrix} \lambda_{m,1}(p,c,\delta) & 0 & 0\\ 0 & \lambda_{m,2}(p,c,\delta) & 0\\ 0 & 0 & \lambda_{m,3}(p,c,\delta) \end{pmatrix}.$$
 (2.1.2)

Each of diagonal elements of  $\Lambda_m$ ,  $\lambda_{m,i(i=1..3)}$ , is a principle axis polarizability function (PAPF) which contains orientation-invariant spectral signatures and can be written as infinite sum of one pole transfer functions in the frequency domain or sum of exponentials in the time domain:

$$\lambda_{m,i}(p,c,\delta) = \begin{cases} \delta_i + \sum_{\substack{l=0\\\infty}}^{\infty} \frac{c_{i,l} \, jf_m}{p_{i,l} + jf_m} & (Frequency) \\ \delta_i \delta(t_m) + \sum_{\substack{l=0\\n=0}}^{\infty} -p_{i,l}c_{i,l}e^{-p_{i,l}t_m} & (Time). \end{cases}$$
(2.1.3)

Going forward, we refer to the parameters  $p_i$  in (2.1.3) as "poles" in recognition of the fact that they are the roots of the transfer functions in the frequency-domain form of the  $\lambda_{m,i}$ . Combining the data from all spatial locations and time or frequency samples gives the model:

$$y(\theta) = [y_1^T(\theta) \quad y_2^T(\theta) \quad \cdots \quad y_M^T(\theta)]^T$$

$$y_i(\theta) = m_i(\theta) + n_i$$
(2.1.4)

where:

.  $\theta = [p^T, \alpha^T, r_0^T, c^T, \delta^T]^T$  with *e.g.* p the vector whose *i*-th element is  $p_i$  from (2.1.3).

- .  $y_i: L \times 1$  received vector signal at frequency/time m.
- .  $m_i$ :  $L \times 1$  vector model in 2.1.1.
- .  $n_i$ :  $L \times 1$  vector of Gaussian noise which is taken to have zero mean and covariance matrix C independent of i.

As the PAPF theoretically contains parameters p which are spatially invariant with respect to the object, these parameters can be used as features for classification purpose. As there are infinite number of poles for a generic object, it is impossible to estimate all the features so we use a truncated version of dipole model where the feature vector to be used for classification is comprised of three poles, one for each of the principal axes of the target. That is, only the l = 0 terms in the summations of (2.1.3) are retained. It then proves convenient for us to partition the parameter vector  $\theta$  into features (*i.e.*, the poles) and nuisance parameters where the former are relevant to classification and the latter consist of location, orientation, coefficient factors and the DC parameter, which are used to infer the geometry of the object rather than its type. The parameter vector will be denoted as  $\theta = [p^T, \nu^T]^T$ , where p is the  $3 \times 1$  vector of poles and  $\nu$  is the vector of the nuisance parameters.

### 2.2 Basic Processing

As noted previously, the approximate nature of this model (where only one term is used in the PAPF expansion) will in fact impart a dependence of these parameters on the location and orientation of the object relative to the sensor. In order to account for the dependency of features on object location and orientation in the processing, our work makes use of a "pole library" for each object. This library is comprised of the estimated values of the poles as a function of object position and orientation. Given either measured or simulated data for an object in one of N possible positions/orientations, we build samples from this library via a maximum likelihood approach <sup>1</sup>:

$$\hat{p}_{ML,j} = \arg\min_{p,\nu} \|C^{-1/2}(y(\alpha_j, r_{0_j}) - m(p,\nu))\|_2^2$$
(2.2.1)

for j = 1, 2, ..., N. Figure 2.2 shows an example of a library (feature space) built for a three inch long by one inch diameter stainless steel cylinder corresponding to five possible object depths, seven possible values for each of the three Euler angles and no horizontal variation of target using frequency domain sensor in [10]. The sensor has two  $1 \times 1(m)$  square coils for transmitter and receiver and samples on a one meter square area using an equally spaced  $5 \times 5$ grid of measurement points which is shown schematically in Figure 2.1. Data were generated using a four pole per axis dipole model in which the sums in (2.1.3) are terminated after four terms [8,10] using ML estimation we try to match the data to the one pole per axis model. (More details on implementation of the estimation algorithm is explained in Section 4.1). Figure 2.3 shows the library for a 81mm mortar shell using the EM63 time domain sensor. The sensor has two  $1 \times 1(m)$  transmitter and receiver square coils and samples on a  $2 \times 4(m)$ rectangular area using an  $9 \times 31$  grid of measurement points. The EM63 sensor samples 26 data points in time from 0.1770 (msec) to 25 (msec) in log-spaced intervals. Because for this target as opposed to cylinder targets we do not have an exact analytical formulation for four pole model, here we used a four pole quasi-real model to simulate the target where we used one pole per axis parameters estimated from the field data (see Appendix B to see how poles

<sup>&</sup>lt;sup>1</sup>See § 4 for a discussion as to how the optimization problem is solved



Figure 2.1: Sensor configuration

for were estimated from real field data) and then we build a four pole model by generating four equally spaced poles within the range of .5 and 1.5 times of the corresponding estimated poles for each axis.

In the previous work [10], classification was done in feature space using a pole library in a fairly simplistic manner. Assuming we had a library of i = 1, 2, ..., N objects, a pole cloud like the one in Fig. 2.2 was generated for each target. For each cloud the mean pole vector  $\bar{p}_i$  and the associated covariance matrix  $C_i$  was computed. Given a data set containing an object to be classified, we first estimated the best fit model parameters using a maximum likelihood scheme much like (2.2.1) with  $y(\alpha_j, r_{0,j})$  replaced by data containing an object of unknown structure:

$$\hat{p}_i = \arg\min_{\substack{p \in object_i, \nu}} \|C^{-1/2}(y - m(p, \nu))\|_2^2$$
(2.2.2)

Classification was then done using a Mahalanobis distance classifier based on the statistics of the different pole clouds [21]:

$$\hat{i} = \arg\min_{i} \left( \hat{p}_i - \overline{p}_i \right)^T C_i^{-1} (\hat{p}_i - \overline{p}_i)$$
(2.2.3)

where  $\hat{p}_i$  is the ML pole estimate from (2.2.2).

Figures 2.2 and 2.3 show that the shape of the pole clouds is far from an ellipsoid, thereby invalidating the Gaussian assumption used in [10]. This leads us to believe that a more accurate model of this distribution could improve the overall performance of the processing. Substantiation of this claim and exploitation of the resulting structure to improve classification as well as performance analysis of the propose algorithms are the essential contributions



Figure 2.2: Pole library for a cylindrical steel object

of this thesis. In the next sections, we will apply nonparametric probability distribution estimation to build a more accurate representation of the pole distribution. These nonparametric distributions therefore represent enhanced prior models for the pole clouds associated with each object. They also provide an analytic representation of the pole clouds which enables us to derive performance bounds and to get insight into how much each parameter is the system influences system performance.

Using this prior information, the processing scheme can be set into a Bayesian framework both for estimation and classification, where ML estimation is replace by MAP estimation, which is known to be more robust to noise (justification of this claim will be given in Chapter 5). For classification these priors enable us to implement optimal minimum error classifiers and approximations of them. As we will see in Chapter 4 the results show significant improvement over the simple Mahalanobis approach, especially for Bayesian classifiers at low signal to noise ratios.

In the following chapter we will introduce the prior model we have used to capture an



Figure 2.3: Pole library for a 81mm mortar shell

analytical representation of the features in the feature space.

# Chapter 3 Feature Prior Model

As explained in the previous chapters, and as will be explored in this chapter, the assumption of a Gaussian distribution over features is not valid. Therefore we need to have an accurate and analytic distribution of the features to be able to develop more powerful classification algorithms and also to be able to derive analytical performance bounds on the system. In this chapter we review the kernel density estimation method, a powerful non-parametric density estimator, and its application to UXO discrimination.

### 3.1 Kernel Density Estimation

The assumption of parametrically defined probability distributions for inference and hypothesis testing fails in many practical engineering problems; therefore techniques which make less rigid assumptions about distribution of data are required. Nonparametric estimation schemes have been considered for this purpose [22–24]. Given a set of N p-dimensional data samples  $\{x_n, n = 1...N\}$  of an unknown distribution, the problem is to estimate the probability distribution in some flexible manner. Kernel methods proceed by locating a function  $\phi$  with "width" parameter h at each observed datum to generate the estimated density  $\hat{f}(x)$  as [22]:

$$\hat{f}(x) = \frac{1}{Nh^p} \sum_{n=1}^{N} \phi(\frac{1}{h}(x - x_n)).$$
(3.1.1)

The kernel function  $\phi$  is chosen to satisfy:

$$\phi(x) \ge 0 \text{ and } \int_{R^p} \phi(x) dx = 1.$$
 (3.1.2)

Here, we use Gaussian kernels which take the form:

$$\phi(x) = (2\pi)^{-\frac{p}{2}} \exp(-\frac{1}{2}x^T x).$$
(3.1.3)

The optimal kernel width, h, is determined through the minimization of mean integrated square error. More complicated methods for determining the optimal kernel band width are use likelihood cross validation, expectation maximization (EM) and graph test approaches [22].

#### 3.1.1 Whitening Transformation

In order to obtain a probability distribution over feature space, the pole estimates comprising the cloud are considered as samples of an unknown probability distribution function, and kernel density estimation is used to estimate the corresponding PDF. As the poles are not equally spread in all directions, the data are first whitened by a linear transformation, yielding data with zero mean and unit covariance matrix. Then density estimation is applied to transformed data. This method was proposed in [25], and helps provide a better approximation of optimal bandwidth. Given observed data x, the whitened data z is written as:

$$z = S^{-\frac{1}{2}}(x - Ex), \tag{3.1.4}$$

where Ex is the sample mean and S is sample covariance:

$$S = E[(x - Ex)(x - Ex)^{T}].$$
(3.1.5)

The resulting kernel density estimation for the original data can be written as:

$$\hat{f}(x) = \frac{\det(S)^{-\frac{1}{2}}}{Nh^p} \sum_{n=1}^{N} \phi(\frac{1}{h}S^{-\frac{1}{2}}(x-x_n))$$
(3.1.6)

For our estimation, the Gaussian kernel with band width:

$$h = AN^{-\frac{1}{p+4}}$$
 where  $A = \left[\frac{4}{2p+1}\right]^{\frac{1}{p+4}}$  (3.1.7)

is used, which provides a good approximation for the whitened data [23].

Figure 3.1 shows results of density estimation for a two dimensional Gaussian distribution. For the case when the whitening (data sphering) approach is used it is clear that whitening preprocessing results in more accurate density estimation for the case of non spherical distributed sample clouds.



Figure 3.1: Effect of whitening transformation on density estimation

To investigate the distribution of the feature library, here we applied kernel density estimation to the feature space corresponding to a three inch long by one inch diameter stainless steel cylinder as shown in Figure 3.2, using the frequency domain sensor in [10]. The sensor has two  $1 \times 1(m)$  square coils for transmitter and receiver and samples on a one meter square area using an equally spaced  $5\times5$  grid of measurement points. In order to visualize the resulting density, slice cuts along the x axis were made corresponding to the x-pole axis,  $p_x = 3500, 3720, 3940, 4160, 4380, 4600(Hertz)$ . The contour plots shown in Figure 3.3 show the estimated PDF on each slice. As another example Figure 3.5 shows the slice plot corresponding to the pole cloud in Figure 3.4 for a 81mm mortar shell using the EM63 time domain sensor. The sensor has two  $1 \times 1(m)$  square coils for transmitter and receiver and samples on a  $2 \times 4(m)$  rectangular area using an  $9 \times 31$  grid of measurement points. From these figures it is obvious that the Gaussian assumption on pole clouds is not valid and kernel density estimation captures this non-Gaussian distribution very well. Now that we have an accurate analytical



Figure 3.2: Feature space for cylindrical steel object

representation of the PDF of features, we can use these PDF's as the prior probability distribution for our processing. As will be explained in the following chapter, these priors will be used in a Bayesian framework for a new suite of estimation and classification algorithms. As we will see the incorporation of priors into processing gives significant improvement over conventional methods, specially at low signal to noise ratios, and results in more robust classification.



Figure 3.3: Contours corresponding to slice cuts across x-axis poles for pole cloud in Fig. 3.2


Figure 3.4: Feature space for 81mm mortar shell object



Figure 3.5: Contours corresponding to slices across the  $45^{\circ}$  plane of x and y-axis poles for pole cloud in Fig. 3.4

# Chapter 4 Estimation and Classification Methods

In this chapter, we make use of the non-parametrically constructed PDFs discussed in the previous chapter to develop new parameter estimation algorithms as well as a collection of classification schemes. While the details of all methods are discussed at length in this Section, the reader is referred to Table 4.1 for a summary of the different techniques. Finally performance of classification algorithms are evaluated using Monte Carlo simulations and real field data sets in Section 4.3.

# 4.1 Estimation Methods

As explained in Chapter 2 the conventional estimation method for extracting feature from the model is maximum likelihood, which can be written as:

$$\hat{p}_i = \arg\min_{\substack{p \in object_i, \nu}} \|C^{-1/2}(y - m(p, \nu))\|_2^2.$$
(4.1.1)

Using the PDF constructed over features, we can incorporate this information into estimation routine where the maximum likelihood estimation used in (4.1.1) is replaced by MAP estimation for features using our nonparametric priors:

$$\hat{p}_i, \hat{\nu}_i = \operatorname*{arg\,max}_{p,\nu} f_i(y/p,\nu) f_i(p)$$
(4.1.2)

where  $f_i(p)$  is the PDF of the poles corresponding to *ith* object. Given the assumption of Gaussian noise. Equation (4.1.2) is equivalent to:

$$\hat{p}_i, \hat{\nu}_i = \arg\min_{p,\nu} \|C^{-1/2}(y - m(p,\nu))\|_2^2 - \ln f_i(p).$$
(4.1.3)

Since  $-\ln f_i(p) \ge 0$ , we can write (4.1.3) as:

$$\hat{p}_i, \hat{\nu}_i = \arg\min_{p,\nu} \|C^{-1/2}(y - m(p,\nu))\|_2^2 + (\sqrt{-\ln f_i(p)})^2.$$
(4.1.4)

Thus, the optimization problem can be implemented using nonlinear least squares estimation. We have used the Levenberg Marquardt (LM) algorithm [26]. Gradients were computed using a finite difference approximation and a simple one dimensional search was used to find the regularization parameter in the LM method [26]. A coarse grid search in parameter space was employed to find good initial values for the optimization methods.

# 4.2 Classification Methods

## 4.2.1 Optimal classification

It is known that the classification rule which minimizes the probability of error is the Bayes decision rule, [21]:

$$\hat{i} = \arg\max_{i} f(H_i/y) \tag{4.2.1}$$

where  $f(H_i/y)$  is the posterior probability distribution of *ith* hypothesis. Using the Bayes rule, (4.2.1) can be written as:

$$\hat{i} = \arg\max_{i} \frac{f(y/H_i)f(H_i)}{f(y)}$$
(4.2.2)

where  $f(H_i)$  is the prior distribution of each hypothesis, and is assumed to be uniform. Expanding  $f(y/H_i)$  in terms of priors using the Bayesian rule and ignoring the constant denominator, the optimal Bayes classifier can be written as:

$$\hat{i} = \arg\max_{i} \int f(y/H_i, \theta) f_i(\theta) d\theta.$$
(4.2.3)

where  $f_i(\theta) = f_i(p)f(\nu)$  is the prior probability distribution of parameters, assuming a uniform distribution for the nuisance parameters. The vector  $\theta = [p^T, \alpha^T, r^T, c^T, \delta^T]^T = [p^T, \nu^T]^T$ has 13 parameters which makes the integration in (4.2.3) almost impossible numerically. Although Monte Carlo methods can be used to evaluate (4.2.3), they are quite time consuming. Therefore, we investigate other solutions in this thesis. As can be seen from (4.2.3), the optimal classifier operates on the entire signal y. It is known that if the model parameters (such as the poles in our case) are *sufficient statistics* [27] then optimal classification can be done in a lower dimensional feature space. Unfortunately, for the UXO problem poles are *not* sufficient statistics [7] and therefore do not lead to optimal classifiers. Despite this theoretical shortcoming, these parameters still enjoy wide use as the basis for solving the UXO classification problem, due in no small part to convenience and to the success of such methods in practice, where other issues (such as the inaccuracies of the underlying dipole model) become important.

Based on the above discussion we see that classifiers are divided into two groups: signal based or feature-based, where the former employs the entire received signal for classification and in latter only makes use of the feature parameters. Both approaches will be considered in this section.

# 4.2.2 Signal Space Classifiers GLRT

The generalized likelihood ratio test (GLRT) classifier [17] approximates the optimal classifier in (4.2.3) using estimated values for the parameters. We have implemented two forms of the GLRT. The first is based on a maximum likelihood approach both for estimation of the parameters as well as classification. We refer to this method as the ML-GLRT and it takes the form

$$\hat{i} = \arg\max_{i} f(y/H_i, \hat{\theta}) \tag{4.2.4}$$

$$\hat{\theta} = \arg\min_{\theta} \|C^{-\frac{1}{2}}(y - m(\theta))\|_{2}^{2}.$$
(4.2.5)

The second approach uses the prior models developed in Chapter 3 both for purposes of estimation as well as classification. As such, it is a maximum *a posterioiri* technique and is referred to as the MAP-GLRT:

$$\hat{i} = \arg \max_{i} f(y/H_{i}, \hat{p}_{i}, \hat{\nu}_{i}) f_{i}(\hat{p}_{i})$$
(4.2.6)
where  $\hat{p}_{i}, \hat{\nu}_{i} = \arg \min_{p,\nu} \|C^{-\frac{1}{2}}(y - m(p,\nu))\|_{2}^{2} - \ln f_{i}(p)$ 

Applying natural logarithm to both sides of first equation in (4.2.6) and assuming the noise is Gaussian, we have:

$$\hat{i} = \arg\min_{i} \left\{ \|C^{-\frac{1}{2}}(y - m(\hat{\theta}))\|_{2}^{2} - \ln f_{i}(\hat{p}_{i}) \right\}$$
(4.2.7)

We note that the first term can be interpreted as the residual and extends the feature based kernel classifier (see Section 4.2.3) to a signal space classifier.

