PERFORMANCE EVALUATION OF THE ADAPTIVE COSINE ESTIMATOR DETECTOR FOR HYPERSPECTRAL IMAGING APPLICATIONS

A Thesis Presented

by

Eric Truslow

to

The Department of Electrical and Computer Engineering

in partial fulfillment of the requirements for the degree of

Master of Science

in

Electrical Engineering

Northeastern University
Boston, Massachusetts

August 2012
© Copyright 2012 by Eric Truslow
All Rights Reserved
Performance Evaluation of the Adaptive Cosine Estimator Detector for Hyperspectral Imaging Applications

Eric Truslow

August 24, 2012
Abstract

Detecting subpixel objects is a difficult but core problem in hyperspectral imagery (HSI). This is accomplished by using a detection algorithm with the object’s signature on each pixel in a scene. However, the effectiveness of a detector for a subpixel object depends on the background and the fraction of the pixel inhabited by the object (fill fraction). In this work, two spectral detectors are considered: the matched filter, and t-detector (cosine). A technique, relying on characteristic function inversion, is developed for computing the probability of detection and probability of false alarm for each detector under a multivariate normal (MVN) model and a multivariate t (MVT) model. Two novelties of this technique are: (1) it does not put constraints on the covariances or means of the inputs (eg. zero mean, identity covariance, etc.), (2) it uses numerical integration instead of Monte Carlo techniques. Using this technique, the performance of each detector for specific fill fractions and backgrounds is assessed using a multivariate t model for the background and a multivariate normal model for the object. The model and algorithm are evaluated using real HSI data with ground truth. Finally, the relationship between the predicted output at various fill fractions and false alarm mitigation techniques is investigated.
Acknowledgements

This thesis is the culmination of a year of work that was done at the direction of Dr. Dimitris G. Manolakis at MIT Lincoln Laboratory. I am sincerely grateful for his guidance and patience in finishing this work. The contents of this thesis rely heavily on previous research at MIT Lincoln Laboratory by Michael Rossacci, who developed the prediction model for the matched filter; Robert DiPietro, who developed the matched filter with false alarm mitigation; and Michael Pieper whose work on ACE necessitated the development of this thesis. I would also like to thank Professor Jennifer Dy for her wonderful class in pattern recognition and for reading this thesis, and Professor Vinay K. Ingle for giving me the opportunity to work with him and Dr. Manolakis over the past two years.
## Contents

Abstract 2
Acknowledgements 3
List of Abbreviations 12
List of Symbols 13

1 Introduction 14
   1.1 Introduction to Hyperspectral Imagery (HSI) 14
   1.2 Background 17
      1.2.1 The Linear Mixing Model and Subpixel Targets 17
      1.2.2 Preliminary Signal Model 19
      1.2.3 Hypothesis Testing 22
   1.3 Introduction to Signal Detection Algorithms 27
      1.3.1 A Derivation of the Matched Filter (MF) 27
      1.3.2 Statistics of the Matched Filter 29
      1.3.3 The Adaptive Cosine Estimator (ACE) 31
      1.3.4 The Statistics of ACE 34
   1.4 Thesis Organization 35
2 Performance of the MF and ACE in Unimodal Clutter

2.1 Unimodal Test Cases .................................................. 37

2.2 Test Case 1: Effect of Dimension .................................. 39

2.3 Test Cases 2: Effect of Covariance Scaling ..................... 40

2.4 Test Case 3: Effect of Linear Mixing ............................. 43

2.5 Test Case 4: Effect of Tail Heaviness ............................ 45

2.6 Test Case 5: Effect of Covariance Mismatch ..................... 46

2.7 Summary ................................................................. 48

3 The Geometry of Detectors and False Alarm Mitigation Techniques

3.1 The Signal and the Orthogonal Subspaces ....................... 49

3.1.1 Signed Detectors .................................................... 51

3.1.2 Subspace Detectors ............................................... 52

3.1.3 Detector Summary .................................................. 54

3.2 The Geometry of False Alarm Mitigation Techniques .......... 57

3.3 Relation of Fill Fraction to ACE ................................. 58

3.4 Summary ................................................................. 60

4 The Distribution of ACE with Unconstrained Covariance Mismatch

4.1 The CDF of ACE and the $\beta$-detector with Covariance Mismatch .... 62

4.2 Computing the CDF of ACE ......................................... 63

4.2.1 The CDF of the t-detector ....................................... 64

4.3 Computing the CDF of the $\beta$-detector ........................ 66

4.3.1 The CDF of the F-detector ....................................... 66

4.4 Handling MVT Distributed Input ................................... 67

4.5 Algorithm Verification ............................................... 68

4.6 Summary ................................................................. 70
C.2.1 Obtaining the PDF from the Characteristic Function       105
C.2.2 Obtaining the CDF from the Characteristic Function       106

Bibliography                   107
List of Figures

1.1 Illustration of hyperspectral system. Adapted from [17].......................... 16
1.2 Illustration of hyperspectral cube. Adapted from [17]........................... 17
1.3 (a) Illustration of unresolved target and (b) resulting pixel fill fractions.... 19
1.4 Illustration of target, background and mixed pixel distributions using the LMM. 22
1.5 Illustration of detector densities under $H_0$ and $H_1$. Setting a threshold (black)
determines the probability of false alarm $P_{FA}$ and probability of detection $P_D$. 26
1.6 Illustration of a ROC curve (black) using the densities in Figure 1.5. Selecting
a probability of false alarm of $10^{-3}$ results in a probability of detection of 0.46
(red). ................................................................. 26
1.7 Illustration of de-meaning and whitening in 2-D with MVN distributed back-
ground (green) and target (red). Distribution of data (a) at the input, (b)
after de-meaning, and (c) after de-meaning and whitening. ......................... 29
1.8 Illustration of the (a) MF and (b) ACE in two dimensions with an uncorrelated
MVN distributed background (green) and target (red). For demonstration,
the input is already white, $\mathbf{x} \sim \mathcal{N}(\mathbf{s}_t, \mathbf{I})$, making whitening unnecessary. 32
2.1 ROC curves for the matched filter and cosine detector for various dimensions
$p$, and a VSNR of 5. ....................................................... 40
2.2 Example distributions where \( (\Sigma_t = \sigma^2 \Sigma_b) \): (a) Target distribution is scaled version of background; (b) target and background share the same scale but mixed pixels do not; (c) full pixels and mixed pixels are scaled down.

2.3 Test Case 2, ROC curves when \( \Sigma_t \) is \( \sigma^2 \Sigma_b \); ie. the target covariance matrix is a scaled version of the background covariance matrix.

2.4 Test Case 3 for ACE (solid) and MF (dashed). \( P_D \) vs. fill fraction for fixed \( P_{FA} \).

2.5 MF (solid) and ACE (blue dashed) detectors where the background is MVT with tail parameter \( \nu \).

2.6 Probability of detection of ACE (colored) and MF (black) for various fill fractions when there is covariance mismatch. Only the MF curves for \( P_{FA} = 10^{-3} \) are included.

3.1 Illustration of projection of whitened input \( \tilde{x} \) (blue) on the signal subspace (green) and orthogonal subspace (pink), when \( p = 2 \).

3.2 Matched filter and orthogonal projection distance for some synthetic data. Decision boundaries at various thresholds for the MF, MF-FAM, and cosine detectors are overlaid.

3.3 Illustration of how the CDF of ACE changes with fill fraction. The pink lines are the 10, 50 and 90 percent CDF points for each fill fraction.

4.1 Illustrations and definitions of detection geometry when (a) there is single target vector and (b) there are several target vectors. \( \theta \) is the angle between the vector \( \tilde{x} \) and the subspace formed by \( \tilde{s} \) or \( \tilde{S} \).

4.2 Univariate t-distribution CDF using our technique (blue), Monte Carlo simulation (red) and MATLAB’s built-in function (green) with various non-centrality parameters and dimension \( p = 20 \).
6.8 (a) Orthogonal projection distance vs. MF with MF-FAM thresholds (dashed); (b) ACE vs. FFE and same MF-FAM thresholds. The true target pixels (red) in (a) have small orthogonal projections and from (b) we see that the MF-FAM would be able to remove a large number of confuser material pixels (purple, green). In contrast, an ASB threshold, (using both the MF and ACE) may still have many false alarms.

6.9 ACE vs. FFE for Material B. (a) MF-FAM threshold and predicted constant CDF curves using the background covariance matrix as the target covariance matrix ($\Sigma_t = \Sigma_b$). (b) True targets (red) and confusers (green, purple). True targets should lie between the blue and green curves.

6.10 ACE vs. FFE for Material B. (a) MF-FAM threshold and predicted constant CDF curves using the estimated target covariance ($\Sigma_t = \hat{\Sigma}_t$). (b) True targets (red) and confusers (green, purple). True targets should lie between the blue and green curves.
## List of Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMF</td>
<td>Adaptive matched filter</td>
<td>57</td>
</tr>
<tr>
<td>ASB</td>
<td>Adaptive sidelobe blanker</td>
<td>57</td>
</tr>
<tr>
<td>CDF</td>
<td>Cumulative density function</td>
<td>35</td>
</tr>
<tr>
<td>CFAR</td>
<td>Constant false alarm rate</td>
<td>45</td>
</tr>
<tr>
<td>FAM</td>
<td>False alarm mitigation</td>
<td>57</td>
</tr>
<tr>
<td>FAR</td>
<td>False alarm rate</td>
<td>24</td>
</tr>
<tr>
<td>FFE</td>
<td>Fill fraction estimator</td>
<td>30</td>
</tr>
<tr>
<td>GLRT</td>
<td>Generalized likelihood ratio test</td>
<td>23</td>
</tr>
<tr>
<td>HSI</td>
<td>Hyperspectral image(ry)</td>
<td>14</td>
</tr>
<tr>
<td>LMM</td>
<td>Linear mixing model</td>
<td>18</td>
</tr>
<tr>
<td>LRT</td>
<td>Likelihood ratio test</td>
<td>23</td>
</tr>
<tr>
<td>MLE</td>
<td>Maximum likelihood estimate</td>
<td>32</td>
</tr>
<tr>
<td>MVN</td>
<td>Multivariate normal</td>
<td>20</td>
</tr>
<tr>
<td>MVT</td>
<td>Multivariate t</td>
<td>20</td>
</tr>
<tr>
<td>PDF</td>
<td>Probability density function</td>
<td>23</td>
</tr>
<tr>
<td>ROC</td>
<td>Receiver operating characteristic</td>
<td>24</td>
</tr>
<tr>
<td>SNR</td>
<td>Signal to noise ratio</td>
<td>30</td>
</tr>
<tr>
<td>VSNR</td>
<td>Voltage signal to noise ratio</td>
<td>30</td>
</tr>
</tbody>
</table>
List of Symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathbf{x}$</td>
<td>test vector</td>
<td>17</td>
</tr>
<tr>
<td>$\mathbf{x}_b$</td>
<td>background pixel</td>
<td>20</td>
</tr>
<tr>
<td>$\mathbf{s}_t$</td>
<td>target vector</td>
<td>19</td>
</tr>
<tr>
<td>$\tilde{\mathbf{x}}$</td>
<td>whitened and de-meaned test vector</td>
<td>28</td>
</tr>
<tr>
<td>$\tilde{s}$</td>
<td>whitened and de-meaned target vector</td>
<td>28</td>
</tr>
<tr>
<td>$\mathbb{E} [\mathbf{x}]$</td>
<td>expected value of the vector $\mathbf{x}$</td>
<td>20</td>
</tr>
<tr>
<td>$\text{Cov} [\mathbf{x}]$</td>
<td>covariance of the vector $\mathbf{x}$</td>
<td>20</td>
</tr>
<tr>
<td>$\mu_b$</td>
<td>background mean</td>
<td>19</td>
</tr>
<tr>
<td>$\Sigma_b$</td>
<td>background covariance matrix</td>
<td>20</td>
</tr>
<tr>
<td>$\Sigma_b^{-1/2}$</td>
<td>square root matrix of $\Sigma_b^{-1}$</td>
<td>28</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>target fill fraction</td>
<td>19</td>
</tr>
<tr>
<td>$\langle \tilde{s} \rangle$</td>
<td>subspace containing $\tilde{s}$</td>
<td>50</td>
</tr>
<tr>
<td>$\mathbf{P}_{\tilde{s}}$</td>
<td>projection matrix on the subspace $\langle \tilde{s} \rangle$</td>
<td>50</td>
</tr>
<tr>
<td>$\mathcal{N}(\cdot)$</td>
<td>multivariate normal distribution</td>
<td>20</td>
</tr>
<tr>
<td>$t_\nu(\cdot)$</td>
<td>multivariate t-distribution with $\nu$ degrees of freedom</td>
<td>20</td>
</tr>
<tr>
<td>$</td>
<td></td>
<td>\mathbf{x}</td>
</tr>
<tr>
<td>$P_D$</td>
<td>probability of detection</td>
<td>24</td>
</tr>
<tr>
<td>$P_{FA}$</td>
<td>probability of false alarm</td>
<td>24</td>
</tr>
<tr>
<td>$\sim$</td>
<td>“is distributed as”</td>
<td>20</td>
</tr>
<tr>
<td>$\doteq$</td>
<td>“equal in distribution to”</td>
<td>67</td>
</tr>
<tr>
<td>$\Pr { y \leq b }$</td>
<td>probability that the random variable $y$ is less than or equal to $b$</td>
<td>64</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

Identifying materials by radiometric measurements usually requires taking a sample of the material to a laboratory and making measurements in a controlled environment. This is impractical when the goal is to identify materials or map materials over large areas of the earth. While, airborne and spaceborne sensors have made it possible to rapidly image large swaths of the earth, the large amount of data they produce makes it necessary to have algorithms that can automatically process this data. To detect materials in these images, so-called detection algorithms are usually employed [17]. However, the success of these algorithms depends entirely on the type of material and the type of scene. In this thesis we develop models and algorithms for predicting how well these detection algorithms will perform for a given material and scene.

1.1 Introduction to Hyperspectral Imagery (HSI)

A hyperspectral image (HSI) is much like an image taken with a regular camera. In a regular camera there are usually three spectral bands: red green and blue; these bands cover only the visible range which is a narrow slice of the electromagnetic spectrum. The range of the electromagnetic spectrum that hyperspectral sensors are sensitive in is typically far greater
than a normal camera, and the number of spectral channels or bands that the sensor is sensitive in can number in the hundreds.