#### Hybrid Classifier

As mentioned in § 4.2.1, evaluation of the optimal MAP classifier in (4.2.3) is difficult because of the number of variables of integration. However, the only parameters relevant to classification are the three poles. The idea here is to use the ML estimates of nuisance parameters in (4.2.3)and to evaluate the integral only over the pole parameters. Mathematically, then our hybrid classifier is:

$$\hat{i} = \arg\max_{i} \int f(y/H_i, p, \hat{\nu}) f_i(p) dp$$
(4.2.8)

where  $\hat{\nu} = [\hat{\alpha}^T, \hat{r}^T, \hat{c}^T, \hat{\delta}^T]^T$  consists of the estimated orientation, location, coefficient factors and DC parameters. Integration is over three dimensional feature space (poles) and is evaluated easily numerically using quadratic basis functions.

#### Laplacian approximated classifier (LA-GLRT)

The Laplacian approximation [28] is used to approximate the posterior distribution in (4.2.3). It is based on the asymptotic Gaussian approximation of the likelihood in (4.2.3), and can be written as:

$$f(H_i/r) \propto \int f(y/H_i, \theta) f_i(\theta) d\theta \approx \frac{(2\pi)^{q/2}}{\sqrt{\det(J)}} f_i(\hat{\theta}) f(y/\hat{\theta}, H_i)$$
(4.2.9)

where q is the number of parameters,  $\hat{\theta}$  is ML or MAP estimation of parameter  $\theta$  and J is the observed Fisher information matrix (OFIM) [29] defined as:

$$J_{ij} = -\frac{\partial^2}{\partial \theta_i \partial \theta_j} \ln p(y/\theta)|_{\theta=\hat{\theta}}.$$
(4.2.10)

Applying natural logarithm to both side of equation (4.2.9), we have a classifier based on the Laplacian approximation which can be written as:

$$\hat{i} = \arg\max_{i} \left\{ -\frac{1}{2} \ln(\det(J)) + \ln f_i(\hat{p}) + \ln f(y/\hat{\theta}, H_i) \right\}.$$
(4.2.11)

This is the same as the MAP-GLRT classifier with the exception of the first term on the right hand side of (4.2.11), which is related to the accuracy of the underlying parameter estimates. More specifically, for the case where the noise is taken as additive and Gaussian, the OFIM can be interpreted as an approximation to the Hessian of the residual function,  $||C^{-1/2}(y-m(\hat{\theta}))||_2^2$ , evaluated at the MAP estimate of the parameters. Thus for example, when the parameter estimates are quite accurate, the Hessian would be "small" and the first term on the right hand side of (4.2.11) large, thereby adding to the evidence that the corresponding hypothesis is correct. Compared to the hybrid classifier, this method is faster as it does not require any integration and, as we shall see in § 4.3, it has the same performance for moderate and high SNR as the hybrid classifier.

# 4.2.3 Feature Space Classifiers Kernel classifier (KDE)

As an approximation of the optimal feature space classifier, the kernel classifier can be written as :

$$\hat{i} = \arg\max_{i} f_i(\hat{p}_i) \tag{4.2.12}$$

where  $f_i$  is the PDF of the poles of the *ith* object and  $\hat{p}_i$  is the ML or MAP estimated vector of poles assuming the *ith* object is the true object. In [10], a feature space classifier which was based on the Gaussian assumption on pole distributions was used where the poles were estimated using ML estimation. The method reduced to the Mahalanobis classifier in (2.2.3). Using the kernel density estimated prior (3.1.1), we can estimate the poles using the MAP estimator and implement (4.2.12) directly:

$$\hat{i} = \arg\max_{i} f_i(\hat{p}_i) \quad \text{where} \quad \hat{p}_i, \hat{\nu}_i = \arg\min_{p,\nu} \|C^{-\frac{1}{2}}(y - m(p,\nu))\|_2^2 - \ln f_i(p).$$
 (4.2.13)

Compared to the GLRT data space classifier, this classifier can be considered the GLRT "approximation" to the optimal feature based classifier.

#### Support Vector Machine

Support vector machines (SVM) are based on structured risk minimization, which was introduced in [30], and have been considered extensively recently. The SVM is a linear two category classifier which maximizes the margin of a two hypothesis space [31]. In [2], the authors used SVM for a UXO detection problem, a binary detection problem. Here we have applied SVM for multiclass UXO discrimination employing algorithms for extension of SVM to the multiclass case. Given a set of data points which belong to either of two classes, a SVM finds the optimal separating hyperplane that minimizes the risk of misclassifying the training samples and unseen test samples. Given a set of test points  $x_i$  and corresponding target class  $y_i$ , and assuming a linear classifier of the form  $x \cdot w + b$ , we will have:

$$y_i(x_i \cdot w + b) - 1 \ge 0, \ \forall i$$
 (4.2.14)

where w is the norm of hyperplane that divides the hypothesis space. To maximize the margin of the classifier, we need to minimize this norm, leading to the following problem:

min 
$$||w||_2^2$$
, subject to  $y_i(x_i \cdot w + b) - 1 \ge 0$ ,  $\forall i$ . (4.2.15)

The equivalent quadratic programming problem is to determine the minimum value of Lagrangian function:

$$L_p = \frac{1}{2}w.w - \sum_{i=1}^{N} \alpha_i (y_i(w \cdot x_i + b) - 1)$$
(4.2.16)

with  $L_p = L(w, b, \alpha)$ . The final classifier then can be explained in terms of support vectors:

$$f(x) = sgn(\sum_{s_i \in \text{support vectors}} \alpha_i y_i s_i x + b)$$
(4.2.17)

For non-separable test data, the solution is a generalized optimal separating hyperplane:

$$\min \frac{1}{2}w \cdot w + x \sum_{i=1}^{N} \xi_i \quad i = 1...N \quad \xi \ge 0$$
(4.2.18)

Data which are not linearly separable are mapped to a higher dimensional space, and kernel functions are used to lower the complexity of computation in this high dimensional space. Kernels operate on low dimensional space and give the same result as inner products in high dimensional space. Thus if the mapping is  $\phi(x)$  for kernel k we will have:

$$k(x_i, x_j) = \phi(x_i)\phi(x_j)$$
 (4.2.19)

There are different kernels that can be used, the most common of which are: -Simple dot product kernel:

$$k(x_i, x_j) = \phi(x_i) \cdot \phi(x_j) \tag{4.2.20}$$

-Radial basis function (RBF):

$$k(x,y) = \exp(-\|x-y\|^2/2\sigma^2)$$
(4.2.21)

-Two layer neural network:

$$k(x,y) = \tanh(kx \cdot y - \delta) \tag{4.2.22}$$

Though SVM was defined for the two class case, it can be used for the multiclass case. Several methods were proposed in [32], [33]. The most commonly used methods are one-against-all and one-against-one. In the former method the *ith* class is trained by labelling all the samples in the *ith* class as positive and the rest as negative. For a N-class problem, we will have N classifiers. The one-against-one method constructs N(N-1)/2 classifiers consisting of all pair wise combination of classes. To combine classifiers, different algorithms can be used, the most common of which is the 'max wins' algorithm, where the final decided class is the one with maximum votes [32]. There is another method proposed in [33], where graph theory is used to combine the classifiers. We have used the one-against-one algorithm, in which the poles were estimated using ML estimation and Gaussian kernels were used. Table 4.1 summarizes the classification methods and the corresponding estimation algorithms.

# 4.3 Classification Results

Classification performance was evaluated using simulated and real data. For simulation, the responses of four targets were simulated for a GEM3 sensor [34]. The sensor has two 1(m)

ace	Classifier	$Classification \ method$	Estimation method
s Sp	Mahalanobis	$\hat{i} = \arg\min_i (\hat{p}_i - \overline{p}_i)^T C_i^{-1} (\hat{p}_i - \overline{p}_i)$	$\hat{p}_i, \hat{\nu}_i = \arg\min_{p,\nu} \ C^{-\frac{1}{2}}(y - m(p,\nu))\ _2^2$
ture	KDE	$\hat{i} = \arg\max_i f_i(\hat{p}_i)$	$\hat{p}_i, \hat{\nu}_i = \arg\min_{p,\nu} \ C^{-\frac{1}{2}}(y - m(p,\nu))\ _2^2$
Fea			$-\ln f_i(p)$
	SVM	$\operatorname{sgn}(\sum_{s_i \in \text{support vectors}} \alpha_i y_i s_i \hat{p}_i + b)$	$\hat{p}_i, \hat{\nu}_i = \arg\min_{p,\nu} \ C^{-\frac{1}{2}}(y - m(p,\nu))\ _2^2$
Ge	ML-GLRT	$\hat{i} = \arg\max_{i} f(y/H_i, \hat{\theta})$	$\hat{\theta} = \arg\min_{\theta} \ C^{-\frac{1}{2}}(y - m(\theta))\ _2^2$
Spa	MAP-GLRT	$\hat{i} = \arg\max_{i} f(y/H_i, \hat{p}_i, \hat{\nu}_i) f_i(\hat{p}_i)$	$\hat{p}_i, \hat{\nu}_i = \arg\min_{p,\nu} \ C^{-\frac{1}{2}}(y - m(p,\nu))\ _2^2$
lal 9			$-\ln f_i(p)$
Sigr	LA-GLRT	$\hat{i} = \arg\max_{i} \left\{ -\frac{1}{2} \ln(\det(J)) \right\}$	$\hat{p}_i, \hat{\nu}_i = \arg\min_{p,\nu} \ C^{-\frac{1}{2}}(y - m(p,\nu))\ _2^2$
		$+\ln f_i(\hat{p}_i) + \ln f(y/\hat{p}_i, \hat{\nu}, H_i)\}$	$-\ln f_i(p)$
	Hybrid	$\hat{i} = \arg\max_{i} \int f(y/H_i, p, \hat{\nu}) f_i(p) dp$	$\hat{\nu}_i = \arg\min_{\nu} \ C^{-\frac{1}{2}}(y - m(p, \nu))\ _2^2$

Table 4.1: Summary of classification and estimation methods

diameter circular coils for transmitter and receiver and samples on a one meter square area using an equally spaced  $5\times5$  grid of measurement points. There are 20 frequency sample ranging from 20(Hertz) to 20k(Hertz) on log scale. The objects include: a three inch long by one inch diameter stainless steel cylinder (S1), a six inch long by one inch diameter stainless steel cylinder (S2), a three inch long by one inch diameter aluminum cylinder (A1) and a six inch long by one inch diameter aluminum steel cylinder (A2). The target responses were generated with a four pole per axis dipole model in which the sums in (2.1.3) are terminated after four terms [8, 10]. The poles used for this simulation were generated using the methods in [8]. The objects were chosen such that the pole distribution covers most of the feature space with similar features for the pair (S1, S2) and (A1, A2) to make classification more challenging. Figure 4.1 shows the pole distributions for these objects. Classification performance was evaluated at different SNR values using 100 Monte Carlo runs for each case. The SNR is computed as:

$$SNR = 10\log_{10}\left(\frac{\parallel y \parallel^2}{N\sigma^2}\right) \tag{4.3.1}$$

where y is the received signal, N is the length of y and  $\sigma^2$  is noise variance. The probability of error is calculated using:

$$p_e = \frac{1}{4} \sum_{i=1}^{4} P(\text{error}/H_i)$$
(4.3.2)

which corresponds to average sum of off-diagonal elements of confusion matrix. Figures 4.2



Figure 4.1: Pole distributions for S1 (blue), S2 (red), A1 (black) and A2(green) objects. See text for a physical description of targets.

and 4.3 show the probability of error as a function of SNR for feature and signal classifiers respectively. Comparing the plots, we see that the signal classifiers (GLRT, Hybrid and Laplacian) have considerably better performance compared to the feature space classifiers (Mahalanobis, kernel density classifier(KDE) and SVM). Also the probability of error decreases rapidly as SNR increases for signal space classifiers comparing to feature space classifiers.

Table 4.2 shows the confusion matrices for the Mahalanobis, KDE, SVM, MAP-GLRT and hybrid classifiers. It can be seen that only the Mahalanobis classifier confuses a steel target with an aluminum target. The hybrid classifier has the best performance, and shows an 80% improvement over simple mean-covariance classifier used in [10]. The Laplacian classifier follows the hybrid classifier except at low SNR. This situation can be explained in a couple of ways. Most directly, as the SNR decreases, the Gaussian approximation to the integrand of (4.2.9) becomes less valid. Alternatively, as the SNR decreases we are seeing a corresponding



Figure 4.2: Probability of error as a function of SNR for feature based classifiers



Figure 4.3: Probability of error as a function of SNR for signal based classifiers

Mahalanobis classifier								KDE		
	S1	S2	A1	A2			S1	S2	A1	A2
S1	66	34	0	0		S1	72	28	0	0
S2	44	56	0	0		S2	28	72	0	0
A1	1	6	77	16		A1	0	0	72	28
A2	1	4	0	95		A2	0	0	0	100
		SVM					MA	P-GI	$_{\rm LRT}$	
	S1	S2	A1	A2			S1	S2	A1	A2
S1	78	21	0	0		S1	100	0	0	0
S2	50	45	5	0		S2	8	92	0	0
A1	0	0	98	2		A1	0	0	100	0
A2	0	0	0	100		A2	0	0	15	85
LA-GLRT Hybrid						h				
	S1	S2	A1	A2	] [		S1	S2	A1	A
S1	100	0	0	0		S1	100	0	0	0
S2	2	98	0	0	] [	S2	0	100	0	0
$\Delta 1$	0	0	03	7	ן ן	Δ1	0	0	97	3

Table 4.2: Confusion matrices for simulated targets at SNR of 30 DB

A2

0

7

93

0

93

A2

0

0

7

decrease in accuracy in the estimation of the model parameters. Roughly speaking, as the parameter estimates worsen the OFIM gets larger, indicating a worse fit of the model to the data, and the  $-\log \det J$  term in (4.2.11) becomes increasingly negative thereby making the corresponding hypothesis less likely.

Classification of field data was undertaken with EMI data collected using an EM63 sensor over various UXO and metallic clutter surveyed at the US Army Corps of Engineers Engineer Research and Development Center (ERDC) UXO/Countermine test stand. The test stand allows precise positioning of the target and the sensor in terms of both relative location and orientation. The data were collected over a variety of spatial grids, ranging in spacing from 5cm to 20cm. Often, several spacings were present in a single grid, with higher sampling density near the target where the gradient of the target field was largest. A suite of 18 standard UXO targets were examined, as well as several clutter items and spheres of assorted compositions. In this study, submunitions of diameters 26mm and 28mm (BDU-26 and BDU-28, respectively) are examined, as well as two mortar shells with diameters of 90mm and 155mm. These four

	Mahal	lanobis	classifi	er
--	-------	---------	----------	----

	26mm	28mm	90mm	$155 \mathrm{mm}$
26mm	2	1	3	1
28mm	0	6	1	1
90mm	2	1	7	0
155mm	0	6	2	0

### Percent Error = 54%

KDE

	26mm	$28 \mathrm{mm}$	$90 \mathrm{mm}$	$155 \mathrm{mm}$
26mm	6	0	0	1
28mm	0	4	1	3
90mm	3	0	5	2
155mm	1	1	0	6
	<b>D</b>		0.001	

Percent Error = 36%

- N/		וי	<b>P</b> '	'
1V.	LL-'	J	ιυ	L

	26mm	$28 \mathrm{mm}$	$90 \mathrm{mm}$	$155 \mathrm{mm}$
26mm	2	4	1	0
28mm	1	2	2	3
90mm	5	0	2	3
$155 \mathrm{mm}$	3	1	2	2

## Percent Error = 75%

## MAP-GLRT

	26mm	$28 \mathrm{mm}$	$90 \mathrm{mm}$	$155 \mathrm{mm}$		
26mm	6	0	1	0		
28mm	0	4	1	3		
90mm	2	0	7	1		
155mm	0	1	2	5		

#### Percent Error = 33%LA-GLRT

26mm 28mm 90mm 155mm							
26mm	6	1	0	0			
28mm	2	6	0	0			
90mm	3	0	5	2			
155mm	0	2	0	6			
Percent Error $= 30\%$							

Table 4.3: Confusion matrices summarizing the classification of several UXO targets from the ERDC test stand data set. Classification was performed using a leave-one-out method.



Figure 4.4: Poles distributions for BDU-26 ( $\times$ ), BDU-28 (o), 90mm (diamond) and 155mm (star) UXO items

targets were chosen because of the relatively large number of separate data sets collected compared to the other targets in the suite. The data sets differ from one another in target pose, orientation, and target instance. The accuracy and statistical significance of the KDE pdf increases as the number of parameter estimates increases. It should be noted, however, that even for these four targets, the number of samples was used to estimate the pdf was small. Appendix B provides details of field data processing including search initialization, early and late time gate emphasis procedures.