In this work we are primarily concerned with sensors that operate in the reflective range of the spectrum; this ranges from 0.4 to 3 \( \mu m \) and includes the visible and near infrared regions [27]. Radiometric sensors measure the amount of energy that reaches the sensor from the ground and other sources. These measurements are said to be in the radiance domain; in the reflective region this energy is dependent on the source of radiation, the atmosphere and the reflectance of materials on the ground (see Figure 1.1). For passive sensors like hyperspectral sensors, the sun is the primary source of radiation; in our sensor system, the sun emits energy as a function of wavelength, which is then absorbed by the atmosphere both before and after it is reflected by materials on the earth’s surface; the energy that reaches these materials is reflected according to each material’s reflectance signature and spatial geometry. Taking out the solar and atmospheric contributions to the sensor measurements and ignoring material geometry, leaves only the ground reflectance; in this thesis all quantities are assumed to be in terms of reflectance or are said to be in the reflectance domain. This simplified view is all that is necessary for our purposes, for more thorough discussions of remote sensing systems see [27] and [28].

The reflectance at any point in a scene is determined by the material at that point, but this continuous measurement is effectively binned by the sensor into a grid of pixels. Each pixel has an aggregate reflectance spectrum, which is binned into spectral channels. This grid of pixels form the first two dimensions of a hyperspectral image, while the reflectance of each pixel becomes the third dimension as illustrated in Figure 1.2. Although, HSI data is typically stored as a “cube”, the algorithms we use do not need to retain the spatial components of the image; instead the entire image can be rearranged into a matrix of pixels.

To actually detect a material, a detection algorithm is designed using the reflectance signature of the target material. This signature is usually obtained in a laboratory at very high spectral resolution as with the National Aeronautics and Space Administration’s ASTER
spectral library [2]. This signature is then downsampled to match the spectral channels of the sensor. This spectral signature and a pixel in the image are the inputs to our detection algorithm. We run the detection algorithm on every pixel in the cube and get a “score” for each pixel which indicates how similar the test pixel is to the signature. Based on this score, we want to identify pixels that contain the target material and pixels that do not contain the target material. To understand and predict how well we can identify these pixels, we need a statistical model for pixels that contain the target and for pixels that do not contain the target. The statistics of the input can then be propagated through the detection system to obtain a statistical description of the output, for both cases. The predicted output, and performance, is ultimately dependent on the statistical model used for each case and on the algorithm.
1.2 Background

Suppose we have a hyperspectral sensor with \( p \) spectral channels. Every pixel then has \( p \) measurements; which can be conveniently represented as a vector in \( p \) dimensions; we will use the notation \( \mathbf{x} \in \mathbb{R}^p \) to mean that the vector \( \mathbf{x} \) has \( p \) dimensions. Generally, we will use \( \mathbf{x} \) to represent the reflectance of each pixel in the image, and \( \mathbf{s}_t \) for the reflectance signature of the target material\(^1\) (\( \mathbf{s}_t \in \mathbb{R}^p \)). Before attaching a statistical model to these two quantities, we will look at a simple physical model for a pixel.

1.2.1 The Linear Mixing Model and Subpixel Targets

Intuitively, every pixel in an image corresponds to an area of the earth; for simplicity we will use a square region, but the reality is much more complicated. The calculated reflectance of that pixel \( \mathbf{x} \) is some function of the reflectances of the materials in the pixel. A very common model, which we use here, assumes linear mixing of the reflectances of the materials in the pixel [1]. That is, the abundance of each material in the pixel determines its contribution to the total reflectance of the pixel. A reasonable assumption is that the abundance of each material is approximately the fraction of the pixel that the material inhabits. Therefore, we will use the term **fill fraction** instead of abundance and use the symbol \( \alpha \) for this quantity.

\(^1\)Spectral library reflectance signatures usually have much higher spectral resolution than a hyperspectral sensor and must be downsampled to the same channels as the sensor.
This model can be summarized mathematically as follows; assume there are \( N \) materials in a scene, with reflectance signatures \( s_1, s_2, \ldots, s_i, \ldots, s_N \). Letting \( \alpha_i \) be the fill fraction of the \( i \)th material for a particular pixel, the reflectance of a pixel \( x \) can be written as

\[
x = \sum_{i=1}^{N} \alpha_i s_i
\]

where we have defined \( \alpha_i \) as

\[
\alpha_i = \frac{\text{Area occupied by material}}{\text{Area of pixel}}
\]

Using this definition, the \( \alpha_i \) must satisfy two constraints for all pixels:

\[
\sum_{i=1}^{N} \alpha_i = 1 \\
\alpha_i \geq 0, i = 1, 2, \ldots, N
\]

Together, these equations are known as the linear mixing model (LMM), while the last two equations are known as the sum-to-one and non-negativity constraints respectively or collectively as mixing constraints. In practice the \( s_i \) and \( \alpha_i \) are not known and are difficult to estimate well. Trying to find the \( s_i \) and \( \alpha_i \) for every pixel in a scene is called unmixing; a survey of the various unmixing techniques can be found in [12]. This is inherently a very difficult problem and is not our goal here; instead we utilize a simplified LMM consisting of two materials.

For the purposes of detection it is very difficult to utilize the full LMM, instead we opt for a relatively simple statistical model that uses only a background reflectance \( \mu_b \) and a target signature \( s_t \). Consider a pixel that is completely filled by the target material. From our mixing constraints, the reflectance of this pixel should be \( s_t \); this type of target is called a full pixel target. Pixels that contain the target but are not completely filled by the target are called subpixel targets or mixed pixels, and are pixels that contain some combination of the target signature and other material signatures. For simplicity we call all the non-target materials in the scene the background and we call this simplified model
the target-background model. Assuming there is a single background spectrum $\mu_b$, a mixed pixel $x_m$ becomes

$$x_m = \alpha s_t + (1 - \alpha)\mu_b \quad (1.1)$$

This model is illustrated for an unresolved target (red) in a uniform background (green) in Figure 1.3. However, without any variability, this model is too restrictive; allowing the background and target to follow a statistical model is our next step.

![Figure 1.3](image-url)  

**Figure 1.3:** (a) Illustration of unresolved target and (b) resulting pixel fill fractions.

### 1.2.2 Preliminary Signal Model

One major factor that our LMM is missing is a noise component. Suppose that the background of a scene is a single material $\mu_b$, but because of surface geometry, inherent variations in the material’s reflectance and other factors, there is uncertainty about a pixel’s reflectance. This uncertainty can be modeled as an additive noise vector $n_b$, which follows some statistical distribution. Then the background of the scene $x_b$ becomes

$$x_b = \mu_b + n_b$$
where the subscript $b$ denotes background. The two noise models that we will use for $n_b$ are the multivariate normal (MVN) or Gaussian and multivariate $t$ (MVT) distributions (see Appendix B). If $n_b$ follows a zero mean MVN distribution with covariance matrix $\Sigma_b$, then we write

$$n_b \sim \mathcal{N}(0, \Sigma_b)$$

where the symbol $\sim$ is read “is distributed as” and means that the random vector on the left follows the statistical distribution on the right. The MVT distribution has an additional parameter $\nu$ that controls the “tail-heaviness” of the distribution. If $n_b$ follows this distribution with “dispersion” matrix $\Sigma_b$, we write

$$n_b \sim \mathcal{t}_\nu(0, \Sigma_b)$$

Here, we use the term “dispersion” because the true covariance of the vector $n_b$ is no longer $\Sigma_b$. Using the notation $E[z]$ to denote the expected value of the vector $z$ [23], the covariance matrix of $z$ (assuming $z$ is zero mean) is defined as

$$\text{Cov}[z] = E[zz^T]$$

It can be shown [6] that if $z \sim \mathcal{t}_\nu(0, R)$ with $\nu > 2$ and positive definite $R$, then

$$\text{Cov}[z] = \frac{\nu}{\nu - 2} R$$

If, for the MVT and MVN distributions we want the same covariance, then we need to be careful when selecting the matrix $R$ above.

The background vector $x_b$, with the addition of the noise vector $n_b$, becomes

$$x_b \sim \mathcal{N}(\mu_b, \Sigma_b) \text{ or } x_b \sim \mathcal{t}_\nu(\mu_b, \Sigma_b)$$

(1.2)

Though the MVN distribution is a good model for the majority of pixels in a hyperspectral background, the MVT distribution can be used to simulate outliers and has been shown to fit hyperspectral backgrounds better than the MVN model [18].

While the background may follow a heavy tailed model, we assume that the target follows
a MVN distribution. Intuitively, the target should be less variable than the background, and should not have large outliers. In addition, there is no guarantee that the target varies in the same way as the background. Therefore, adding a zero mean MVN noise vector with covariance matrix $\Sigma_t$ to the target $s_t$ (denoting the resulting vector as $x_t$) we have

$$x_t \sim \mathcal{N}(s_t, \Sigma_t)$$  \hspace{1cm} (1.3)

In this model the mean vector for a full pixel target is the library spectrum $s_t$, which agrees with the LMM given earlier.

Statistically, in the target-background model we have $x_b$ for the background reflectance and $x_t$ for the target reflectance; applying our linear mixing assumption for mixed pixels $x_m$ we have

$$x_m = \alpha x_t + (1 - \alpha) x_b$$

Finally a mixed pixel can be described statistically as

$$x_m \sim \mathcal{N}(\mu_m(\alpha), \Sigma_m(\alpha))$$  \hspace{1cm} (1.4a)

$$\mu_m(\alpha) = \alpha s_t + (1 - \alpha) \mu_b$$  \hspace{1cm} (1.4b)

$$\Sigma_m(\alpha) = \alpha^2 \Sigma_t + (1 - \alpha)^2 \Sigma_b$$  \hspace{1cm} (1.4c)

where $\mu_m(\alpha)$ is the mean of the mixed pixel and $\Sigma_m(\alpha)$ is the covariance of the mixed pixel. We note that the mean of a mixture at a particular fill fraction always lies on the line between the background mean $\mu_b$ and the target mean $s_t$. This is illustrated in Figure 1.4 for a MVN target and background with unequal covariances in two dimensions; in general these distributions may be more complicated. In Chapter 2 we explore this simplified model further, while in Chapter 5 we allow the background to be multimodal, which is closer to the behavior of real hyperspectral data.
Figure 1.4: Illustration of target, background and mixed pixel distributions using the LMM.

1.2.3 Hypothesis Testing

Developing a statistical description for the target, background and mixed pixels allows us to easily formulate our detection problem in terms of a hypothesis test. In fact, any detection problem can be posed as a hypothesis testing problem where there are two competing hypotheses: $H_0$ and $H_1$; for each input we want to determine which hypothesis was active when the input was generated. For simplicity and clarity we have two cases:

$H_0$: Target Absent (Background only)

$H_1$: Target Present (Target plus background)

For each hypothesis we assume the input follows some statistical model. For example, using the target-background model of Section 1.2.2 one set of hypotheses is

$$H_0 : \mathbf{x} \sim \mathcal{N}(\mu_b, \Sigma_b)$$

$$H_1 : \mathbf{x} \sim \mathcal{N}(\mathbf{s}_t, \Sigma_t)$$

Using hypothesis tests, a variety of detectors can be be developed (see [30] and [25] for thorough discussions of detectors and detection theory); the resulting detectors satisfy Likelihood
Ratio Tests (LRTs):

\[ L(x) = \frac{f(x|H_1)}{f(x|H_0)} \]  

(1.5)

where \( f(x|H_i) \) is the conditional probability density function (PDF) of the input \( x \) given hypothesis \( H_i \) (\( i = 0, 1 \)). When parameters of the conditional density are assumed unknown we can use a generalized likelihood ratio test (GLRT) instead. The GLRT simply replaces unknown parameters with their maximum likelihood estimates. The two most important detectors can be developed in this way: the matched filter as a LRT and ACE as a GLRT [13]. In Section 1.3, we will derive these detectors in this fashion.

A major feature of these types of detectors is that they take a single vector as input, and produce a single number. This score indicates how close the input is to one hypothesis or the other. After producing a score for each input vector, a user must set a threshold to declare which inputs were generated when \( H_1 \) was active, and which were generated when \( H_0 \) was active. Given an input \( x \) and a threshold \( \eta_0 \) this operation is as follows

\[ \text{If } L(x) \leq \eta_0 \rightarrow \text{Declare that } x \text{ was generated under } H_0 \]

\[ \text{If } L(x) > \eta_0 \rightarrow \text{Declare that } x \text{ was generated under } H_1 \]

Setting a threshold divides the number line into two regions: a declared target region, and a declared background region. The probability of misclassification is the key performance measure for this type of algorithm.

Determining the misclassification rate for detection algorithms is typically done by finding the conditional density of \( L(.) \) under each hypothesis. Letting \( \eta = L(x) \), we try to find analytical expressions for \( f(\eta|H_0) \) and \( f(\eta|H_1) \). The measures of misclassification we will use are called “probability of false alarm” and “probability of detection.” The probability of false alarm (\( P_{FA} \)) is the probability that a background pixel (generated under \( H_0 \)) is incorrectly declared a target; the probability of detection (\( P_D \)) is the probability that a target pixel is correctly declared a target. These quantities are defined in terms of the conditional
probability density functions (PDFs) of the output $\eta$ as

$$P_D(\eta) = 1 - \int_{-\infty}^{\eta} f(\eta'|H_1) d\eta'$$

$$P_{FA}(\eta) = 1 - \int_{-\infty}^{\eta} f(\eta'|H_0) d\eta'$$

where $\eta'$ is a dummy variable. The integrals on the right-hand side are also known as cumulative distribution functions (CDFs); the CDF of a random variable is simply a running integral of its PDF. In terms of the conditional CDFs, $F_y(\eta|H_1)$ and $F_y(\eta|H_0)$, we can rewrite these equations as

$$P_D(\eta) = 1 - F_y(\eta|H_1)$$

$$P_{FA}(\eta) = 1 - F_y(\eta|H_0)$$

The quantity $1 - F_y(\eta)$ is also known as the probability of exceedance and will be used throughout this thesis. An example of two conditional densities of $L(.)$ is shown in Figure 1.5; setting a threshold (black) and computing the area under the curves we get the $P_D$ and $P_{FA}$ for that threshold; in the figure these areas are marked $P_D$ and $P_{FA}$. We can set a threshold by choosing a minimum acceptable false alarm rate (FAR) and then finding the threshold that gives that rate. Setting a threshold this way then determines the probability of detection. This is generally how we will select thresholds, but do note that the probability of detection will be determined by the separation between the target and background densities, and that the higher the FAR the higher the probability of detection.