Figure 4.4 shows the pole distribution of these objects. A leave-one-out method was employed, in which all data sets but one for a particular target are used to construct a target library, along with the data from all other targets. Classification is then attempted using the remaining data set. The noise and its covariance matrix to be used in classification algorithms was estimated using residual snapshots over spatial domain after estimating the parameters



Figure 4.5: Noise histogram estimated using spatial and temporal residuals

using unit variance noise. So noise estimate can be written as:

$$\hat{n}^{T} = [\hat{n}_{1}^{T} \quad \hat{n}_{2}^{T} \quad \cdots \quad \hat{n}_{M}^{T}]^{T} = y - m(\hat{\theta})$$
(4.3.3)

with each column corresponding to spatial estimated residuals at each time gate. Estimated noise covariance matrix now can be calculated as:

$$\hat{C} \approx \frac{1}{M} \hat{n} \cdot \hat{n}^T \tag{4.3.4}$$

Figure 4.5 shows histogram of the estimated noise using all spatial and temporal residual information. Figure 4.6 shows noise covariance matrix estimated using residual snapshots over spatial domain from (4.3.4). Twenty-six time gates where used in the experiments, and the covariance matrix rank was 23. The ideal matrix should be a diagonal one, but due some model mismatch which is one of the characteristics of field data processing we see some cross correlation between temporal samples. As we are using these matrix as weighting in the optimization problem, we are basically considering this model mismatch into are processing. Figure 4.7 shows spectral structure of estimated noise covariance matrix. The results are summarized in Table 4.3 as confusion matrices for each algorithm. Because of low training



Figure 4.6: Noise covariance matrix estimated using residual snapshots over spatial domain



Figure 4.7: spectral structure of estimated noise covariance matrix

samples the SVM classifier did not work for this case, demonstrating the superiority of kernel classifier when few training samples are available.

Finally, the absence of the hybrid classifier in Table 4.3 is due to its poor performance in the processing of field data. For these data sets, the integrand in (4.2.8) evaluates to zero even in double precision arithmetic; a situation driven by the approximate nature of the underlying dipole signal model. To be specific, within the integrand of the hybrid model is a term of the form:

$$\exp\left\{-\frac{1}{2}\|C^{-1/2}(y-m(p,\hat{\nu}))\|_{2}^{2}\right\}$$
(4.3.5)

that arises under the assumption that the difference between the data and the model used in classification is Gaussian noise. Moreover, given that the data from ERDC only supported the estimation of the diagonal elements of C, the noise is also assumed to be uncorrelated. Now, in truth it is well known that the dipole model used to construct m is at best an approximation. Thus, even when the model parameter estimates are accurate, the difference between y and mwill not be uncorrelated Gaussian noise. Thus,  $\|C^{-1/2}(y-m(p,\hat{\nu}))\|_2^2$  can be substantial resulting in (4.3.5) being incorrectly rounded to zero in finite precision arithmetic. This situation is avoided in the other classifiers as they operate on the logarithm of (4.3.5) thereby avoiding what are essentially a mix of model mismatch and dynamic range difficulties. In this sense then, the classifiers considered in Table 4.3 though not "optimal" under the assumptions used to construct (2.2.1) are certainly more robust in practice to these assumptions being violated.

Another question that arises here is the effect of the bandwidth of the KDE on classification performance for this low training ERDC samples. To this end we have investigated the effect of bandwidth on the classification error performance where the nominal bandwidth is varied by a multiplication factor from .5 to 2. It was observed that only result of the KDE classifier was sensitive to the variation of the bandwidth and its performance achieved its minimum at nominal value of the bandwidth (corresponding to multiplication factor 1). Figure 4.8 shows the result which justifies bandwidth selection method explained in Chapter 3.



Figure 4.8: Effect of bandwidth on performance of KDE classifier for ERDC data sets.

# Chapter 5 Performance Analysis

In previous chapter, new suite of estimation and classification algorithms using non-parametric priors were introduces. In this chapter we will derive analytical bounds on performance of relevant estimation and classification algorithms. As explained in the introduction here we will focus on optimal classifiers in data and feature space for the purpose of analysis as their analytic nature are suited well for deriving analytical performance bounds using statistical methods. These performance bounds will be used in the next Section to explore the effect of sensor parameters on system performance as well as exploring ways towards developing optimized sensors tuned for best performance. First we will review shortly the relevant estimation and classification algorithms which we will be using for purpose of analysis.

# 5.1 Estimation And Classification Methods

In Chapter 4 we introduced new set of estimation and classification algorithms. Here we will review shortly methods which we will use in our analysis.

## 5.1.1 Estimation

As explained in Section 4.1, there are two approaches for estimation of features: maximum likelihood (ML) estimation and maximum a-posteriori (MAP) estimation for features using our nonparametric priors. ML estimation can be written as:

$$\hat{p}_i, \hat{\nu}_i = \operatorname*{arg\,max}_{p,\nu} f_i(y/p,\nu) \tag{5.1.1}$$

MAP estimation can be written as:

$$\hat{p}_i, \hat{\nu}_i = \operatorname*{arg\,max}_{p,\nu} f_i(y/p,\nu) f_i(p)$$
(5.1.2)

where  $f_i(p)$  is the PDF of poles corresponding to *ith* object. Given the assumption of Gaussian noise, (5.1.2) is equivalent to:

$$\hat{p}_i, \hat{\nu}_i = \arg\min_{p,\nu} \|C^{-1/2}(y - m(p,\nu))\|_2^2 - \ln f_i(p).$$
(5.1.3)

## 5.1.2 Optimal Classification

As explained in Section 4.2, there are two types of optimal classifiers: data space and feature space. The optimal classifier is a signal based classifier which employs the entire received signal for classification. The other optimal classifier which is based on assumption that features are sufficient statistics, operates in feature space and uses estimated PDF of features to implement optimal classifier in feature space. For analysis purpose here we will consider optimal data space in Equation (4.2.3) and kernel feature based classifier classifier which are reviewed next.

#### Optimal data space classifier

The classification rule which minimizes the probability of error is the Bayes decision rule [21]:

$$\hat{i} = \underset{i}{\arg\max} f(H_i/y) \tag{5.1.4}$$

where  $f(H_i/y)$  is the posterior probability distribution of *ith* hypothesis. Using Bayes rule, (5.1.4) can be written as:

$$\hat{i} = \arg\max_{i} \frac{f(y/H_i)f(H_i)}{f(y)}$$
(5.1.5)

where  $f(H_i)$  is the prior distribution of each hypothesis, and is assumed to be uniform in this paper. Expanding  $f(y/H_i)$  in terms of priors using the Bayesian rule and ignoring the constant denumerator, the optimal Bayes classifier can be written as:

$$\hat{i} = \arg\max_{i} \int f(y/H_i, \theta) f_i(\theta) d\theta.$$
(5.1.6)

where  $f_i(\theta) = f_i(p)f(\nu)$  is the prior probability distribution of parameters, assuming a uniform distribution for the nuisance parameters. The vector  $\theta = [p^T, \alpha^T, r^T, c^T, \delta^T]^T = [p^T, \nu^T]^T$  has 13 parameters including features and nuisance parameters.

#### Optimal Feature space classifier(KDE)

If we were to observe directly the pole vector, the optimal feature-space classifier takes a form like:

$$\hat{i} = \arg\max_{i} f_i(p) \tag{5.1.7}$$

where  $f_i(p)$  is the PDF associated with the hypothesis that the observed poles come from object *i*. Now, it is clearly the case that we do not observe the poles directly, but rather they are estimated from the data. Thus, as explained in previous chapter we have developed generalized ratio-type classifiers which basically used our estimates of the poles in place of *p* in (5.1.7). Using the kernel density estimated prior (3.1.1), we can estimate the poles using the MAP estimator and implement (5.1.7) directly:

$$\hat{i} = \arg\max_{i} f_i(\hat{p}_i) \quad \text{where} \quad \hat{p}_i, \hat{\nu}_i = \arg\min_{p,\nu} \|C^{-\frac{1}{2}}(y - m(p,\nu))\|_2^2 - \ln f_i(p).$$
(5.1.8)

# 5.2 Estimation Performance Analysis

## 5.2.1 Cramer-Rao Lower Bounds

Cramer-Rao theorem provides lower bounds of error variance for unbiased estimators [15] [35]. To identify our estimator performance the bound is derived for both scalar and random parameter assumptions which correspond to maximum likelihood and Bayesian estimation. Using Fisher Information Matrix sensitivity of feature parameters respect to object spatial information such as orientation and location are evaluated, this analysis reveals how much compromise we make to estimate spatial information while at the same time attempting to classify the objects. As explained in the introduction this would be useful as instruments like magnetometers can assist us to infer about spatial geometry of the object. For maximum likelihood estimator where the parameters are assumed to be scalars the estimator error variance for a vector parameter  $\theta$  can be written as:

$$C = E[\hat{\theta} - \theta][\hat{\theta} - \theta]^T \ge J^{-1}$$
(5.2.1)

$$J(\theta) = E[s(\theta, y)s(\theta, y)^{H}] \quad \text{where} \quad s(\theta, y) = \frac{\partial}{\partial \theta} \ln f_{\theta}(y) \tag{5.2.2}$$



Figure 5.1: CRLB and BCRLB for poles estimates of x, y and z axis using EM63 sensor for 81mm object

Applying the above equation to our model (2.1.4) and using matrix derivative formula, we have:

$$J = E \sum_{i=1}^{M} \sum_{j=1}^{M} \frac{\partial m_{i(\theta)}}{\partial \theta} C_{N_i}^{-1} (y_i - m_i(\theta)) (y_j - m_j(\theta))^T C_{N_j}^{-1} \left(\frac{\partial m_{j(\theta)}}{\partial \theta}\right)^H$$
(5.2.3)

simplifying this expression gives:

$$J = Re \sum_{i=1}^{M} \frac{\partial m_i(\theta)}{\partial \theta} C_{N_i}^{-1} \left( \frac{\partial m_i(\theta)}{\partial \theta} \right)^H$$
(5.2.4)

In Bayesian framework where the parameters are considered as random variables we have the following bound for mean square error of estimators namely Bayesian Cramer Rao lower bound (BCRLB) [17]:

$$C = E_{y,\theta} [\hat{\theta} - \theta] [\hat{\theta} - \theta]^T \ge J_T^{-1}$$
(5.2.5)

$$[J_T]_{ij} = -E_{y,\theta} \left[ \frac{\partial^2 \ln f(y/\theta)}{\partial \theta_i \partial \theta_j} \right] - E_{y,\theta} \left[ \frac{\partial^2 \ln f(\theta)}{\partial \theta_i \partial \theta_j} \right]$$
$$= \int d\theta f(\theta) [J_D]_{ij}(\theta) + [J_A]_{ij}$$
(5.2.6)

where  $f(\theta)$  is the prior probability density function,  $J_D$  is local fisher information and  $J_A$  is the local fisher information corresponding to prior information. ML CRLB is derived numerically using finite difference approximation and is validated using Monte Carlo simulation at different signal to noise ratios. Parameter vector is  $\theta = [p^T, \alpha^T, r^T, c^T, \delta^T]^T$  consisting of poles, orientation, location, coefficient factors and DC parameter. Figure 5.1 shows Cramer Rao lower bounds on variance of poles using (5.2.1) and (5.2.5) and Monte Carlo simulations at different SNR ratios in terms of standard deviation per nominal value corresponding to low, moderate low, moderate high and high SNRs. It can be seen that incorporation priors has reduced the estimation error specially at low SNR even below the ML CRLB, this is because prior increases the information by amount of a positive definite matrix  $J_A$  in Equation (5.2.6) and it is more noticeable at low SNR. Also it can be seen that MAP is more robust to SNR.

#### Effect of sampling and unknown spatial information on CRLB

Up to now all the analysis were based on the assumption that we are trying to infer about geometry of the UXO as well as classification. To see how much we are compromising between accuracy in inferring these parameters and classification performance. In this Section we will analyze effect of inferring about geometry on the estimation of poles. Furthermore we investigate the effect of sampling density on the estimation of poles in the case of unknown and known geometry. As we will see the sampling density effect can be modelled as an equivalent noise.

In order to explore the effect of spatial information on the estimation of poles, the parameter space is divided into features and nuisance parameters as  $\theta = [p^T \nu^T]^T$  as explained in Section 2.1 then Fisher information matrix (FIM) can be expressed as a partitioned matrix:

$$J(p,\nu) = \begin{pmatrix} J_{pp}(p,\nu) & J_{\nu p}(p,\nu) \\ J_{pp}^{H}(p,\nu) & J_{\nu \nu}(p,\nu) \end{pmatrix}$$
(5.2.7)

Using FIM the effect of spatial information such as location and orientation on the variance of feature parameters can be investigated which gives a sensitivity analysis of pole space to nuisance parameters [36]. Applying partitioned matrix inversion formula [37] to (5.2.7) we will have:

$$C_{p,\nu}(p,\nu) = [J_{p\,p}(p,\nu) - J_{\nu\,p}(p,\nu)J_{\nu\,\nu}^{-1}(p,\nu)J_{\nu\,p}^{H}(p,\nu)]^{-1}$$
(5.2.8)

where  $C_{p,\nu}$  is a 3 × 3 matrix corresponding to error covariance matrix for feature space (poles) and its diagonal elements are the error variance of each pole. This matrix gives CRLB for unknown geometry. To evaluated CRLB for the case of known geometry it is sufficient to invert the sub matrix  $J_{pp}(p,\nu)$ :

$$C_p(p,\nu) = J_{pp}^{-1}(p,\nu)$$
(5.2.9)

Using above equation CRLB for known and unknown geometry can be calculated as a function of sampling density. This is done in Figures 5.2, 5.3 and 5.4 show CRLB for 81mm mortar shell using one pole per axis model (see appendix B for details) for known and unknown geometry at different samplings and noise variance corresponding to 5cmm and 20cmm sampling grids and 10 times nominal noise variance. As can be seen from the figures CRLB increase as the depth of UXO increases. Also we note that as dip angel varies from 0° to 90° degrees, CRLB for z axis pole increases, which indicates that we are less able to estimate this pole well. This make sense as the object is becoming parallel to the sensor and sensor is not being able to see the object on that axis. Also we note that though the object is symmetric on x and y axis CRLB for this axis have slight difference, this is due to square grid sampling instead of circular sampling which makes the object slightly nonsymmetric with respect to sensor. Also this is because object is not exactly located at the center of sampling.



Figure 5.2: Effect of unknown geometry on CRLB as a function of sampling and noise variance for x axis pole

Previous plots shows the effect of known and unknown geometry as a function of sampling density and noise variance. In order to investigate this, we would like to have a sensitivity



Figure 5.3: Effect of unknown geometry on CRLB as a function of sampling and noise variance for y axis pole



Figure 5.4: Effect of unknown geometry on CRLB as a function of sampling and noise variance for z axis pole

measure which is independent of noise variance. To calculated sensitivity of CRLB with respect to geometry, relative percentage increase in CRLB for the case of unknown geometry with respect to known geometry is calculated as

$$r = \frac{C_{p,\nu}(p,\nu) - C_p(p,\nu)}{C_p(p,\nu)}(\%)$$
(5.2.10)

at different orientations (dip angle) and depths. Figure 5.5 shows the result of sensitivity



Figure 5.5: Sensitivity of CRLB with respect to orientation and depth uncertainty

analysis for the same 81mm mortal shell object as dip angel varies from  $0^{\circ}$  to  $90^{\circ}$  degrees. It was noted that different sampling densities and noise variance as shown in Figures 5.2, 5.3 and 5.4, have the same sensitivity curves, so the effect of sampling density can be modeled as an effective increase or decrease in the noise or signal power. Also it can be seen that z-axis pole is influence more that other poles specially at  $90^{\circ}$  dip angle cause in that case the z-axis of the object is in parallel with sensor and there is low information to be able to estimate this pole. It is also interesting that we have a maximum at  $45^{\circ}$  dip angle which occurs because at two extreme dip angles the estimation performs the same for the case of known and unknown geometry (both are poor or good estimations at these angles) so we have the maximum increment around 45°.

# 5.3 Error Probability Analysis

To evaluate error performance of UXO classification, Chernoff upper bound on error probability of optimum data space classifier and feature based kernel density classifier (KDE) introduced in Section 5.1 are evaluated analytically using our prior model. For data space classifier Laplacian and asymptotic techniques are used to overcome the problem of evaluating high dimensional integration and to get closed form analytical expression which will be useful for sensor performance optimization. For feature based classifier the upper bound is evaluated directly using constructed libraries.

### 5.3.1 Optimum Data Space Classifier

Using the processing model in equations (2.1.1) and (2.1.4) we denote the sampled signal vector as  $r^n$  where  $n = L \times M$  the total sampled points in location and frequency. The optimum classifier as mentioned in Section 5.1 is MAP classifier and from equation (5.1.6) it is proportional to conditional likelihood given each hypothesis. Assuming we have two objects the Chernoff bound on the probability of error of a maximum *a posteriori* (MAP) classifier with decision function  $u^{MAP} = \operatorname{argmax}_{k \in 1,2} \pi_k f_{y^n \mid H_k}(y^n \mid H_k)$  based on *n* observation is given by

$$P_e^n \le \inf_{0 \le s \le 1} \int [f_1(y^n)]^s [f_2(y^n)]^{1-s} dr^n = D_{12}^n$$
(5.3.1)

where D denotes the Kullback divergence [38].