To visualize the tradeoff between $P_D$ and $P_{FA}$ a receiver operation characteristic (ROC) curve can be constructed by plotting $P_D(\eta)$ versus $P_{FA}(\eta)$ for a range of thresholds $\eta$. An example ROC curve is shown in Figure 1.6 for the two densities in Figure 1.5. Qualitatively, the closer the black curve is to the upper left corner of the plot, the better the performance is. ROC curves are useful for comparing different detection algorithms as they provide a threshold-free way to compare their performance.

The probability of exceedance and ROC curves precisely describe a detector in theory, but
in practice a finite number of observations are available. To see whether a theoretical CDF or ROC agrees with an observed CDF, an estimate of the CDF can be obtained from the observations. For example, the CDF $F_y(\eta_0)$ tells us exactly the fraction of all observations that should be less than a given threshold $\eta_0$. Calculating what fraction of the observations are less than the threshold yields an estimate $\hat{F}(\eta_0)$ of the CDF. This estimate takes the form

$$\hat{F}(\eta_0) = \frac{\# \text{ of observations with } y \leq \eta_0}{\text{Total } \# \text{ of pixels}}$$

where $y$ is the detector output (observation). Similarly, the quantity $1 - \hat{F}(\eta_0)$ is an estimate of probability of exceedance. When the input samples be labeled as $H_0$ or $H_1$ the estimated probability of exceedance is also an estimate of $P_D$ and $P_{FA}$. In HSI, labeled target pixels are called ground-truthed targets. Using the detector responses for these labeled target pixels the probability of detection can be estimated. Similarly, by defining a set of background pixels, the probability of false alarm can be estimated. Combining these two estimates, an estimated ROC curve can be obtained.
Figure 1.5: Illustration of detector densities under $H_0$ and $H_1$. Setting a threshold (black) determines the probability of false alarm $P_{FA}$ and probability of detection $P_D$.

Figure 1.6: Illustration of a ROC curve (black) using the densities in Figure 1.5. Selecting a probability of false alarm of $10^{-3}$ results in a probability of detection of 0.46 (red).
1.3 Introduction to Signal Detection Algorithms

In this section we derive and describe the two most common detectors used in HSI: the matched filter (MF) and the adaptive cosine estimator (ACE). Both detectors arise under the hypotheses:

\[ H_0 : x \sim \mathcal{N}(0, \sigma^2 \Sigma_b) \]
\[ H_1 : x \sim \mathcal{N}(as, \sigma^2 \Sigma_b) \]

where \( a \) and \( \sigma \) are scalars, \( s \) is the target signature and \( \Sigma_b \) is the background covariance matrix. To use hypotheses similar to the target-background model of section 1.2.2, we have to assume that the input \( x \) has been de-meaned, and the target signature has been de-meaned making \( s = s_t - \mu_b \). In our discussion here, the variable \( a \) is not to be confused with fill fraction \( \alpha \); \( a \) is a scaling like \( \alpha \), but \( a \) is not constrained between 0 and 1, and it does not affect the covariance matrix under either hypothesis. To understand how the MF and ACE arise from this hypothesis test we derive them following [26] and [14]. Starting with the MF we will develop notation and terminology that will simplify our discussions of other detectors.

1.3.1 A Derivation of the Matched Filter (MF)

The matched filter is probably the most well known and best understood detector. It is also the easiest to derive. Recall that the likelihood ratio of Equation 1.5 is a ratio of conditional densities; for our current hypotheses the densities are both MVN where one is zero-mean and the other is not. Assuming both \( a \) and \( \sigma \) are known, the densities are known and the LRT becomes

\[ L(x) = \frac{f(x|H_1)}{f(x|H_0)} \]
where the conditional densities \( f(x|H_1) \) and \( f(x|H_0) \) are

\[
\begin{align*}
\frac{\gamma}{|\Sigma_b|^{1/2}} & \frac{1}{\pi^r} \exp \left( - \frac{1}{2\sigma^2} x^T \Sigma_b^{-1} x \right) \\
\frac{\gamma}{|\Sigma_b|^{1/2}} & \frac{1}{\pi^r} \exp \left( - \frac{1}{2\sigma^2} (x - \mu)^T \Sigma_b^{-1} (x - \mu) \right)
\end{align*}
\]

We can apply a monotonic function to \( L(x) \) without not change the performance of the detector \([5]\). Taking the logarithm (a monotonic function) of \( L(x) \) results in the log likelihood ratio, which we denote \( L'(x) \). Substituting the conditional densities into the LRT, the leading terms cancel and the exponents combine, resulting in

\[
L'(x) = - \frac{1}{2\sigma^2} (x - \mu)^T \Sigma_b^{-1} (x - \mu) + \frac{1}{2\sigma^2} x^T \Sigma_b^{-1} x
\]

Dropping leading constants, expanding and canceling like terms, yields

\[
L'(x) = s^T \Sigma_b^{-1} x - s^T \Sigma_b^{-1} s
\]

Applying the scaling \( 1/\sqrt{s^T \Sigma_b^{-1} s} \) and dropping the right-most term, we obtain the MF

\[
y_{MF} = \frac{s^T \Sigma_b^{-1} x}{\sqrt{s^T \Sigma_b^{-1} s}}
\]

This definition can be simplified by introducing the whitening transformation. Assuming that \( \Sigma_b \) is positive definite, and has a square root matrix \( \Sigma_b^{1/2} \) such that \( \Sigma_b = \Sigma_b^{1/2} \Sigma_b^{1/2} \). The inverse of the square root matrix \( \Sigma_b^{-1/2} \) is known as a **whitening matrix** or whitening transformation; multiplying \( x \) and \( s \) by this matrix yields the whitened vectors \( \tilde{x} \) and \( \tilde{s} \):

\[
\tilde{s} = \Sigma_b^{-1/2} s \quad \text{and} \quad \tilde{x} = \Sigma_b^{-1/2} x
\]

The input under each hypothesis after whitening becomes

\[
H_0 : \tilde{x} \sim \mathcal{N}(0, \mathbf{I})
\]

\[
H_1 : \tilde{x} \sim \mathcal{N}(a\tilde{s}, \mathbf{I})
\]
where \( \mathbf{I} \) is a \( p \) dimensional identity matrix. This allows us to write the MF as

\[
y_{MF} = \frac{\mathbf{x}^T \tilde{s}}{\sqrt{\tilde{s}^T \tilde{s}}}
\]

where the numerator is simply the dot product of the whitened input with the whitened target, and the denominator is the length of the whitened target vector. The de-meaning and whitening operations are shown in succession in Figure 1.7.

![Figure 1.7: Illustration of de-meaning and whitening in 2-D with MVN distributed background (green) and target (red). Distribution of data (a) at the input, (b) after de-meaning, and (c) after de-meaning and whitening.](image)

1.3.2 Statistics of the Matched Filter

There are many different versions and ways to define the MF; in order of decreasing verbosity, we can define the MF as

\[
y_{MF} = \frac{s^T \Sigma_b^{-1} x}{\sqrt{s^T \Sigma_b^{-1} s}} \quad \text{(1.6)}
\]

or in terms of the whitened vectors \( \tilde{s} \) and \( \tilde{x} \) as

\[
y_{MF} = \frac{\tilde{s}^T \tilde{x}}{\sqrt{\tilde{s}^T \tilde{s}}} \quad \text{(1.7)}
\]

or for brevity as

\[
y_{MF} = \mathbf{h}^T \tilde{x} \quad \text{(1.8)}
\]
where \( h = \frac{s}{\sqrt{s^T s}} \). In the last case, the vector \( h \) is known as the matched filter vector. We will generally use Equation 1.8 because it is extremely concise, and very general.

Note that we can scale the MF arbitrarily but the definitions above are beneficial for this thesis because they facilitate the use of statistical relations, but others may choose a scale that is appropriate with no change in performance. As an example, we will use a different scaling to use the MF as a fill fraction estimator (FFE). We define the FFE as

\[
y_{\text{FFE}} = \frac{1}{\sqrt{s^T s}} y_{\text{MF}}
\]

where \( y_{\text{FFE}} \) is an estimate of the fill fraction \( \alpha \).

To obtain basic statistical results for the MF, we assume that we have subtracted \( \mu_b \) from the data and set \( \sigma = 1 \), resulting in the hypotheses

\[
H_0 : x \sim \mathcal{N}(0, \Sigma_b) \quad (1.10a)
\]
\[
H_1 : x \sim \mathcal{N}(a\bar{s}, \Sigma_b) \quad (1.10b)
\]

Following the notation of Scharf [14], the mean of the output of the matched filter defined in Equation 1.8 is called the Voltage Signal to Noise Ratio (VSNR), and its square is the Signal to Noise Ratio (SNR), defined as

\[
\text{VSNR} = \mu_{\text{MF}} \quad (1.11a)
\]
\[
\text{SNR} = \mu_{\text{MF}}^2 \quad (1.11b)
\]

where \( \mu_{\text{MF}} = a h^T \bar{s} = a \sqrt{\bar{s}^T \bar{s}} \). Immediately we see that using the FFE, the mean of the output becomes the scaling of the target mean \( a \). For the hypotheses in Equations 1.10a and 1.10b the distribution of the MF becomes [17]

\[
H_0 : y_{\text{MF}} \sim \mathcal{N}(0, 1) \quad (1.12a)
\]
\[
H_1 : y_{\text{MF}} \sim \mathcal{N}(\text{VSNR}, 1) \quad (1.12b)
\]

The separation between the means of the target-absent and target-present distributions in the output is controlled by the VSNR; a high VSNR results in very good performance, while
a low VSNR results in bad performance.

1.3.3 The Adaptive Cosine Estimator (ACE)

Although the MF is a simple and ubiquitous detector, our main focus will be the Adaptive Cosine Estimator (ACE) or Adaptive Coherence Estimator. This is a simple extension of the MF where we compute the MF value, and then normalize by the length of $\tilde{x}$. This statistic is simply the cosine of the angle between $\tilde{x}$ and $\tilde{s}$, which we designate $\theta$; the MF and ACE statistics are shown in Figure 1.8 when the input is already white. We see that ACE ($y_{ACE}$) is equal to $\cos(\theta)$, and is defined as

$$ y_{ACE} = \frac{s^T \tilde{x}}{\sqrt{s^T \tilde{s} \sqrt{\tilde{x}^T \tilde{x}}} } $$

In the literature, ACE usually refers to cosine-squared ($\cos(\theta)^2$), but for simplicity we will use ACE to refer to cosine. The term “adaptive” in ACE generally refers to estimating the covariance matrix of the background $\Sigma_b$. Throughout this thesis, for statistical purposes, we assume covariance matrices are not random quantities, even though we have estimated them, but we will not make a distinction in terminology. This may lead to some confusion because in the literature the term adaptive has been applied to detectors where the output has been squared. For example, squaring the MF results in the so-called Adaptive MF, even though the covariance matrix was estimated for the MF too. In addition to the adaptive and non-adaptive variations, there are squared and subspace versions; we will cover these in Section 3.1.2 and summarize them in Table 3.2. Derivations of these versions of the detectors are given in [9], [26] and [25], but we follow the derivation in [26] for ACE.

In deriving the MF, we assumed that the parameters $a$ and $\sigma$ were known, but if $\sigma$ is unknown we may seek a generalize likelihood ratio test (GLRT) [30]. In this GLRT, we replace $\sigma^2$ with its maximum likelihood estimate $\hat{\sigma}^2$ under each hypothesis.
Returning to our likelihood ratio and canceling like terms we have
\[ L(x) = \left( \frac{\hat{\sigma}_1^2}{\hat{\sigma}_0^2} \right)^{-p/2} \exp \left\{ -\frac{1}{2\hat{\sigma}_1^2} (x - as)^T \Sigma^{-1} (x - as) + \frac{1}{2\hat{\sigma}_0^2} x^T \Sigma^{-1} x \right\} \] (1.13)

The maximum likelihood estimate (MLE) of the variance for each hypothesis can be shown to be
\[ \hat{\sigma}_1^2 = \frac{1}{p} (x - as)^T \Sigma^{-1} (x - as) \]
\[ \hat{\sigma}_0^2 = \frac{1}{p} x^T \Sigma^{-1} x \]

These estimates are obtained by differentiating \( f(x|H_1) \) and \( f(x|H_0) \) with respect to \( \sigma^2 \), setting the derivative to 0, and solving for \( \sigma^2 \). Substituting these estimates into Equation 1.13 and dropping the constant right-hand term, yields
\[ L(x) = \left( \frac{\hat{\sigma}_1^2}{\hat{\sigma}_0^2} \right)^{-p/2} \]
and an equivalent statistic is
\[ L''(x) = \left( \frac{\hat{\sigma}_0^2}{\hat{\sigma}_1^2} \right) \]

Figure 1.8: Illustration of the (a) MF and (b) ACE in two dimensions with an uncorrelated MVN distributed background (green) and target (red). For demonstration, the input is already white, \( x \sim \mathcal{N}(s_t, \mathbf{I}) \), making whitening unnecessary.
Substituting $\hat{\sigma}_1^2$ and $\hat{\sigma}_0^2$ we have

$$L''(x) = \frac{(x - as)^T \Sigma^{-1}(x - as)}{x^T \Sigma^{-1} x} \quad (1.14a)$$

Distributing $\Sigma^{-1}$ using its square root matrix $\Sigma^{-1/2}$ we have

$$L''(x) = \frac{(\tilde{x} - a\tilde{s})^T (\tilde{x} - a\tilde{s})}{\tilde{x}^T \tilde{x}} \quad (1.14b)$$