As the number of samples goes to infinity the Chernoff bound becomes achievable and the asymptotic probability of error can be written as

$$P_e = \lim_{n \to \infty} (P_e^n)^{\frac{1}{n}} = (D_{12}^n)^{\frac{1}{n}}.$$
(5.3.2)

Or equivalently for sufficient number of samples

$$\ln(P_e) \approx \frac{1}{n} \ln P_n = \frac{1}{n} \ln D_{12}^n.$$
(5.3.3)

For the multiclass hypothesis case, we have M distributions corresponding to each object as:

$$\{f_1(y^n), f_2(y^n), ..., f_M(y^n)\}$$
 with priors  $\{\pi_1, \pi_2, ..., \pi_M\}.$  (5.3.4)

Using the union bound, we can write

$$P_e \le \sum_{k=1}^{M-1} \sum_{j=k+1}^{M} \pi_j P_e[\pi_k f_k(y^n) > \pi_j f_j(y^n) | H_j] + \pi_k P_e[\pi_j f_j(y^n) > \pi_k f_k(y^n) | H_k]$$

Now using the Chernoff bound we will have

$$P_e \le \sum_{k=1}^{M-1} \sum_{j=k+1}^{M} \pi_k \exp\left(\frac{1}{n} \ln D_{kj}^n\right).$$
(5.3.5)

For equal priors the bound can be expressed as

$$P_{e} \leq M^{-1} \sum_{k=1}^{M-1} \sum_{j=k+1}^{M} \exp\left(\frac{1}{n} \ln D_{kj}^{n}\right)$$

$$D_{kj}^{n} = \min_{0 \leq s_{jk} \leq 1} \int [f_{k}(y^{n})]^{s_{jk}} [f_{j}(y^{n})]^{1-s_{jk}} dr^{n}.$$
(5.3.6)

To obtain the Chernoff upper bound we need to find the likelihood distribution of each hypothesis  $f_i(y^n)$ . We use Laplacian and asymptotic approximations to derive analytical expressions for this upper bound. Using our prior distribution of parameters,  $f_i(\theta)$ , the likelihood function can be written as:

$$f_{y^n|H_i}(y^n|H_i) = \int f(y^n|H_i,\theta)f_i(\theta)d\theta$$
(5.3.7)

where  $f(y^n|H_i,\theta)$  is a Gaussian distribution with mean  $m_i(\theta)$ , the sensor model evaluated at specific object parameters. Now we use Laplacian approximation to evaluate the right hand side of (5.3.7) as:

$$\int f(y^n | H_i, \theta) f_i(\theta) d\theta \approx \frac{(\sqrt{2}\pi)^d}{\sqrt{\det[J_i(\hat{\theta})]}} f_i(\hat{\theta}) f_i(y^n | \hat{\theta}, H_i)$$
(5.3.8)

where  $J_i(\hat{\theta}) = -\frac{\partial}{\partial \theta} \left( \frac{\partial}{\partial \theta} \log f_i(y^n | \hat{\theta}, H_i) \right)^T \Big|_{\theta = \hat{\theta}}$  is the Observed Fisher Information Matrix (OFIM). From the physical model we know that  $f_i(y^n | \hat{\theta}, H_i)$  has a gaussian distribution with

mean  $m_i(\hat{\theta})$ , the sensor model evaluated at estimated parameter. Now equations (5.3.6) and (5.3.8) can be used to evaluate the performance bound:

$$D_{kj}^{n} = \min_{0 \le s_{jk} \le 1} \int [f_{k}(y^{n})]^{s_{jk}} [f_{j}(y^{n})]^{1-s_{jk}} dy^{n}$$

$$\approx \min_{0 \le s_{jk} \le 1} \frac{(\sqrt{2}\pi)^{d} f_{k}^{s_{kj}}(\hat{\theta}) f_{j}^{1-s_{kj}}(\hat{\theta})}{\sqrt{(\det[J_{k}(\hat{\theta})])^{s_{kj}} (\det[J_{j}(\hat{\theta})])^{1-s_{kj}}}} \int f_{k}^{s_{kj}} (y^{n}|\hat{\theta}, H_{k}) f_{j}^{1-s_{kj}} (y^{n}|\hat{\theta}, H_{j}) dy^{n}$$

$$= \min_{0 \le s_{jk} \le 1} \frac{(\sqrt{2}\pi)^{d} f_{k}^{s_{kj}}(\hat{\theta}_{k}) f_{j}^{1-s_{kj}}(\hat{\theta}_{j})}{\sqrt{(\det[J(\hat{\theta}_{k})])^{s_{kj}} (\det[J(\hat{\theta}_{j})])^{1-s_{kj}}}} \int f_{k}^{s_{kj}} (y^{n}|\hat{\theta}_{k}) f_{j}^{1-s_{kj}} (y^{n}|\hat{\theta}_{j}) dy^{n}$$
(5.3.9)

where we have substituted  $(\theta, H_k)$  by  $\theta_k$  which indicates estimated parameter assuming hypothesis  $H_k$  is in force. Because the additive noise is assumed to be IID, the second term can be written as the product of n one dimensional integrals involving Gaussian distributions where each distribution has a mean equal to sensor model evaluated at specific sample for corresponding object and we will have:

$$D_{kj}^{n} \approx \min_{0 \le s \le 1} \frac{(\sqrt{2}\pi)^{d} f_{k}^{s}(\hat{\theta}_{k}) f_{j}^{1-s}(\hat{\theta}_{j})}{\sqrt{(det[J(\hat{\theta}_{k})])^{s} (det[J(\hat{\theta}_{j})])^{1-s}}}$$
(5.3.10)  
$$\cdot \prod_{i=1}^{n} \left\{ \int \left[ \frac{1}{\sqrt{2\pi}\sigma_{j}} \exp \frac{-(y_{i} - m_{i}(\hat{\theta}_{j}))^{2}}{2\sigma_{j}^{2}} \right]^{s} \cdot \left[ \frac{1}{\sqrt{2\pi}\sigma_{k}} \exp \frac{-(y_{i} - m_{i}(\hat{\theta}_{k}))^{2}}{2\sigma_{k}^{2}} \right]^{1-s} dy_{i} \right\}$$

After taking natural logarithm of both sides and using integration formulas [39], equation (5.3.10) can be simplified to:

$$\ln D_{kj}^{n} \approx \min_{0 \le s \le 1} \frac{d}{2} \ln(\pi) - \frac{s}{2} \ln(\det[J(\hat{\theta}_{k})]) - \frac{1-s}{2} \ln(\det[J(\hat{\theta}_{j})])$$

$$+ s \ln(f_{k}(\hat{\theta}_{k})) + (1-s) \ln(f_{j}(\hat{\theta}_{j})) -$$

$$\sum_{i=1}^{n} \left\{ \frac{s(1-s)}{2} \frac{(m_{i}(\hat{\theta}_{j}) - m_{i}(\hat{\theta}_{k}))^{2}}{s\sigma_{ij}^{2} + (1-s)\sigma_{ik}^{2}} + \frac{1}{2} \ln \frac{|s\sigma_{ij}^{2} + (1-s)\sigma_{ik}^{2}|}{|\sigma_{ij}^{2}|^{s}|\sigma_{ik}^{2}|^{1-s}} \right\}$$
(5.3.11)

As can be seen the upper bound requires a minimization over parameter s, usually in practice for simplicity of evaluation it is assumed that  $s = \frac{1}{2}$  which corresponds to Bhattacharya bound. Now equation (5.3.12) cab be written as:

$$\ln D_{kj}^{n} \approx \frac{d}{2}\ln(\pi) - \frac{1}{4}\ln(\det[J(\hat{\theta}_{k})]) - \frac{1}{4}\ln(\det[J(\hat{\theta}_{j})]) + \frac{1}{2}\ln(f_{k}(\hat{\theta}_{k})) + \frac{1}{2}\ln(f_{j}(\hat{\theta}_{j}))) \\ - \sum_{i=1}^{n} \left\{ \frac{1}{4} \frac{(m_{i}(\hat{\theta}_{j}) - m_{i}(\hat{\theta}_{k}))^{2}}{\sigma_{ij}^{2} + \sigma_{ik}^{2}} + \frac{1}{2}\ln\frac{|\sigma_{ij}^{2} + \sigma_{ik}^{2}|}{2|\sigma_{ij}||\sigma_{ik}|} \right\}.$$
(5.3.12)



Figure 5.6: Error probability upper bound as a function of signal to noise ratio for distinguishing between aluminum and a steel objects using GEM3 sensor

Now the upper bound on error probability can be calculated easily using equations (5.3.12) and (5.3.6). It can be seen that the upper bound is exponentially related to model differences and variance of samples. In the case of equal variance the bound will be an exponential function of euclidian distance of signal models:

$$\ln D_{kj}^{n} \approx \frac{d}{2}\ln(\pi) - \frac{1}{4}\ln(\det[J(\hat{\theta}_{k})]) - \frac{1}{4}\ln(\det[J(\hat{\theta}_{j})]) + \frac{1}{2}\ln(f_{k}(\hat{\theta}_{k})) + \frac{1}{2}\ln(f_{j}(\hat{\theta}_{j}))) - \frac{1}{8}\sum_{i=1}^{n}\frac{(m_{i}(\hat{\theta}_{j}) - m_{i}(\hat{\theta}_{k}))^{2}}{\sigma_{i}^{2}}.$$
(5.3.13)

The terms  $\ln(f_k(\hat{\theta}_k))$  which are always negative (therefor reduce the upper bound) are the result of incorporating the priors in processing model and can be interpreted as the performance boosting factor as a result of incorporation of priors into the processing. The terms  $\ln(det[J(\hat{\theta}_k)])$  relate the estimation accuracy to error performance through FIM and reveals the influence of estimation accuracy on the classification performance. The integral involves evaluation of  $\hat{\theta}_j$ , which is the estimate of  $\theta$  for object j where the hypothesis corresponding to object j is in force. In asymptotic analysis, it is assumed that the noise variance  $\sigma_j^2 \to 0$ , so that the estimates approach the true values  $(\hat{\theta}_j \to \theta_j)$ , which corresponds to a deterministic variable. Figure 5.6 shows Chernoff upper bound as a function of SNR for simulated



Figure 5.7: Error probability upper bound for distinguishing between 81 mm and a 60 mm object using EM63 sensor on a  $5 \times 5$  sampling grid



Figure 5.8: Error probability upper bound for distinguishing between 81 mm and a 60 mm object using EM63 sensor on a  $15 \times 15$  sampling grid



Figure 5.9: Error probability upper bound and SNR for distinguishing between 105 mm, 81 mm, 60 mm and 40 mm objects using EM63 sensor as a function of number of spatial samples



Figure 5.10: Error probability upper bound and SNR for distinguishing between 81 mm and a 60 mm object using EM63 sensor as a function of number of spatial samples
Aluminum and Steel objects explained in Section 4.3. Figure 5.7 shows the Chernoff upper bound as a function of SNR for the problem of distinguishing the 81 mm from a 60 mm object using EM63 sensor on a  $5 \times 5$  square grid and Figure 5.8 shows the same results for a  $15 \times 15$ square grid. Here again the pole clouds were generated using the quasi-real objects explained in Section 2.2. Figure 5.9 shows error probability and SNR as a function of number of spatial samples for distinguishing between 105mm, 81mm, 60mm and 40mm quasi-real objects using EM63 sensors while the size of the overall square is kept constant. SNR is calculated using average SNR of targets defined as:

$$SNR = \frac{1}{4} \sum_{i=1}^{4} SNR_i$$
 (5.3.14)

where  $SNR_i$  is SNR for *ith* target and can be written as:

$$SNR_i = 10 \log_{10} \left( \frac{\parallel y_i \parallel^2}{N\sigma^2} \right).$$
 (5.3.15)

We note that as the number of samples increases the SNR remains almost constant, this is because both signal power and number of samples increase as the number of samples increases within that range. Figure 5.10 shows error probability and SNR as a function of number of spatial samples for the problem of distinguishing the 81 mm from a 60 mm object using EM63 sensors from low spatial samples to high spatial samples for square grid sampling scheme while the size of the overall square is kept constant. It is interesting that for low spatial sampling SNR does not necessary increase with number of samples, this is because the corresponding square grid sampling for low sampling does not necessary sample the samples with high amplitude. This fact leads to the idea of optimal sampling should hit the high amplitude samples. This idea is explored more in Chapter 6.

A question that one might ask here is how realistic these analytical bounds are. Comparing the results for EM63 sensor in Figure 5.9 which show that we can have very accurate classification of these targets with the results we got from real data sets in Tables 4.3 might lead us to believe that the bounds here do not have any practical application. We first note that the bounds here are derived for optimal Bayesian classifier which are intractable for implementation, therefore in that sense they are inherently optimistic. Second it should be noted that the objects used in the bounds for EM63 in this section are quasi-real objects which have a very good model match to the three pole model we use in our processing. In reality the underlying model is far from quasi-real dipole model and therefore the results will vary, an accurate model for the real data would capture this mismatch and will result in more feasible numerical conclusions from these analytical bounds. The result of GEM3 sensor here confirms this fact, where the upper bound for GEM3 sensor for aluminum and steel object in Figure 5.6 are in coincidence with number simulation in Section 4.3. This is because the model used for deriving the bounds and the model used in experiments are the same. Another approach beside using accurate models is to use bounds in feature space as they do not require analytical models for their calculation and basically capture the model mismatch through scattering behavior of the pole clouds. This is discussed in the next section. But the problem with bounds in feature space is that they do not give us a closed analytical formulation which is desirable in optimization of the bounds with respect to sensor parameters.

#### 5.3.2 Feature Based Classifier

Kernel density feature based classifier was reviewed in Section 5.1 which can be written as:

$$\hat{i} = \arg\max_{i} f_i(\hat{p}_i) \tag{5.3.16}$$

Now the Chernoff bound in equation (5.3.6) can be written as:

$$P_{e} \leq M^{-1} \sum_{k=1}^{M-1} \sum_{j=k+1}^{M} Q_{kj}(n)$$

$$Q_{kj}(n) = \min_{0 \leq s \leq 1} \int [f_{k}(p)]^{s} [f_{j}(p)]^{1-s} dp = \min_{0 \leq s \leq 1} D_{kj}(s)^{n}$$
(5.3.17)

where the probability density functions are estimated using kernel density estimation as explained in Chapter 3 which integration is done in three dimensional pole space.

For simulation, the responses of two targets are simulated for a GEM3 sensor [34] sampled on a one meter square area using an equally spaced  $5 \times 5$  grid of measurement points. The objects include: a three inch long by one inch diameter stainless steel cylinder (S1) and a six inch long by one inch diameter stainless steel cylinder (S2). The target responses were



Figure 5.11: Pole distributions for S1 and S2 steel objects at different SNR

generated with a four pole per axis dipole model in which the sums in (2.1.3) are terminated after four terms [8,10]. The poles used for this simulation were generated using the methods in [8]. Figure 5.11 shows the distribution of poles as a function of signal to noise ratio. Chernoff bound for the same targets as a function of signal to noise ratio are calculated using 1000 samples for each target generated using random noise and random orientation angel in order to generate different realization of PDF cloud. Monte Carlo simulation were performed using constructed libraries with 100 runs for each object. Figure 5.12 shows Chernoff upper bound and Monte Carlo simulations using KDE classifier. Comparing Figure 5.12 for feature based classifier with that of corresponding upper bound for data space classifier in Figure 5.6 we see that upper bound for data space classifier is lower than feature based classifier. For



Figure 5.12: Error probability upper bound for feature space classifier

example at 35dB error upper bound for feature based classifier is about 20% while for data space classifier is about 5%, around 75% lower than data space classifier.

#### UXO vs. Clutter error analysis

Feature based classifiers explained in Section 4.2.3 can be used to discriminate between clutter and UXO objects. Clutters are metallic objects that are not UXOs. Because of their random shapes they usually do not scatter like dipoles. One approach for UXO vs. clutter detection is to add a null hypothesis  $H_0$  to our statistical classification model in Section 4.2 which corresponds to the hypothesis that the current object is not in our library. Using this approach we can include a threshold test in our hypothesis tests before applying comparing different hypothesis [17]. Assume that our statistic is  $T(\hat{\theta})$  then we will declare the object as clutter if  $T(\hat{\theta}) < \tau$  where  $\tau$  is a threshold corresponding a detection and false alarm rate.

Another approach is to use just feature space and combine all UXO pole clouds for each

target together to represent the UXO pole cloud. Then we will assume a uniform probability distribution over clutter. Now using feature space Chernoff bound we are able to compute upper bounds on error probability of detection of UXOs from clutter. Figure 5.13 shows pole clouds of UXOs targets combined together including M42, 90mm, 76mm and BLU26 targets. We assumed a uniform distribution for clutter starting at origin in a box shape domain, then we increase the limits of the uniform box up to twice its original length. This corresponds to a more distributed clutter in the space. The edge of the box starts at [2000, 2000, 1500](*Hertz*) and extends to [4000, 4000, 3000](*Hertz*) Now we can evaluate feature based Chernoff upper bound using Equation 5.3.2. Figure 5.14 shows the upper bound as a function of uniform box bounds. Parameter  $\lambda = 0$  corresponds to [2000, 2000, 1500](*Hertz*) and  $\lambda = 1$  corresponds to coordinate [4000, 4000, 3000](*Hertz*). Figure 5.14 shows that the more the clutter is spread out over a wider area in the space, the lower is the error probability after some limits determined by the range of UXO clutter. Up to that limit (which corresponds to  $\lambda = 0.3$  for our case we have increasing error probability. This is because up to that limit the error probability is mostly determined by uniform distribution which makes the integral increasing.



Figure 5.13: Pole clouds of combined UXO targets and uniform clutter distribution bounds



Figure 5.14: Error probability upper bound for UXO vs. clutter for uniform clutter PDF

# Chapter 6 Towards Sensor Optimization

In previous chapter we derived analytical bounds on error performance of optimal classifiers. In this chapter we try to optimize this error bound (i.e. minimize error upper bound) with respect to sensor parameters like spatial sampling, sensor dimensions and sampling frequencies. This will lead us to the idea of sensor optimization were we are willing to have sensors that are tuned for best discrimination performance. Optimal sensor design has been considered extensively in recent year using theory of optimal experiments [19] [40] to mitigate irrelevant information gathered by a sensor in order to utilize system resources. In [41], the authors used theory of optimal experiments to have a UXO sensor system for which minimizes variance of estimated model parameters. As the ultimate goal of UXO detection and classification is to have minimum misclassification error, In this thesis we use upper bounds on error probability developed in Chapter 5 to synthesize sensors that have the optimal performance. First we will review upper bound on error probability of optimal Bayesian classifier and some properties of it, then we will investigate the effect of sensor parameters on the upper bound.