At this point we assume that the scaling $a$ is unknown, and substitute the MLE $\hat{a}$. When $a$ is unconstrained the MLE is

$$\hat{a} = \frac{x^T \Sigma^{-1}s}{s^T \Sigma^{-1} s} = \frac{\tilde{x}^T \tilde{s}}{\tilde{s}^T \tilde{s}}$$

when we constrain $a \geq 0$, the MLE becomes: $\hat{a} = \max \left[0, \frac{\tilde{x}^T \tilde{s}}{\tilde{s}^T \tilde{s}}\right]$ (see [26] section V). Substituting the unconstrained estimate into Equation 1.14b yields

$$L''(x) = \frac{(\tilde{x} - \tilde{s}\tilde{s}^T \tilde{x})^T (\tilde{x} - \tilde{s}\tilde{s}^T \tilde{x})}{\tilde{x}^T \tilde{x}}$$

The term resulting from $\hat{a}\tilde{s}$ can be written using projection matrix notation (see Appendix A) as

$$\tilde{s}\tilde{s}^T \tilde{x} = P_{\tilde{s}} \tilde{x}$$

Factoring $\tilde{x}$ the numerator of $L''(x)$, yields

$$\tilde{x}^T (I - P_{\tilde{s}})(I - P_{\tilde{s}})\tilde{x} = \tilde{x}^T P_{\tilde{s}}^\perp \tilde{x}$$

where $P_{\tilde{s}}^\perp$ is known as an orthogonal projection matrix. Finally, we have

$$L''(x) = \frac{\tilde{x}^T P_{\tilde{s}}^\perp \tilde{x}}{\tilde{x}^T \tilde{x}} = \sin^2(\theta)$$

where we are measuring the angle between $\tilde{s}$ and $\tilde{x}$. This is monotonically related to cosine squared

$$\cos^2(\theta) = \frac{\tilde{x}^T P_{\tilde{s}} \tilde{x}}{\tilde{x}^T \tilde{x}}$$

33
which is the classic definition of ACE found in the literature. Since we assume $\alpha$ is non-negative, we constrain the scaling $a$ to be non-negative, so the log-likelihood function becomes

\[
L''(x) = \begin{cases} 
\frac{\tilde{x}^T P \tilde{s}}{\tilde{x}^T \tilde{x}} & \text{for } \hat{a} \geq 0 \\
0 & \text{for } \hat{a} < 0 
\end{cases}
\]

In this case, $L''(x)$ is a monotonic function of

\[
L''(x) = \begin{cases} 
\sqrt{\frac{\tilde{x}^T P \tilde{s}}{\tilde{x}^T \tilde{x}}} & \text{for } \hat{a} \geq 0 \\
0 & \text{for } \hat{a} < 0 
\end{cases}
\]

and when $s$ is a single vector the first term becomes

\[
y_{ACE} = \sqrt{\frac{\tilde{x}^T P \tilde{s}}{\tilde{x}^T \tilde{x}}} = \frac{\tilde{s}^T \tilde{x}}{\sqrt{\tilde{x}^T \tilde{x} \tilde{s}^T \tilde{s}}}
\]

which is equivalent to the cosine between $\tilde{x}$ and $\tilde{s}$, as expected. This is the definition of ACE that we will use throughout this thesis, but we do not condition the output on the sign of $\hat{a}$.

### 1.3.4 The Statistics of ACE

As with the MF, there are many equivalent representations for ACE. The most important versions are the cosine, cotangent and the t-detector, which we define only in terms of the whitened and de-meaned vectors $\tilde{x}$ and $\tilde{s}$ as

\[
y_{ACE} = \sqrt{\frac{\tilde{s}^T \tilde{x}}{\tilde{s}^T \tilde{s} \sqrt{\tilde{x}^T \tilde{x}}}} \tag{1.15a}
\]

\[
y_{cot} = \sqrt{\frac{\tilde{s}^T \tilde{P}_s \tilde{x}}{\tilde{s}^T \tilde{s} \sqrt{\tilde{x}^T \tilde{P}_s \tilde{x}}}} \tag{1.15b}
\]

\[
y_t = \sqrt{p - 1} \sqrt{\frac{\tilde{s}^T \tilde{P}_s \tilde{x}}{\tilde{s}^T \tilde{s} \sqrt{\tilde{x}^T \tilde{P}_s \tilde{x}}}} \tag{1.15c}
\]

We note that these three detectors are monotonically related; in particular, the cosine and cotangent are both ratios of sides from the same triangle. It can be shown that the cosine is related monotonically to the cotangent by

\[
y_{ACE} = \frac{y_{cot}}{\sqrt{1 + y_{cot}^2}}
\]

meaning they are equivalent detection statistics.
Interestingly, the t-detector has this name because it follows a t-distribution under certain circumstances. Although, these three detectors are equivalent, only the t-detector follows a well known distribution; so, we will use this version when discussing statistical distributions. The distributions of the t-detector for the hypotheses of Equations 1.10a and 1.10b are \[14\]

\[H_0 : y_t \sim t_{p-1}(0,1) \] (1.16a)

\[H_1 : y_t \sim t_{p-1}(VSNR,1) \] (1.16b)

Under both hypotheses, the detector follows a univariate t-distribution (see Appendix B.3 or [10]). We note that this distribution only occurs when the detector is designed with the exact covariance matrix of the input \(x\). When the design covariance matrix is different from the input covariance matrix, we say there is covariance mismatch. For cases where there is arbitrary mismatch, we opt to use the techniques of Chapter 5 for computing cumulative density functions (CDFs) of these detectors.

1.4 Thesis Organization

In this thesis we have reviewed some statistical models for HSI, and described the principal detectors that we use for material detection. The goal for the remainder of the thesis is to develop and evaluate techniques that can be used for performance prediction in HSI. In essence, we want to have a model for the background and a model for mixed pixels of any fill fraction, that together can be used to estimate the performance of our detection algorithms.

We build these models and techniques in increments. In Chapter 2 we examine the effects of various model parameters for the target-background model of Section 1.2.2. In Chapter 5 we expand our discussion of detection algorithms to include the squared and subspace versions, and we discuss false alarm mitigation techniques. In Chapter 4 we develop methods for computing CDFs of ACE when there is arbitrary covariance mismatch. Chapter 5 extends the target-background model to a multimodal background model and we describe how this
can be used to estimate detection performance. In Chapter 6 we apply the multimodal background model and the techniques of Chapter 4 to real data with ground-truthed targets. Finally, we explore the effects of target parameters on the prediction and attempt to put into perspective false alarm mitigation techniques with the prediction model.
Chapter 2

Performance of the MF and ACE in Unimodal Clutter

In deriving the MF and ACE, we assumed that the target and background were pretty well behaved. However, in reality the data will not exactly fit the assumptions, although it may be close to the truth. Deviations from the ideal should be expected and will most likely have a detrimental effect on detection performance.

In this chapter we explore the effects that various parameters have on detection performance. By varying the distribution parameters (the fill fraction, target covariance matrix and tail-heaviness) we can compare the MF and ACE quantitatively and understand qualitatively what parameters are most important for detection performance.

2.1 Unimodal Test Cases

Because the true number of parameters that can be varied, is extremely large, we have chosen five different target-background pairs as summarized in Table 2.1. For each pair of hypotheses we will use a variation of the statistical model used to derive the MF and ACE. First, in Test 1 we will examine how the dimension $p$ effects performance. Interestingly it
will not affect the MF but will affect ACE.