## 6.1 Upper Bound on Error Probability

In Section 5.3.1, upper bound on error probability for optimal Bayesian classifier was derived using Laplacian approximation and asymptotic analysis. The optimal classifier can be written as:

$$\hat{i} = \arg\max_{i} \int f(y/H_i, \theta) f_i(\theta) d\theta.$$
(6.1.1)

The upper bound on pairwise error probability (in natural logarithm scale) of this classifier using Equation (5.3.12) and asymptotic approximation can be written as:

$$\ln Q_{kj}(n) \approx \frac{d}{2}\ln(\pi) - \frac{1}{4}\ln(det[J(\theta_k]) - \frac{1}{4}\ln(det[J(\theta_j)]) + \frac{1}{2}\ln(f_k(\theta_k)) + \frac{1}{2}\ln(f_j(\theta_j)) - \sum_{i=1}^n \left\{ \frac{1}{4} \frac{(m_i(\theta_j) - m_i(\theta_k))^2}{\sigma_{ij}^2 + \sigma_{ik}^2} + \frac{1}{2}\ln\frac{|\sigma_{ij}^2 + \sigma_{ik}^2|}{2|\sigma_{ij}||\sigma_{ik}|} \right\}$$
(6.1.2)

and the total upper bound can be calculated using union bound:

$$P_e \le M^{-1} \sum_{k=1}^{M-1} \sum_{j=k+1}^{M} Q_{kj}(n).$$
(6.1.3)

In Equation (6.1.2) it is assumed that sensor parameters are fixed. We can include sensor parameters such as spatial sampling scheme, sensor dimension or sampling frequencies in this bound to explore the effect of these parameters on the upper bound. We denote sensor parameter as  $\zeta$ , then the pairwise upper bound can be written as:

$$\ln Q_{kj}(n,\zeta) \approx \frac{d}{2}\ln(\pi) - \frac{1}{4}\ln(\det[J(\theta_k,\zeta)]) - \frac{1}{4}\ln(\det[J(\theta_j,\zeta)]) + \frac{1}{2}\ln(f_k(\theta_k,\zeta)) + \frac{1}{2}\ln(f_j(\theta_j,\zeta)) - \sum_{i=1}^n \left\{ \frac{1}{4} \frac{(m_i(\theta_j,\zeta) - m_i(\theta_k,\zeta))^2}{\sigma_{ij}^2 + \sigma_{ik}^2} + \frac{1}{2}\ln\frac{|\sigma_{ij}^2 + \sigma_{ik}^2|}{2|\sigma_{ij}||\sigma_{ik}|} \right\}$$
(6.1.4)

and the total error upper bound can be written as:

$$P_e(\zeta) \le M^{-1} \sum_{k=1}^{M-1} \sum_{j=k+1}^{M} Q_{kj}(n,\zeta).$$
(6.1.5)

In equation (6.1.5), J function is fisher information matrix and for Gaussian noise assumption can be written as:

$$J(\theta,\zeta) = \frac{\partial m\left(\theta,\zeta\right)}{\partial \theta} C^{-1} \left(\frac{\partial m\left(\theta,\zeta\right)}{\partial \theta}\right)^{H}$$
(6.1.6)

where C is assumed to be a positive definite diagonal matrix so we will have:

$$J(\theta,\zeta) = C^{-1} \frac{\partial m(\theta,\zeta)}{\partial \theta} \left(\frac{\partial m(\theta,\zeta)}{\partial \theta}\right)^{H}.$$
(6.1.7)

It can be seen from (6.1.7) that J is a Hermitian positive definite matrix provided that  $\frac{\partial m(\theta,\zeta)}{\partial \theta}$  is not singular <sup>1</sup>.

<sup>&</sup>lt;sup>1</sup>This condition is usually satisfied as long as we don't have two equal pole values, otherwise we can impose non-singularity by adding a small positive definite matrix  $\epsilon I$ 

If A is  $n \times n$  positive definite matrix with eigenvalues  $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ . Denote the set of all eigenvalues by  $\lambda(A)$ , trace of A by  $\mu_1 = tr(A)$  and square of Euclidian norm by  $\mu_2 = ||A||_F^2$ . The parameters  $\alpha$  and  $\beta$  denote the bounds for the smallest and largest eigenvalues  $\lambda_1$  and  $\lambda_n$  of A,  $0 < \alpha \leq \lambda_1, \lambda_n \leq \beta$ . Using eigenvalue decomposition and the definition of matrix function [42], it can be shown that we have:

$$\ln(\det(A)) = tr(\ln(A)) \tag{6.1.8}$$

In [43] it is shown that:

$$tr(\ln(A)) \le \begin{bmatrix} \ln \beta & \ln \bar{t} \end{bmatrix} \begin{bmatrix} \beta & \bar{t} \\ \beta^2 & \bar{t}^2 \end{bmatrix} \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}$$
(6.1.9)

where

$$\bar{t} = \frac{\beta \mu_1 - \mu_2}{\beta n - \mu_1}.$$
(6.1.10)

If A is Hermitian positive definite matrix, denote the singular value decomposition of A as  $USU^T$ , where U contains eigenvectors of A, then to compute  $\ln(A)$  we can use the following equation:

$$\ln(A) = U\ln(S)U^T \tag{6.1.11}$$

which is again a Hermitian positive definite matrix, so  $tr \ln(A)$  can be written as the sum of norms of the rows of  $U(\ln(S))^{\frac{1}{2}}$  or equivalently square Frobenius norm (Euclidean norm) of  $U(\ln(S))^{\frac{1}{2}}$ , denoted as  $\mu_2 = \|U(\ln(S))^{\frac{1}{2}}\|_F^2$ . So we will have:

$$\ln(\det(A)) = tr(\ln(A)) = \|U(\ln(S))^{\frac{1}{2}}\|_F^2$$
(6.1.12)

Equation (6.1.9) shows that for Hermitian positive definite matrix A,  $\ln(\det(A)) = tr(\ln(A))$ is bounded. Equation (6.1.12) shows that it can written as a sum of square function. Therefore for Hermitian positive definite matrix A,  $\ln(\det(A))$  can be written as a sum of square functions.

Turning back to Equation (6.1.7), from previous discussion we can conclude that  $J(\theta, \zeta)$ can be written as sum of square functions in terms of our model  $m(\zeta)$ . Also  $\ln(f_i(\theta, \zeta))$  terms in (6.1.5) can be written in term of square function in the form of  $\sqrt{-\ln(f_i(\theta, \zeta))}^2$ . The terms  $\frac{(m_i(\theta_j,\zeta)-m_i(\theta_k,\zeta))^2}{\sigma_{ij}^2+\sigma_{ik}^2}$  are already in the form of square function. Therefore the upper bound on error probability derived in Chapter 5 is well suited for optimizations using nonlinear least square minimization methods [26] and we can apply optimization techniques to minimize the upper bound with respect to sensor parameter  $\zeta$  as will be explained in next Section.

## 6.2 Minimization of Error Upper Bound

In this Section we use optimization algorithms to minimize the upper bound with respect to sensor parameters  $\zeta$ . The problem can be formulated as:

$$\zeta_{\text{opt}} = \arg\min_{\epsilon} P_e(\zeta) \tag{6.2.1}$$

where  $P_e(\zeta)$  is computed using Equation (6.1.5)

#### 6.2.1 Optimal Sensor Dimension

To see the effect of sensor dimension on error probability, we used EM63 time domain sensor, the spatial sampling for sensor configuration is shown in Figure 6.1 which corresponds to 441 spatial samples and 26 time domain sample resulting in 11466 data samples. The objects are 105mm, 81mm, 60mm and 40mm mortar shells buried at 60cm under ground in xy plane at 0°, 90°, 0° and 90° orientations respectively. The SNR is about 30dB. EM63 sensor has two square shape coils for transmitter and receiver. The nominal sensor length is 1(m).

Figures 6.2 to 6.5 show collected spatial signal at time gate 5 for each object as the sensor dimensions increases.<sup>2</sup> We note that as the sensor dimension increases the distance between peaks of the signal which are relevant to classification performance start to increase, in other words in as the sensor dimension increases beyond some limit we can not see the whole target in the scene. Figure 6.6 shows the corresponding error upper bound for theses targets as the sensor dimension increases. It can be seen from the figure that error upper bound is minimum at 2.6mm, also it can be seen that increasing sensor dimension form 1mm to 1.5mm reduces the error upper bound from  $10^{-1}$  to  $10^{-3}$ . A reduction of about 100 times, of course with the

 $<sup>^{2}</sup>$  for simplicity of plots we just show the case where the transmitter and receiver have the same size

expense of building a bigger sensor. To explore this further we denote  $\zeta = \begin{bmatrix} l_{Tx} \\ l_{Rx} \end{bmatrix}$ , where  $l_{Tx}$  and  $l_{Rx}$  denote sensor transmitter and receiver length. As the number of variables is low here we can plot upper bound (in log scale) as a function of sensor dimensions. Figures 6.7 and 6.8 show the results. We see that the upper bound has a valley shape and achieves a minimum around  $l_{Tx} = l_{Rx} = 2.8(m)$ . Intuitively it tells us that neither very small sensors that resemble dirac function, nor very large sensor are useful for discrimination.

Optimization of sensor dimension may not be a very feasible task as the sensors are already manufactured, but as we stated in previous paragraph the optimal sensor size is about the same size of the sensing area. So corresponding to each sensor size there is an optimal sampling dimension, this factor is totally under control and feasible in practice and can be used for optimal system configuration design.

Optimal sensor dimension approach can be used by manufacturers to build high performance sensors. To that end error upper bound is not the only factor that influences the design of the sensor, but cost of design plays an important role. For example assume that cost of building a sensor increases with sensor dimensions as a linear function, we can model this cost as  $C(l_{T_x}, l_{R_x}) = \kappa(L_{T_x} + l_{R_x})$ , where  $\kappa$  denotes the cost rate, Now we can add this cost function to error upper bound to find new set of optimal sensor dimensions satisfying both low cost and low error upper bound criteria. The new cost function can be written as:

$$C_T(l_{T_x}, l_{R_x}) = \kappa(l_{T_x} + l_{R_x}) + P_e(l_{T_x}, l_{R_x})$$
(6.2.2)

Figure 6.9 shows the new cost function as a function of sensor dimension with  $\kappa = \$10$  we see that the optimum sensor size is now around 1.42(m).



Figure 6.1: Sensor spatial sampling configuration

### 6.2.2 Optimal Dimension of Sensing Area

As stated in previous section changing sensor dimension may be a difficult task if not impossible. In this section we see the effect of length of sampling area (assuming a square grid) on the performance of the system. As can be seen from Figure 6.1 the sensor samples at a rectangular grid from .5 to 2.5 (m) in x axis direction and from 0 to 4 (m) in y direction which results in a  $2 \times 4$  (m) sampling rectangle. The object is located at the center in location [1.5, 2]. To see the effect of this dimension we use a sensor with  $l_{Tx} = l_{Rx} = 1$ m and sample with different sensing dimensions where the object is located at the center and sampling is done in a square grid of variable length starting from 2cm square to 16 m with the number of samples kept constant to 441 samples on a 21 × 21 square grid. This will correspond to sampling spaces from 1cm to 77cm <sup>3</sup>. Figures 6.10 and 6.11 show the upper error bound as the function of sampling dimension (length square sampling area). Interestingly we again see that we have an optimal value for the sampling dimension. Which in this experiment is 1.24m corresponding to sampling space of 6 cm, resulting in error of 0.03. Comparing this with the nominal sensor

 $<sup>^{3}</sup>$ It should be noted that minimum sampling space that is feasible is 5 cm, as we will see that optimal value will be in the feasible range.



Figure 6.2: Collected data as the sensor size changes for 105mm mortar shell



Figure 6.3: Collected data as the sensor size changes for 81mm mortar shell



Figure 6.4: Collected data as the sensor size changes for 60mm mortar shell



Figure 6.5: Collected data as the sensor size changes for 40mm mortar shell



Figure 6.6: Error upper bound as the sensor size increases



Figure 6.7: Error upper bound as a function of sensor dimensions for four mortar shell objects



Figure 6.8: Error upper bound as a function of sensor dimensions



Figure 6.9: Performance upper bound as a function of sensor dimensions considering sensor cost



Figure 6.10: Error upper bound as a sensing dimension



Figure 6.11: Error upper bound as a sensing dimension

used for field data with  $2 \times 4$  (m) sampling rectangle which has error upper bound of about .12 we see that using this configuration we can reduce error upper bound by a factor of 4. Intuitively very small sampling dimension does not captures object's signal well. As in our case we have an 105mm mortar shell which is about 10cmm long. Also very large sampling dimension (not that we have the same number of samples for both case) does not result in informative samples because most outer samples would be zero. Figure 6.12 shows how the sensor views the object as we sample over a larger space with the same number of samples. As we see as the space of sampling gets larger we will have less informative signal samples which are almost zero. The optimal dimension chooses the sampling frame for best performance by focusing on the best view in terms of signal strength. As explained before changing sampling space is a very feasible task and can boost performance by a large amount.



Figure 6.12: Signal view as the sensor sampling space grows larger





Figure 6.13: Mortar shell objects configuration

As we discussed in previous Section, it is desirable to be able to optimize sensor discrimination capability using parameters that are easy to work with. Sensor spatial sampling is a parameter that is most under control during sensing operation. Therefore in this Section we seek to find spatial sampling schemes that minimize error upper bound. Again as before we denote spatial samples as parameter vector  $\zeta = \begin{bmatrix} l_1 & l_2 & \cdots & l_M \end{bmatrix}$  corresponding to M samples in two dimensional space. Now we use Equation (6.2.1) to find the optimal sampling scheme. We used MATLAB *fmincon* function to to minimize the upper bound with respect to sensor dimensions. The function uses sequential quadratic programming(SQP)method. In this



Figure 6.14: Optimum sampling scheme for system configuration in Figure 6.13

method, a Quadratic Programming (QP) subproblem is solved at each iteration. An estimate of the Hessian of the Lagrangian is updated at each iteration using the BFGS formula [44]. A line search is performed using a merit function similar to that proposed in [45]. Figure 6.14 shows optimal sampling schemes for 105mm ,81mm, 60mm and 40mm for configuration in Figure 6.13 , indicating initial normal square sampling grid in red circles and final optimized sampling in black crosses. The starting error corresponding to normal square grid was .114, after optimization the resulting sampling results in error upper bound of .0211. About one fifth of normal square grid. Figure 6.15 shows the optimal sampling scheme for the same object with a 5 × 5 grid. The starting error upper bound was .1156 and after optimization the resulting upper bound achieved was .0037. The final result for this case is much lower that the 10 × 10 grid. This is mainly because of complexity of optimization where in low sample grid the optimization algorithm achieves the optimal value for the same number of iterations where for high sampling grid we need to have more iterations to achieve the optimal value. We note that optimization of Equation (6.1.5) results in maximizing the term  $\frac{(m_i(\theta_{j,i})-m_i(\theta_k,\zeta))^2}{\sigma_{i,i}^2+\sigma_{i,k}^2}$ , which



Figure 6.15: Optimum sampling scheme for system configuration in Figure 6.13

intuitively tell us to choose sampling scheme which maximizes the euclidian distance between two objects data in M dimensional space. From Figure 6.14 it can be seen that the optimal schemes basically samples the places where we have maximum differential signals which corresponds to values corresponding to peak values in objects response. For our case the larger targets 105mm and 81mm have higher peaks therefore the optimal samples are basically over peak signal areas for two larger targets.

# Chapter 7 Sensor Positional Uncertainty

Having developed and analyzed UXO classification methods in previous chapters, we will quantify and analyze the issue of sensor signal processing in the presence of sensor positional uncertainly in this chapter. Up to now we assumed that positions of sensor are known precisely with respect to some know reference, a condition which is often not met in practice [46,47]. For example in field data processing the sensor position is governed by a not precise instruments. The problem of sensor positional uncertainly can be divided in two main folds, first developing uncertainly model and second processing methods to handle uncertainty in the model. To this end in this section we have developed an uncertainty position model using Markov chain/ random walk model which captures spatial dependency of error locations and developed new robust estimation algorithms with low computational cost using min-max approach. First we will review the problem and then introduce the model and processing methods.

## 7.1 Problem Formulation

As explained in Section 2.1, we are using an approximation model (dipole model) for physics of the underlying sensing mechanism. The model is parameterized in term of two sets of quantities. First a collection of unknowns in the vector  $\theta$  that are of primary concern in the processing. These are the relevant to solve some underlying estimation, detection, or classification problem. Second a high dimensional collection of nuisance parameters associated with the positioning of the sensor (or more general uncertain structure in the sensor itself). Concentrating on the issue of positional uncertainty, let us assume that the sensor collects data a a total of N locations. The nominal location of the sensor at the *i*th such stop is denoted by

$$r_{i,0} = \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix}.$$
(7.1.1)

The true location of the sensor is not the same as the nominal location, but is perturbed a bit so that:

$$r_{i} = r_{i,0} + \delta_{i} = \begin{bmatrix} x_{0} \\ y_{0} \\ z_{0} \end{bmatrix} + \begin{bmatrix} \delta_{x,i} \\ \delta_{y,i} \\ \delta_{z,i} \end{bmatrix}.$$
 (7.1.2)

We assume that  $\theta$  is decomposed into two sub-vectors as explained in Section 2.1. We model the first, p, as a random vector (poles) with probability density function f(p). We model the second,  $\nu$ , as a collection of non random parameters (spatial information plus coefficient factors and DC value) for which we have no such prior model. To model the uncertainty quantities we might want to remain within the framework we have been using [46], which treats these quantities in a bounded, but unknown manner; i.e. no probability. The simplest model would be to bound each elements of the  $\delta_i$  vectors as:

$$\underline{\delta} < \delta_{\alpha,i} < \overline{\delta} \qquad \alpha \in \{x, y, z\}.$$
(7.1.3)

This though may not be sufficient since we know that as the sensor collects data from one location to the next, the positional errors are related. At the same time, such errors do not grow without bound. Sensors more or less stay close to the course they are supposed to follow. This means specifically that a random walk-type model would not be appropriate for this problem. Or equivalently we want to have:

$$\|\delta_i\| < \epsilon_{1,i} \text{ and } \|r_i - r_{0,i}\| < \epsilon_{2,i}.$$
 (7.1.4)

The first bound indicates that we cannot move too far from one step to the next. The second bound is used to enforce the idea that the overall error in positioning is always bounded. Finally, we assume that the perturbations in position are always sufficiently small to allow us to consider a model linear in these parameters. Thus, the overall model takes the form:

$$y = A \begin{pmatrix} \theta, & r_{0,1}, & \cdots & r_{0,N} \end{pmatrix} \begin{bmatrix} r_1 - r_{0,1} \\ \cdots \\ r_N - r_{0,N} \end{bmatrix} + \delta y + w.$$
(7.1.5)

Here y is the vector of data, A is the Jacobian matrix used to linearized the exact model (such as the dipole) about the nominal sensor locations, w is zero mean additive white Gaussian noise with an appropriate covariance matrix and  $\delta y$  is a vector of perturbations the fact that the e.g. dipole model is not exact. Here we assume that this quantity is also bounded:

$$\|\delta y\| < \epsilon_3 \tag{7.1.6}$$

In the following section we propose uncertainty position model using Markov chain/ random walk model which captures the spatial dependency of error locations as well as the bounded condition on errors and developed new robust estimation algorithms with low computational cost using min-max approach.

### 7.2 Markov Chain/ Random Walk Model

Here we propose uncertain position model that accounts both spatial dependence and bounded condition on error locations. The perturbation is modelled as a random walk on a graph, where each state corresponds to a spatial perturbation and transition weights correspond to probability that the particle (perturbation in our case) moves to the next position. Using probabilities at each node we can impose the condition where it is more probable that the perturbation in the next stage stays within its previous neighbors. The graph approach also takes into account bounds on the perturbation automatically so the perturbation will always stays within the limits. Figure 7.1 shows a simple one dimensional version of the graph. For simplicity only additive perturbations  $\delta_i$  at each location is shown so for processing we will have  $r_i = r_{0,i} + \delta_i$ . which means at each location, perturbation can be found from the graph and will be added to the nominal value of the corresponding sensor location. Being able to impose conditions on perturbation spatial movement is very desirable for example assume that



Figure 7.1: Random walk/Markov chain model for sensor position uncertainty

we want that error stays within its previous neighbor, then it is sufficient to change probability transition matrix so that the particle at each node can be moved to its neighbor location this is shown in Figure 7.2. For another example assume that we want that error has a tendency to go to zero, then it is sufficient to amplify path that converge to origin. This is in the control of the designer considering real field scenarios. Markov chain/random walk model also takes into account bounds in equation (7.1.4) as the both errors and location changes are bounded within limits of  $\delta$ . Another nice property of the model is that the model is well suited for dynamic processing algorithms like Viterbi algorithm [48] which have low computations and easy to implement. This will be explained in the next section.

To demonstrate application and flexibility of random walk on a graph to our model, we simulated a two dimensional random walk on a graph using above model. This corresponds to perturbation at each sensor location assuming we have errors in two dimensional space. Figure 7.3 shows a random walk were the transition matrix was chosen so that each new error is confined to be within it 9 points neighbors in plane with equal probability. Figure 7.4 shows a random walk were the transition matrix was chosen so that each new error is confined to be within it 9 points neighbors in plane with equal probability. Figure 7.4 shows a random walk were the transition matrix was chosen so that each new error is confined to be within it 9 points neighbors in plane with equal probability. Figure 7.4 shows a random walk were the transition matrix was chosen so that each new error is confined to be within it 9 points neighbors in plane with more probability to stay in it previous position. It can be seen that the later is smoother, as we imposed more memory on the walk. Again we



Figure 7.2: Imposing neighborhood condition



Figure 7.3: 2D random walk with equal probability neighborhood condition



Figure 7.4: 2D random walk with high correlation with previous position



Figure 7.5: Nominal and perturbed sensor grid



Figure 7.6: Sensor sampling path

note that at each position step only error is plotted and the actual sensor position is computed by adding the errors to corresponding nominal position vectors.

Figure 7.5 shows a real field example of sensor sampling grid, the samples are taken in a zigzag manner starting shown in figure 7.6. The grid spacing for nominal sampling is 22cm. Perturbations were generated according to our model on a square of size 2cm with 121 points which corresponds to errors of step size 2mm. It can be seen that error locations are correlated on the path of sampling. For simulations we will use such a random walk in two or three dimensional plane.

## 7.3 Min-Max Estimation Using Viterbi Algorithm

Estimation algorithms explained in Chapter 4 in the presence of uncertainty in sensor position can be implemented using Min-Max approach [27]. In this method we will extract the parameters in the worst case condition on cost function with respect to uncertain parameters. Using equation 4.1.4, Min-Max estimation can be written as:

$$\hat{p}, \hat{\nu} = \arg\min_{p,\nu} \left\{ \arg\max_{\delta} \left\{ \sum_{i=1}^{N} \|y_i - m_i(p,\nu,\delta_i)\|_2^2 \right\} + \alpha \left(\sqrt{-\ln f(p)}\right)^2 \right\}.$$
(7.3.1)

Where  $\delta = (\delta_1 \ \delta_2 \ \cdots \ \delta_N)$  and  $\alpha$  balances the effect of prior factor and noise variance, setting  $\alpha = 0$  means we are not including any prior in our processing.

The Max part of Min-Max approach is:

$$\hat{\delta} = \arg\max_{\delta} \sum_{i=1}^{N} \|y_i - m_i(p, \nu, \delta_i)\|_2^2$$
(7.3.2)

from above equation and trellis in Figure 7.1 it can be seen that the max part is equivalent to finding the shortest path (for negative cost function) on the corresponding trellis. This is a well know problem in communication and dynamic programming [48], where Viterbi algorithm is usually used as a solution because of its low computation and optimal nature. Using equation 7.3.2 it can be seen that the total cost at (j+1)th,  $T_{j+1}$  is a recursive function of partial costs  $C_{j+1}$  and can be written as:

$$T_{j+1} = \sum_{i=1}^{j+1} \|y_i - m_j(p,\nu,\delta_i)\|_2^2 = \sum_{i=1}^j \|y_i - m_j(p,\nu,\delta_j)\|_2^2 + \|y_{j+1} - m_{j+1}(p,\nu,\delta_{j+1})\|_2^2 = T_j + C_{j+1}.$$
(7.3.3)

In Viterbi algorithm [49], each cost metric  $T_i$  entering a state in trellis is calculated and only the path with maximum metric is stored which is known as survivor path. If the trellis has |Q| states and length N, then time complexity of Viterbi algorithm is  $O(N.|Q|^2)$  in contrast to step by step decoding which is an exponential function of N and |Q|. Usually to overcome the long delay of optimal algorithm a window about five times length of trellis memory is used to estimate the optimum states. To account for probability transition matrix we use a modified version of Viterbi algorithm where we weight each partial metric  $C_i$  by corresponding probability path to take into account the weights of trellis.

## 7.4 Numerical Results

In order to see the effect of uncertain sensor position on the performance of classification algorithms, Monte Carlo simulations are used, sensor and objects were choused as explained in Section 4.3. samples are collected using an equally spaced  $5 \times 5$  grid of measurement points with path similar to Figure 7.6 and 25cm spacing. Perturbations are generated using the

Mahalanobis classifier							
	S1	S2	A1	A2			
S1	27	20	39	14			
S2	21	20	41	18			
A1	0	1	92	7			
A2	0	0	1	99			

KDE						
	S1	S2	A1	A2		
S1	52	48	0	0		
S2	54	46	0	0		
A1	2	14	84	0		
A2	0	0	0	100		

Table 7.1: Confusion matrices in presence of sensor position uncertainty at SNR of 30 dB

Mahalanobis classifier

A2 || 0

	S1	S2	A1	A2	
S1	32	29	31	8	S
S2	36	34	17	13	S
A1	0	0	94	6	Α
A2	0	0	2	98	A

KDE S1S2A1 A25842 1 0 0  $\overline{2}$ 46540 0 1 9 0 0 91  $\overline{2}$ 100 0 0 0

Table 7.2: Confusion matrices in presence of sensor position uncertainty at SNR of 30 dB using Min-Max approach

model explained in Section 7.2 corresponding to a 10cm square uncertainty and 25 points. This will result in a trellis of length 25 with 25 states. As each perturbation is assumed to be in the neighbor of previous error, the transition matrix is a  $25 \times 25$  sparse matrix with at maximum 6 nonzero elements in a row. This lowers the computational complexity of Viterbi algorithm to one forth, as we just can compute 6 metrics for each state instead of 25 metrics at each iteration.

Table 7.1 shows classification results in the presence of sensor positional uncertainty using Mahalanobis and KDE classifiers (see Chapter 4) at SNR of 30dB. Comparing this results with no sensor positional uncertainty in Table 4.2 we notice about 50% increase in misclassification error for both classifiers. Table 7.2 show classification results using Min-Max approach explained in Section 7.3. Comparing Tables 7.1 and 7.17.2 reveals improvement in classification in the presence of sensor positional uncertainty where there is about 20% improvement in KDE classifier and about 14% in Mahalanobis classifier using Min-Max approach.

# Chapter 8 Conclusion and Future Work

To achieve robust methods for classification of UXO, a collection of classification methods based on the use of nonparametric prior models for target features were proposed and compared. Kernel density estimation methods formed the basis for the construction of these prior models. These priors allowed for the development of a collection of MAP-based estimation and classification schemes. Also considered were GLRT-type methods, approximation to the statistically optimal Bayes classifier for our problem, and an SVM method.

In addition to the processing methods discussed here, powerful performance analysis methods incorporating these prior models is used to bound the performance of both the estimation as well as classification methods. More specifically, Bayesian Cramer-Rao lower bounds is developed which lower bound the variance in our estimates of the target parameters. Similarly, the Chernoff bound provides an upper bound on the probability of error for the Bayes-optimal feature-based as well as data-based classifiers. These bounds were derived analytically and confirmed using Mont-Carlo simulation. Viewing these bounds as a function of system parameters, we have optimized these bounds in terms of those parameters which will lead us to sensors that are tuned for best discrimination performance.

Finally the problem of sensor signal processing in the presence of sensor positional uncertainty was investigated, where a new model using fusion of Random walk and Markov chain was introduced to model the effect of correlated perturbations. Then a low complexity Min-Max approach using Viterbi algorithm was developed to estimate the feature parameters in the presence of sensor positional uncertainty. simulation results indicate significant improvement when we use Min-Max approach. The framework developed in this thesis can be applied to any processing model.

The future work would be including more accurate UXO models and using our performance analysis tools to evaluate their performance which will result in a comprehensive study of performance of different models. Developing more accurate estimation algorithms for estimation of object orientation using quaternion theory or using shaped based models that are useful for extracting objects geometry and then use this information in our processing algorithms to mitigate the effect of unknown geometry on estimation and classification algorithms. Using models that account shape and material of the object to develop classification algorithms. Utilizing a more accurate set of field data to train and to evaluate performance of classification algorithms.

The problem of sensor optimization can be done also using pole clouds utilizing Monte Carlo simulation. We note that using this approach we are unable to use analytical bounds, but this also will provide us insight of how the system parameters affects feature space classifiers.

For more advanced sensors where we are able to use several transmitter and receiver at different location and illumination angles, another idea would be using sensor fusion theory to develop estimation and classification algorithms when we have these multiple sensors, also the idea of sensor optimization can be used in the case of multiple sensors with several transmitter and receivers to see what are the best configuration for best performance and how to combine these sensors with each other. For the sensor positional uncertainty topic we may be interested to derive analytical performance of the system in the presence of uncertainty by deriving CRLBs in the presence of model uncertainty and generalizing Chernoff upper bound for the case of model uncertainty.

Another topic which might be interesting specially for field data is that we have a limited number of training samples which are usually taken arbitrary, using information theory and analytical bounds derived here we might be able to see what specific samples are more useful for the purpose of training so that we have the most informative training data with low number of samples. Markov model presented in the last chapter for uncertain sensor position has the
potential to be trained from real data to find the transitional property of the trellis, this can be done by gathering data from field and train the model using these data and then use this trained model for real data processing.

# Appendix A EMI Models

In this Appendix, we first consider an EMI model that is a combination of the physical EMI model in [50], describing scattering by spherical or spheroidal targets, with the model in [8], which justifies the modelling of the EMI response of a conducting, permeable target as a sum of decaying exponentials in time, or one-pole models in frequency. We also consider the implementation of several alternative models, including the mean internal field model developed by Peter Weichman [51, 52], and a variant of the dipole model that includes conventions for handling the power law nature of the early-time decay of a time domain EMI response.

# A.1 The Dipole Model

The emf induced in a target T located at  $\mathbf{r}_0$  by a transmitter is given by

$$\varepsilon_{T,TX} = -j\omega \int \mathbf{B}_{TX}(\mathbf{r}_0) \cdot d\mathbf{A}_T.$$
 (A.1.1)

Following [53], we we consider the target to be an x-directed, filamentary loop, allowing the induced emf to be simplified as  $\varepsilon_{T,TX} = -j\omega B_x A = -j\omega \mu_0 H_x A$ . It has also been tacitly assumed in the previous equation that the target is small enough and/or far enough from the transmitter that the transmitted field is constant over the target. The frequency-domain current induced in the filamentary loop by the transmitter is therefore

$$I = [R + sL]^{-1}\varepsilon = [R + sL]^{-1}(-j\omega\mu_0 H_x A),$$
(A.1.2)

where R and L are the resistance and inductance of the loop. The moment of the induced current is

$$m_x = IA = [R + sL]^{-1}(-j\omega\mu_0 H_x A^2) = P_x H_x.$$
(A.1.3)

Alternatively, moment may be written as the product of target magnetic polarizability and the incident magnetic field. Rearranging, we have an equation that describes the polarizability of an x-directed filamentary loop

$$P_x = a_x \frac{s}{p_x + s},\tag{A.1.4}$$

where  $a_x$  is an expansion coefficient related to the the area and inductance of the loop, and  $p_x$  is the pole value that is characteristic of the loop, where  $p_x = R/L$ . A metal target may be modeled as consisting of a large number of filamentary loops directed along each principal axis. Following the reasoning above, allowing that the loop impedances are altered via mutual inductance, a moment may be defined for each loop that is the sum of the individual loop moments. By extension, a magnetic polarizability may be defined for each axis *i* as

$$P_i(s) = \sum_{l=1}^{\infty} a_{i,l} \frac{s}{p_{i,l} + s}.$$
 (A.1.5)

We must also consider the case of a ferrous target, as many UXO contain iron or steel components. We therefore introduce a zero-frequency "DC" term to the polarizability defined above:

$$P_i(s) = d_i + \sum_{l=1}^{\infty} a_{i,l} \frac{s}{p_{i,l} + s}.$$
(A.1.6)

The emf generated in the receiver by the magnetic field scattered from the target is given by  $\varepsilon_{RX,T} = -j\omega \int \mathbf{B}_T(\mathbf{r}_{RX}) \cdot d\mathbf{A}_{RX}$ . As in [20], we may invoke the concept of reciprocity to rephrase the emf in terms of the emf generated in the target by a hypothetical current flowing in the receiver:

$$\varepsilon_{RX,T} = \frac{I_T}{I_{RX}} \varepsilon_{T,RX}.$$
(A.1.7)

It is assumed as above that the receiver field is uniform over the target, reducing the integral to a simple multiplication. The emf may thus be written as

$$\varepsilon_{RX,T} = -j\omega\mu_0 \frac{\mathbf{H}_{RX}}{I_{RX}} \cdot \mathbf{m}_T = -\frac{j\omega\mu_0}{I_{RX}} \mathbf{H}_{RX} \cdot P \cdot \mathbf{H}_{TX}.$$
 (A.1.8)

When the target is allowed to assume any orientation described by the rotation matrix R the entries of which are functions of three Euler angles  $\phi$ ,  $\psi$ , and  $\theta$ , the equation above may be rewritten by diagonalizing the polarizability tensor, such that  $P = R\Lambda R$ , where the diagonal elements,  $\lambda_i$ , of  $\Lambda$  contains the principal axis polarizability functions defined in equation A.1.6. The time domain versions of the PAPFs are easily calculated for a step input signal by inverse Fourier transform:

$$\lambda_i(t) = \sum_{l=1}^{\infty} a_{i,l} p_{i,l} e^{-p_{i,l}} u(t),$$
(A.1.9)

where u(t) is the unit step function, and it has been assumed that the first measurement is made after t = 0.

Our model may be succinctly written as

$$r_{n,k} = g_n^T R^T \Lambda_k R f_n + \sigma \omega_{n,k} \equiv s_{n,k} + \sigma \omega_{n,k}$$
(A.1.10)

where there are k = 1, 2, ..., M time or frequency samples from each of n = 1, 2, ..., N combinations of transmitter and receiver positions. The transmitter magnetic field at  $r_0$  is described by the  $3 \times 1$  vector f, and a similar vector, g, describes the hypothetical field of the receiver coil at  $r_0$ . The quantity  $\sigma$  is the standard deviation of the assumed-additive white Gaussian measurement noise, and  $\omega_{n,k}$  is a zero mean, unit variance normal random variable.

The PAPFs are dependent only upon the target size, shape, and material, and not on the orientation and position of the target relative to the sensor. Therefore, the poles and expansion coefficients that are contained in the PAPFs are good candidates for use in a classification routine. Position information in contained in the f and g vectors, and orientation information is conveyed by the rotation matrix R.

Several caveats must be considered, however. Though the model has been validated using real sensor data [9, 50, 53, 54], it should be noted that real targets do not behave exactly as dipoles. Further, the data generally only support the estimation of a single pole per axis [9, 13, 53]. Poles estimated from real data therefore vary with respect to target orientation and position, and classifications schemes must explicitly account for this variation.

### A.2 Sensor-Specific Calibration of the Physical Model

Computation of the response of an actual, real-world EMI sensor requires some modification of the physical model. A realistic sensor transmitter field must be modelled. The receiver antenna and any onboard post-processing must also be accounted for in order to properly model the sensor response. To generically account for most commercial sensor configurations, our code includes the ability to model the magnetic field of both multi-turn circular and rectangular transmitter and receiver coils.

#### A.2.1 Magnetic Field of a Rectangular Coil

From Das, et al. [50], the magnetic field at (x, y, z) of a rectangular coil (required in equation A.1.8 for the fb and g vectors) with sides of dimensions 2A and 2B carrying a current I is

$$H_{x} = \frac{I}{4\pi} (H_{x2} + H_{x4}) \quad H_{y} = \frac{I}{4\pi} (H_{y1} + H_{y3})$$
$$H_{z} = \frac{I}{4\pi} (H_{z1} + H_{z2} + H_{z3} + H_{z4})$$
(A.2.1)

where

$$\begin{split} H_{x2} &= -\frac{z}{(x+A)^2 + z^2} \cdot \left[ \frac{(y-B)}{\sqrt{(x+A)^2 + (y-B)^2 + z^2}} - \frac{(y+B)}{\sqrt{(x+A)^2 + (y+B)^2 + z^2}} \right] \\ H_{x4} &= -\frac{z}{(x-A)^2 + z^2} \cdot \left[ \frac{(y+B)}{\sqrt{(x-A)^2 + (y+B)^2 + z^2}} - \frac{(y-B)}{\sqrt{(x-A)^2 + (y-B)^2 + z^2}} \right] \\ H_{y1} &= +\frac{z}{(y-B)^2 + z^2} \cdot \left[ \frac{(x-A)}{\sqrt{(x-A)^2 + (y-B)^2 + z^2}} - \frac{(x+A)}{\sqrt{(x+A)^2 + (y-B)^2 + z^2}} \right] \\ H_{y3} &= +\frac{z}{(y+B)^2 + z^2} \cdot \left[ \frac{(x-A)}{\sqrt{(x+A)^2 + (y+B)^2 + z^2}} - \frac{(x-A)}{\sqrt{(x-A)^2 + (y+B)^2 + z^2}} \right] \\ H_{z1} &= -\frac{(y-B)}{(y-B)^2 + z^2} \cdot \left[ \frac{(x-A)}{\sqrt{(x-A)^2 + (y-B)^2 + z^2}} - \frac{(x+A)}{\sqrt{(x+A)^2 + (y-B)^2 + z^2}} \right] \\ H_{z2} &= +\frac{(x+A)}{(x+A)^2 + z^2} \cdot \left[ \frac{(y-B)}{\sqrt{(x+A)^2 + (y-B)^2 + z^2}} - \frac{(x-A)}{\sqrt{(x+A)^2 + (y+B)^2 + z^2}} \right] \\ H_{z3} &= -\frac{(y+B)}{(y+B)^2 + z^2} \cdot \left[ \frac{(x+A)}{\sqrt{(x+A)^2 + (y+B)^2 + z^2}} - \frac{(x-A)}{\sqrt{(x-A)^2 + (y+B)^2 + z^2}} \right] \\ H_{z4} &= +\frac{(x-A)}{(x-A)^2 + z^2} \cdot \left[ \frac{(y+B)}{\sqrt{(x-A)^2 + (y+B)^2 + z^2}} - \frac{(y-B)}{\sqrt{(x-A)^2 + (y-B)^2 + z^2}} \right] . \end{split}$$
(A.2.2)

#### A.2.2 Magnetic Field of a Circular Coil

The magnetic field of a circular coil may be computed using a scalar potential formulation. For a loop of radius *a* carrying a current *I*, a scalar potential  $\phi$  exists for regions not containing the source current, or where the radial coordinate *r* is greater than and less than *a*. The potential  $\phi$  satisfies Laplace's equation and  $\mathbf{B} = -\nabla \phi$ . The magnetic field may be calculated by recalling that  $\mathbf{B} = \mu_0 \mathbf{H}$  for a circular coil in free space.

Expanding the scalar potential in Legendre polynomials,

$$\phi(r,\theta) = \begin{cases} \sum_{n} A_{n} r^{n} P_{n}(\cos\theta) & r < a\\ \sum_{n} B_{n} r^{-n-1} P_{n}(\cos\theta) & r > a. \end{cases}$$
(A.2.3)

The coefficients  $A_n$  and  $B_n$  can be found by comparing equation A.2.3 to an analytically calculable potential. The potential along the z-axis, where  $\theta = 0$ , is easily derived from the

Biot-Savart law:

$$\phi(r = z, \theta = 0) = \frac{\mu_0 I}{2} \left( 1 - \frac{z}{\sqrt{a^2 + z^2}} \right).$$
(A.2.4)

Because  $\cos(0) = 1$ , a Taylor expansion of equation A.2.4 in terms of r/a will yield the expansion coefficients of the general scalar potential where r < a, and an expansion in terms of a/r will provide the potential coefficients where r > a.

Although equation A.2.3 provides an analytical functional form for the computation of the magnetic field **H** of a circular coil at point  $(r, \theta)$  on paper, the implementation of this series is less than ideal for rapid computation of the physical model. The series diverges on the sphere r = a if too few terms are used in the expansion, and this region of divergence may be coincident with the target burial depth. Thus, we have found it necessary to pre-compute the potential for a suitable range of r and  $\theta$ , for  $r \neq (a \pm q)$ , where q is an empirically chosen distance from a at which the series divergence is minimal. The pre-computed potential is then splined, so that the spline function in essence bridges the excised poorly-behaved portion of the truncated series.

# Appendix B Data Survey

In this appendix, different types of field data sets and processing methods used to extract relevant parameters from those data sets are explained in more detail. At the recommendation of SERDP, datasets used in this thesis were acquired from Mr. Jay Bennett and Mr. Cliff Morgan at the WES test site. These data were collected using the ERDC test stand, for a complement of 18 different UXO and several clutter items. The UXO were oriented horizontally or vertically, and at a single azimuth angle. The sensor was positioned in one of eight different grids, dependent upon the type and size of the target in question. Background data were also collected.

An initial survey of the data reveals that the data appear to be clean, with the exception of some unusual peaks near the edges of the grid. These peaks are probably due to the presence of a conductor in the test stand itself, and were removed via background subtraction.

# **B.1** Data Preprocessing

For all the received datasets, any corrupt or invalid datasets must be identified and removed. Rejection criteria include anomalous signal amplitude, identification as suspect by the data collector(s), and a lack of positional data. For many of the datasets, the positional data, collected with either surveying instrumentation or GPS, is delivered apart from the actual EMI sensor readings. Particularly for the ERDC test stand data received from Mr. Jonathan Miller, there is not a one-to-one correspondence between the sensor stations and the positional data points supplied. Thus, a simple algorithm was developed, at the suggestion of Mr. Cliff Morgan of ERDC, to assign the proper spatial location to each sensor station. Essentially, the velocity of the sensor is computed using the GPS data, and the zero velocity GPS locations are assigned to the corresponding sensor station. Any skipped stations are accounted for by monitoring the sensor station index number.

Data must also be calibrated to account for instrument drift. Ideally, the response of a fiducial target, such as an aluminum sphere, will be collected before and after data is collected over a UXO target. Any difference in the pre- and post-survey fiducial data may be attributed to instrument drift. We have assumed the drift to be linear, and a correction is applied to each sensor station using the sensor timestamp.

Practically, targets of interest are not located in purely dielectric media. The extraneous signal from the background must be removed in order to ensure the best possible parameter estimation and classification. There are two methods for background data subtraction. Ideally, the background is surveyed in the absence of the target at the same sensor stations. Then, a simple point-by-point background subtraction may be preformed. In practice, however, this operation is not a feasible one: we cannot remove an unknown target in order to survey the remaining signal contribution. We assume that the measured response at the sensor stations most distant from the peak measured response are due entirely to the environment in which the target is situated. The background response is interpolated linearly between these anchor values. A raw EM-63 data set is compared to a pre-processed data set using this method in Figure B.1.

### **B.2** Model Parameter Estimation

Model parameter estimation is accomplished by least squares optimization. For a general physical model the parameters are determined in a best fit manner as the solution to the optimization problem

$$\hat{\theta}(\theta_0) = \arg\min_{\theta} \|r(\theta_0) - s(\theta)\|_2^2, \tag{B.2.1}$$

where  $\theta_0$  holds the true model parameter vector, r is the true data vector, and s is the simulated (modeled) data vector. It should be noted that the parameter vector  $\theta$  includes the so-called



Figure B.1: A comparison of a raw Geonics EM-63 dataset (left) collected over an 81mm mortar shell at the ERDC UXO/Countermine test stand with a preprocessed dataset (right).

nuisance parameters (such as target orientation and location), but only the target specific parameters (such as pole values) are used for classification purposes. The symbol "~" indicates that the quantity is estimated from data.

Minimization of equation B.2.1 is achieved using the Levenberg-Marquardt algorithm, a method designed to find the minimum of a function F that is the sum of squares of nonlinear functions:

$$F(x) = \frac{1}{2} \sum_{i=1}^{m} [f_i(x)]^2$$
(B.2.2)

Unfortunately, the Levenberg-Marquardt algorithm is a local optimization technique, and as such, it searches out the nearest local minimum to the initial estimate. Therefore, care must be taken to ensure that the initial model parameter estimate is somewhat close to the global minimum value. Two initial estimation approaches have been compared for accuracy and efficiency: a brute force parameter space search, and particle swarm optimization (PSO), a global optimization procedure.

#### **B.2.1** Parameter Space Search

As a first step to proper initialization of the optimization algorithm, we tried a simple search of model space. The dipole model requires an optimization over the space of 15 parameters. A rote space search entails the adjustment of each of these parameters and the calculation of the squared error. The parameters responsible for the lowest residual are recorded and the adjustment is repeated. The optimum parameter values are then used to initialize the Levenberg-Marquardt local optimization algorithm.

The procedure may be tuned by adjusting the step size and range of each parameter, and by eliminating potentially redundant model parameters. The step size must be small enough to avoid stepping over minima in the structure of the residual function. Theoretically, derivative information could be used to determine step size, but for our purposes, the step size was adjusted empirically. The range of the model parameters is determined from the physics of the problem. For example, the pole values are restricted to typical ranges for UXO targets. If it is assumed, as is generally the case, that UXO targets are axially symmetric, we may estimate two (rather than three) pole values, assuming that two axes have identical poles.

#### B.2.2 Particle Swarm Optimization (PSO)

A regular search of model parameter space proves to be quite inefficient, even when the search step size is large and the dimension of the space is small. An alternative approach is to use a global optimization technique in order to find the initial estimate. To this end the idea of using particle swarm optimization (PSO) algorithm, a technique, closely related to genetic algorithms, was suggested and implemented by Dr. Jack Stalnaker. PSO has shown great success with the determination of neural network weighting coefficients [55, 56].

The PSO algorithm was initially developed as a bird flocking simulation where, in two dimensions, the search space was envisioned as a cornfield containing a source of food. It may be observed that a great number of birds will quickly locate the source of food within a short period of time. The question becomes, how is the food location communicated to distant parts of a flock? It is theorized that the location is discerned through the flock dynamic as a whole allows members of the flock to capitalize on the knowledge of other flock members.

Mathematically, the birds are represented as agents initialized to random locations on the cornfield. Each point on the cornfield is assigned a value so that at the center of the field, or the location of the best food source, the value is zero. Each agent evaluates the field at its current position, and if the value is the best encountered thus far, the agent stores that value in the variable pbest[], and the best location in the variables pbestx[] and pbesty[]. Each agent also compares the value of the cornfield at its current location to the best value encountered by any agent yet. The identifying index of the global best agent is stored in the variable gbestso that pbest[gbest] is the optimal field value encountered.

The locations of the agents are updated by a stochastic factor related to the distance of the agent from its personal best location and the distance of the agent from the global best location:

$$\delta_x = \delta_x + 2 * \operatorname{rand}() * (pbestx - x) + 2 * \operatorname{rand}() * (pbestx[gbest] - x)$$
  
$$\delta_y = \delta_y + 2 * \operatorname{rand}() * (pbesty - y) + 2 * \operatorname{rand}() * (pbesty[gbest] - y)$$
(B.2.3)

where the factor of 2 ensures that the random multiplier has a mean of 1, ensuring that the agent will overshoot the target about half the time. The position adjustment factors may also be constrained to prevent the momentum of the agent from growing out of control.

In our cornfield example, then, the birds, or agents fly around the field, directed both by a "nostalgic" acceleration towards local optima and an acceleration towards the global optimum encountered by the entire flock. The concept is easily generalized as a global optimization method, where the field, or hyperspace in the case of multidimensional problems, represents the cost function to be optimized. The dimensions of the cost function are the problem parameters. The agents, or particles, represent potential solutions to the optimization problem flying through hyperspace toward better solutions. The concept is related to genetic algorithms in that the flight path of each particle is directed by a combination of two adjustments, local and global, in essence, breeding the best overall solution with the local best solutions. The success of the particle swarm method appears to be attributable to the tendency of the particles to overshoot the target optima, forcing the method to explore regions of parameter space between the hitherto encountered optima, while the increasing momentum applied to each step causes the particles to explore unknown regions of parameters space.

There are two user controlled variables in the PSO algorithm: the number of iterations (i.e.

Algorithm	No. of Parameters	Algorithm Details	Execution
Space Search	4	10 linearly spaced pa-	11 minutes per data set
		rameter values	
Particle Swarm	4	20 particles, maximum	1.5 minutes per data set
Optimization		100 iterations	

Table B.1: A demonstration of the efficiency of the particle swarm optimization algorithm as compared to a simple parameter space search

the number of flight segments through hyperspace) and the number of particles. Clearly, a larger number of particles will potentially explore a greater portion of parameter space, while an increased number of iterations will cause the particles to explore the best fit regions of parameter space more thoroughly. However, at each iteration, a calculation of the residual (and thus a model simulation) must be performed for each particle. We can show convergence for the dipole model using 20 particles and roughly 100 iterations. The PSO approach to initial parameter estimation is far more efficient than the parameter space search, as detailed in Table B.1. In both cases, it was assumed that the location and orientation of the target could be estimated from outside sources, and it was assumed that the target was axially symmetric, reducing the required number of poles and expansion coefficients to two. In addition to the computational time improvement detailed in Table B.1, the PSO approach appears to yield more optimal initial parameters than the space search, generating a more rapid convergence of the Levenberg-Marquardt algorithm. This is to be expected as the PSO estimated parameters are not constrained to a grid.

#### B.2.3 Model Parameter Estimation from Field Data

The EM63 sensor was positioned over a grid spaced variably between 10 and 25 centimeters. Data were collected over four mortar shells, identified by diameter: 105 mm, 81 mm, 60 mm, and 40 mm. The response was measured for three orientations of each shell: vertical nose-up, vertical nose-down, and horizontal nose north. Similar pole estimates were attained for each orientation, as required by the classification algorithm. The results of this analysis are presented in Figures B.2–B.5, and summarized in Table B.2.

Analysis of the EM63 time domain data obtained from Mr. Jay Bennett and Mr Cliff



Figure B.2: Time domain data collected over a 40 mm mortar shell using the Geonics EM63 (dashed line) is compared to data simulated from estimated poles (solid line). The curves are plotted for a profile from y = 0m to y = 4m, where x = 0m and z = -0.58m. Two target orientations are represented: (a) horizontal nose north, and (b) vertical nose down.



Figure B.3: Time domain data collected over a 60 mm mortar shell using the Geonics EM63 (dashed line) is compared to data simulated from estimated poles (solid line). The curves are plotted for a profile from y = 0m to y = 4m, where x = 0m and z = -0.58m. Two target orientations are represented: (a) horizontal nose north, and (b) vertical nose up.



Figure B.4: Time domain data collected over a 81 mm mortar shell using the Geonics EM63 (dashed line) is compared to data simulated from estimated poles (solid line). The curves are plotted for a profile from y = 0m to y = 4m, where x = 0m and z = -0.58m. Two target orientations are represented: (a) horizontal nose north, and (b) vertical nose up.



Figure B.5: Time domain data collected over a 105 mm mortar shell using the Geonics EM63 (dashed line) is compared to data simulated from estimated (solid line) poles. The curves are plotted for a profile from y = 0m to y = 4m, where x = 0m and z = -0.58m. Two target orientations are represented: (a) horizontal nose north, and (b) vertical nose down.

Target	Orientation	Vertical Distance	Pole 1 (rad/s)	Pole 2 $(rad/s)$	Pole 3 (rad/s)
		from Sensor (cm)			
105 mm	34	VNU	1.1600e+04	1.1635e + 04	1.2221e+03
	37	VND	7.9908e + 03	7.2919e + 03	990.0333
	14	HZN	3.9160e + 03	3.7413e + 03	902.2177
	15	VNU	1.3969e + 04	1.4018e+04	2.6223e+03
81 mm	14	VND	1.3308e + 04	1.2239e + 04	2.6397e + 03
	0	HZN	1.4112e + 04	1.3932e + 04	2.6407e + 03
60 mm	4	VNU	5.5040e + 03	5.5021e + 03	1.7798e + 03
	-3	HZN	5.5963e + 03	4.2664e + 03	1.7606e + 03
40 mm	-11	VND	5.3845e + 03	5.5845e + 03	1.6498e + 03
	-15	HZN	5.5726e + 03	4.4078e + 03	1.3147e + 03
VNU = Vertical Nose Up, VND = Vertical Nose Down, HZN = Horizontal Nose North					

Table B.2: Estimated poles for 105 mm, 81 mm, 60 mm, and 40 mm UXO items at various orientations. The poles are estimated from EM63 data collected on a grid centered on the target.

Target	Pole 1	Pole 2	Pole 3
M42	8942	5626	4416
81mm	4602	4127	2988
$90 \mathrm{mm}$	3885	3729	1811
BLU26	4381	4693	2650
$76 \mathrm{mm}$	4815	4596	2210
60mm	4821	4348	2168
40mm	806	720	1678
20mm	4089	4373	3430
2.75in	6254	7562	2427

Table B.3: Mean pole values estimated from data collected at the ERDC test stand.

Morgan was completed for a suite of UXO targets. The results of the pole estimation analysis are summarized in Table B.3 and visualized in Figure B.6. To generate four pole model from these one pole averages in the Table B.3 we generated four equally spaced poles within the range of .5 and 1.5 times of the corresponding estimated poles for each axis.



Figure B.6: Poles estimated from data collected at the ERDC test stand. Each symbol represents a different target orientation, target instance, or target burial depth.

# Appendix C Hidden Markov Model

In this appendix we provide a detailed explanation of the problem of inference and training a trellis with application to our Markov chain/random walk perturbation model developed in chapter 7. First the problem is explained and then the corresponding algorithms will be discussed.

### C.1 Hidden Markov Model Structure

A hidden Markov model (HMM), M, is described by a set of states,  $\Delta = \{s_1, ..., s_Q\}$  with cardinality Q and its realization  $\delta = \{\delta_1, ..., \delta_N | \delta_i \in \Delta\}$  over time (for our case over space), the probability transition matrix,  $P_{Q \times Q}$ , describing the probability of state transition from time i to time j,  $P(\delta_{i-1} \to \delta_i)$ . Depending on the state  $\delta_i$  HMM will generate a random variable  $y_i$  with probability distribution  $P_{y_i}$  which we will observe. The first problem with HMM is to estimate the states  $\delta$  from observed data  $y_i$  which is known as forward problem. The second problem which is to estimate model parameter  $P_{Q \times Q}$  when we have a set of labelled training samples for  $\delta$ . The probability of generating a specific sequence of  $\mathbf{y} = \{y_0, ..., y_N\}$ can be written as:

$$P(\mathbf{y}|M) = \sum_{\delta_1,\dots,\delta_N \in \Delta^N} \prod_{k=1}^N P(\delta_{k-1} \to \delta_k) P(y_{k-1}|\delta_{k-1})$$
(C.1.1)

Figure C.1 shows the trellis structure of the HMM.



Figure C.1: Hidden Markov model

# C.2 Viterbi Algorithm

As explained in previous section the first problem is the estimation of states,  $\delta$ , when we have a set of observed data  $\mathbf{y} = \{y_0, ..., y_N\}$ . We denote this states as  $\hat{\delta}$  and using equation C.1.1 we can estimate it using maximum likelihood estimation:

$$\hat{\boldsymbol{\delta}} = \operatorname*{arg\,max}_{\delta_1,\dots,\delta_N \in \Delta^N} \prod_{k=1}^N P(\delta_{k-1} \to \delta_k) P(y_k | \delta_k) \tag{C.2.1}$$

Now assume that  $P(y_k|\delta_k)$  has a Gaussian distribution  $N(m(\theta, \delta_k), \sigma^2)$ . Applying the logarithm function to both sides and using the Gaussian distribution we will have:

$$\hat{\boldsymbol{\delta}} = \underset{\delta_1,\dots,\delta_N \in \Delta^N}{\operatorname{arg\,min}} \sum_{k=1}^N \left\{ \|y_k - m(\theta, \delta_k)\|_2^2 - \sigma^2 \log(P(\delta_{k-1} \to \delta_k)) \right\}$$
(C.2.2)

which corresponds to finding the shortest path on the trellis, using an exhaustive search the complexity of the search would be  $O(Q^N)$ . Viterbi algorithm (VA) [49] uses the properties of Markov chain to reduce the complexity of search to  $O(NQ^2)$ . Using equation C.2.2 it can be seen that the total cost at step (j + 1),  $T_{j+1}$  is a recursive function of partial costs  $C_{j+1}$  and can be written as:

$$T_{j+1} = \sum_{k=1}^{j+1} \left\{ \|y_k - m(\theta, \delta_k)\|_2^2 - \sigma^2 \log(P(\delta_{k-1} \to \delta_k)) \right\} =$$
(C.2.3)  
$$\sum_{k=1}^{j} \left\{ \|y_k - m(\theta, \delta_k)\|_2^2 - \sigma^2 \log(P(\delta_{k-1} \to \delta_k)) \right\} + \left\{ \|y_{j+1} - m(\theta, \delta_{j+1})\|_2^2 - \sigma^2 \log(P(\delta_j \to \delta_{j+1})) \right\} = T_j + C_{j+1}.$$



Figure C.2: Viterbi algorithm

In VA, at time step j + 1, Q total cost functions,  $T_{(j+1),q}$ , q = 1...Q for each state in  $\delta_{j+1}$ (corresponding to each path entering to that state in trellis) is calculated using recursive formula C.2.3 and only the path with lowest metric will be stored which is known as the survivor path. So at each time step only  $Q^2$  computation is required. If we have a trellis of length N then we will need  $NQ^2$  computations. After reaching the end of trellis we will follow back the shortest paths and that will be the the shortest path on the trellis. Figure C.2 demonstrates VA indicating the survivor path and shortest path on the trellis.

# C.3 Training the Trellis

Now assume that realization  $\delta$  is not hidden to us and we want to infer about the transition probability  $P(\delta \rightarrow \delta')$  or  $P_{Q \times Q}$ . Here we can have a maximum likelihood estimation of transition probabilities as:

$$\hat{P}(\delta \to \delta') = \frac{c(\delta \to \delta')}{\sum_{s \in \Delta} c(\delta \to s)}$$
(C.3.1)

where  $c(\delta \to \delta')$  is the is the number of transitions we had in the trellis from  $\delta$  to  $\delta'$  and  $\sum_{s \in \Delta} c(\delta \to s)$  indicates the total number of transition from state  $\delta$  to any state in the trellis.

# C.4 MATLAB code

The following VA algorithm code was written in MATLAB and was used in implementation of Min-Max algorithm in chapter 7.

```
function y=find_shortest(x,data,loc,N_pert,length_pert)
% y=find_shortest(x,data,loc,N_pert,length_pert) finds the sensor
%path that maximizes the cost function, x containes the model parameters,
%data is the perturbed signal, N_pert is the number of perturbations
% length_pert is the maximum value for perturbations in each direction.
Length_data=length(data);
% Separate mixed data into time and frequency components
D=reshape(data,Length_data/N_freq,N_freq)';
M=length_pert; % The length of perturbation
Q=N_pert; % The number of samples for perturbation
T=transition(Q); % Build the transition matrix using
%length and number of perturbations
X=linspace(-M,M,N_pert); Y=X; [A B]=meshgrid(X);
Pert=[A(:) B(:)]; % Build the perturbation space
SP=zeros(Q<sup>2</sup>,length(loc)); % Build the Trellis
SPM=zeros(Q<sup>2</sup>,length(loc)); % Build the cost function
%for the Trellis and initialize it
for j=1:Q^2
    SPM(j,1)=norm(D(:,1)-model(x,loc(1,:)+Pert(j,:)))^2;
    % Initialize the first stage state cost functions for the trellis
end
% Find the survivor path using the trellis
for i=2:length(loc)
    for j=1:(Q^2)
        temp1=norm(D(:,i)-model(x,loc(i,:)+Pert(j,:)))^2;
        nz=find(T(:,j)~=0);
        temp=[ ];
        for k=1:length(nz)
            temp(k)=SPM(nz(k),i-1)+temp1;
        end
        [M_val M_idx]=max(temp);
        SP(j,i)=nz(M_idx);
        SPM(j,i)=M_val;
    end
```

 $\operatorname{end}$ 

```
% Track back and find the shortest path
[m_v m_i]=(max(SPM(:,end)));
T2=[]; T2=[T2 m_i]; for i=0:(length(loc)-2)
    T2=[T2 SP(m_i,end-i)];
    m_i=SP(m_i,end-i);
end
T2=T2(end:-1:1);
for i=1:length(T2);
    est_pert(i,:)=Pert(T2(i),:);
end
y=est_pert+loc;
```

# Bibliography

- S. J. Norton and I. J. Won, "Identification of buired unexploded ordnance from broadband electromagnetic induction data," *IEEE trans on geoscience and remote sensing*, vol. 39, pp. 89–129, 2001.
- [2] Y. Zhang, L. Collins, and e. al., "Sensing of unexploded ordnance with magnetometer and induction data: Theory and signal processing," *IEEE Transactions on Geoscience* and Remote Sensing, vol. 47, pp. 1005–1015, 2003.
- [3] H. Huang and I. Won, "Characterization of uxo-like targets using broadband electromagnetic induction sensors," *IEEE Transaction on Geoscience and Remote Sensing*, vol. 41, pp. 652–663, 2003.
- [4] L. Collins, P. Gao, and L. Carin, "An improved bayesian decision theoretic approach for land mine detection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 37, 1999.
- [5] P. Gao and L. M. Collins, "A theoritical performance analysis and simulation of timedomain emi sensor data for land mine detection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 38, 2000.
- [6] L. M. Collins, P. Gao, and D. Schofield, "A statistical approach to landmine detection using broadband electromagnetic induction data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 40, pp. 89–129, 2002.
- [7] S. L. Tantum and L. M. Collins, "A comparison of algorithms for subsurface target detection and identification using time-domain electromagnetic induction data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 39, 2001.

- [8] N. Geng, C. Baum, and L. Carin, "On the low-frequency natural response of conducting and permeable targets," *IEEE Trans. Geoscience and Remote Sensing*, vol. 37, pp. 347– 359, 1999.
- [9] L. S. Riggs and J. E. Mooney, "Identification of metallic mine-like objects using low frequency magnetic fields," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 39, 2001.
- [10] A. B. Tarokh, E. L. Miller, I. J. Won, and H. Huang, "Statistical classification of buried objects from time and frequency domain emi data," *Radio Science*, vol. 39, 2004.
- [11] E. L. Miller, "On some options for statististical classification of buried objects from spatially sampled time or frequency domain emi data," SPIE Conference on Detect. of Mines and Minelike Targets, vol. 4394, 2001.
- [12] A. Aliamiri, J. Stalnaker, and E. Miller, "Statistical classification of buried uexploded ordnance using non-parametric prior models," *submitted to IEEE Transactions on Geoscience and Remote Sensing.*
- [13] L. Carin, H. Yu, and C. E. Baum, "On the wideband electromagnetic inductance signature of conducting and permeable targets," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 39, pp. 1206–1213, 2001.
- [14] L. Pasion and D. Oldenburg, "A discrimination algorithm for uxo using time domain electromagnetics," Jour. Eng. & Envir. Geophys., vol. 47, pp. 91–102, 2001.
- [15] S. M. Kay, Fundamentals of Statistical Signal Processing. Estimation Theory. Prentice Hall, 1993.
- [16] S. M. Kay, Fundamentals of Statistical Signal Processing. Detection Theory. Prentice Hall, 1993.
- [17] V. Trees, Detection, Estimation, and Modulation Theory. Part I. Wiley, 1968.
- [18] A. Aliamiri, J. Stalnaker, and E. Miller, "A bayesian approach for classification of buried objects using non-parametric prior model," in *IEEE Conference on Geoscience and Remote Sensing*, 2006.

- [19] V. B. Melas, Functional Approach to Optimal Experimental Design. Springer, 2006.
- [20] M. L. Burrows, A Theory of Eddy-Current Flaw Detection. PhD thesis, Department of Electrical Engineering, University of Michigan, Ann Arbor, Michigan, 1964.
- [21] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. John Wiley & Sons, 2001.
- [22] B. W. Silverman, Density Estimation for Statistics and Data Analysis. Chapman and Hall, London, 1986.
- [23] J. Hwang, S. Lay, and A. Lippman, "A nonparametric multivariate density estimation: A comparative study," *IEEE Transactions Signal Processing*, vol. 42, pp. 2795–2810, 1994.
- [24] D. Scott, Multivariate Density Estimation: Theory, Practice, and Visualization. John Wiley, New York, 1992.
- [25] K. Fukunaga, Introduction to Statistical Pattern Recognition. Academic Press, 2nd edition, 1990.
- [26] J. Nocedal and S. Wright, Numerical Optimization. Springer Verlag, 2000.
- [27] L. Scharf, Statistical Signal Processing: Detection, Estimation and Time Series Analysis.
   Addison-Wesley, 1991.
- [28] M. Tanner, Tools for Statistical Inference: Methods for Exploration of Posterior Distributions and Likelihood Functions. Springer, New York, 1993.
- [29] M. H. DeGroot and M. J. Schervish, Probability and Statistics, 3rd Edition. Addison Wesley, 2002.
- [30] N. Vapnik, Statistical Learning Theory. Wiley, 1998.
- [31] C. J. Burges, "A tutorial on support vector machines for pattern recognition," Data Mining and Knowledge Discovery, vol. 2, no. 2, pp. 955–974, 1998.
- [32] J. Weston and C. Watkins, "Support vector machines for multiclass pattern recognition," in Proceedings of the Seventh European Symposium on Artificial Neural Networks, 1999.

- [33] J. Platt, A. Cristianini, and J. Shawe-Taylor, "Large marging dags for multiclass classification," in Advances in Neural Information Processing Systems 12, pp. 547–553, 2000.
- [34] I. Won, D. Keiswetter, D. Hanson, E. Novikova, and T. Hall, "Gem-3: A monostatic broadband electromagnetic induction sensor,," *Journal of Environmental and Engineering Geophysics*, vol. 47, pp. 53–64, 1997.
- [35] C. W. Therrien, Discrete Random Signals and Statistical Signal Processing. Prentice Hall, 1992.
- [36] S. L. and L. McWhorter, "Geometry of the cramer-rao bound," Signal Processing, vol. 31, pp. 1–11, 1993.
- [37] C. D. Meyer, Matrix Analysis and Applied Linear Algebra. SIAM, Philadelphia, 2000.
- [38] S. Kullback, Information Theory and Statistics. 1968.
- [39] B. D. Herbert, Tables of Integrals and Other Mathematical Data. Prentice Hall, 1961.
- [40] V. Fedorov, Theory of Optimal Experiments. Academic Press, 1972.
- [41] L. Xuejun and L. Carin, "Application of the theory of optimal experiments to adaptive electromagnetic-induction sensing of buried targets," *IEEE Transactions Geoscience and Remote Sensing*, vol. 26, no. 8, pp. 961–972, 2004.
- [42] G. H. Golub and C. F. V. Loan, Matrix Computations, third ed. 1996.
- [43] Z. Bai and G. H. Golub, "Bounds for the trace of the inverse and the determinant of symmetric positive definite matrices," SIAM AG on Orthogonal Polynomials and Special Functions.
- [44] M. Powell, "A fast algorithm for nonlineary constrained optimization calculations," Numerical Analysis, ed. G.A. Watson, Lecture Notes in Mathematics, Springer Verlag.
- [45] S. Han, "A globally convergent method for nonlinear programming," Journal of Optimization Theory and Applications.
- [46] A. B. Tarokh and E. L. Miller, "Subsurface sensing under sensor positional uncertainty," submitted to IEEE Transactions on Geoscience and Remote Sensing.

- [47] S. L. Tantum, Y. Yu, Q. Zhu, Y. Q. Wang, and L. M. Collins, "Mitigating measurement position uncertainty to improve uxo detection," UXO Forum.
- [48] J. Proakis, *Digital Communications*. McGraw-Hill, 1995.
- [49] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Trans. Info. Theory.*
- [50] Y. Das, J. McFee, J. Toews, and G. C. Stuart, "Analysis of an electromagnetic induction detector for real-time localization of buried objects," *IEEE Trans. Geoscience and Remote Sensing*, vol. 28, pp. 278–287, May 1990.
- [51] P. Weichman, "Universal early-time response in high-contrast electromagnetic scattering," *Physical Review Letters*, vol. 91, pp. 143908(1)–143908(4), October 2003.
- [52] P. Weichman, "Surface modes and multipower-law structure in the early-time electromagnetic response of magnetic targets," *Physical Review Letters*, vol. 93, pp. 023902(1)– 023902(4), July 2004.
- [53] G. Sower, J. Endsley, and E. Christy, "Discrimination of metal land mines from metal clutter: Results of field tests," in SPIE Conference on Detection of Mines and Minelike Targets: IV, pp. 78–88, April 1999.
- [54] L. R. Pasion, "Inversion of magnetic and electromagnetic data for the discrimination of UXO," in *Partners in Environmental Technology Technical Symposium*, (Washington, D.C.), p. 54, SERDP and ESTCP, November 27-29 2001.
- [55] R. C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in Proceedings of the Sixth International Symposium on Micro Machine and Human Science, (Nagoya, Japan), pp. 39–43, 1995.
- [56] J. Kennedy and R. C. Eberhart, New Ideas in Optimization, ch. The particle swarm: social adaptation in information processing systems. McGraw-Hill, 1999.