In Test 2 we see what effect the choice of target covariance has. Since, the target is usually a single material, the variability should usually be less than the variability of any background materials. We use a scaled version of the background covariance $\Sigma_b$ as the target covariance and examine the effects. In Test 3 we introduce fill fraction, but use the same covariance matrix for both the background and target. In this scenario, fill fraction simultaneously shifts the mean of the target and scales the target covariance matrix. Therefore, at any fill fraction we end up with a case of Test 2.

While Test 2 and Test 3 examine the target distribution, in Test 4 we replace the MVN background with a MVT background and vary the tail-heaviness parameter. This will greatly impact the performance of the MF while it will have no impact on the performance of ACE.

In practice we may not be able to measure or even estimate the target covariance matrix, which results in so-called covariance mismatch. In Test 5, we try to simulate covariance mismatch by selecting a target covariance matrix that is slightly different from the background covariance matrix.

<table>
<thead>
<tr>
<th>Test #</th>
<th>$H_0$</th>
<th>$H_1$</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$\mathcal{N}(\mu_b, \Sigma_b)$</td>
<td>$\mathcal{N}(s_t, \Sigma_b)$</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>$\mathcal{N}(\mu_b, \Sigma_b)$</td>
<td>$\mathcal{N}(s_t, \sigma^2 \Sigma_b)$</td>
<td>a</td>
</tr>
<tr>
<td>3</td>
<td>$\mathcal{N}(\mu_b, \Sigma_b)$</td>
<td>$\mathcal{N}(\mu_m, \gamma^2 \Sigma_b)$</td>
<td>a,b,c</td>
</tr>
<tr>
<td>4</td>
<td>$t_\nu(\mu_b, \Sigma_b)$</td>
<td>$\mathcal{N}(s_t, \Sigma_b)$</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>$\mathcal{N}(\mu_b, \Sigma_b)$</td>
<td>$\mathcal{N}(\mu_m, \Sigma_m)$</td>
<td>a,b,c,d</td>
</tr>
</tbody>
</table>

$a \sigma > 0, 1 \geq \alpha \geq 0$

$b \gamma^2 = \{\alpha^2 + (1 - \alpha^2)\}$

c $\mu_m = \alpha s_t + (1 - \alpha) \mu_b$

$d \Sigma_m = \alpha^2 \Sigma_t + (1 - \alpha)^2 \Sigma_b, \Sigma_t \neq \sigma^2 \Sigma_b$

**Table 2.1:** Summary of simple unimodal test cases.
2.2 Test Case 1: Effect of Dimension

While the matched filter is unaffected by data dimension, the performance of the t-detector directly depends on the dimension of the input. In fact, the performance increases with increase in dimension $p$. To demonstrate the effect of $p$ on performance we use the hypotheses:

$$H_0 : x \sim \mathcal{N}(0, \Sigma_b)$$
$$H_1 : x \sim \mathcal{N}(s, \Sigma_b)$$

with $s$ set so that the VSNR of Equation 1.11a is 5. For these hypotheses, using Equation 1.12b, the output of the matched filter is

$$H_0 : y_{MF} \sim \mathcal{N}(0, 1)$$
$$H_1 : y_{MF} \sim \mathcal{N}(5, 1)$$

and using Equation 1.16b the output of the t-detector is

$$H_0 : y_t \sim t_{p-1}(0, 1)$$
$$H_1 : y_t \sim t_{p-1}(5, 1)$$

and $t_{p-1}(.)$ is defined in Appendix B.3. In Figure 2.1 we demonstrate that in this case the MF always outperforms the t-detector and equivalently ACE, but that the performance of ACE is affected by the dimension of the input data, while the MF is not. Finally we note that the performance of the t-detector converges to that of the MF as $p \to \infty$. 

39
2.3 Test Cases 2: Effect of Covariance Scaling

In this case, the target has less variability than than the background. From the LMM, mixed pixels will usually have less variability than the background. The current test is illustrated in Figure 2.2 (a). In (b) we have Test 3, but we see that the covariance of a mixed pixel (purple) will be scaled, even though the covariance of full pixels (red) is the same as the background. The effect is more exaggerated in (c) where the full pixel covariance is smaller than the background covariance.

The distribution of the input under the scaled covariance model is:

\[ H_0 : x \sim \mathcal{N}(\mu_b, \Sigma_b) \]
\[ H_1 : x \sim \mathcal{N}(s_t, \sigma^2 \Sigma_b) \]
Figure 2.2: Example distributions where ($\Sigma_t = \sigma^2 \Sigma_b$): (a) Target distribution is scaled version of background; (b) target and background share the same scale but mixed pixels do not; (c) full pixels and mixed pixels are scaled down.

$s$ was chosen to have a VSNR of 5. The distribution of the output under $H_0$ is unchanged from the previous experiment, but when the target is present we have

\[
H_1 : \ y_{MF} \sim \mathcal{N} \left(5, \sigma^2 \right)
\]

\[
H_1 : \ y_t \sim \mathcal{t}_{p-1} \left(\frac{5}{\sigma^2}, 1 \right)
\]

We assume that the variability of the target is less than the variability of the background so we only consider the case where $0 < \sigma < 1$. This improves performance for both detectors, but has a greater effect on ACE than the MF.
Figure 2.3: Test Case 2, ROC curves when $\Sigma_t$ is $\sigma^2 \Sigma_b$; i.e. the target covariance matrix is a scaled version of the background covariance matrix.
2.4 Test Case 3: Effect of Linear Mixing

To explore the effect of fill fraction, we do not use a normal ROC curve, since we would need one for every fill fraction. Instead, by fixing a particular false alarm rate (FAR) and varying fill fraction (α in Table 2.1), we can see how probability of detection degrades as target size decreases. For simplicity we set $\Sigma_t = \Sigma_b$, making the test hypotheses:

$$H_0 : x \sim N(\mu_b, \Sigma_b)$$
$$H_1 : x \sim N(\mu_m, \gamma^2 \Sigma_b)$$

where $\gamma^2 = \alpha^2 + (1 - \alpha)^2$ and $\mu_m = \alpha s_t + (1 - \alpha) \mu_b$. It can be shown that the detectors are:

$$H_1 : y_{MF} \sim N(\alpha \gamma \sigma, \sigma^2)$$
$$H_1 : y_t \sim t_{p-1} \left( \frac{\alpha}{\gamma}, 1 \right)$$

and we get the distributions under $H_0$ by setting $\alpha = 0$. A VSNR of 5 was selected and the fill fraction varied from 0 to 1. Figure 2.4 shows probability of detection with the $P_{FA}$ set to $10^{-3}$, $10^{-4}$, and $10^{-5}$; this was done in 5, and 75 dimensions for both the MF and ACE.

Figure 2.4 illustrates how the performance of the $t$-detector exceeds the MF for many fill fractions as the dimension of the data increases. It is also apparent that when the dimension of the data is low (eg, $p = 5$) the MF detector is significantly better, but we see that the performance of the MF does not improve with dimension, while the performance of ACE does. This trend suggests that, because HSI data has hundreds of dimensions, the MF based detectors will not perform as well as ACE-based detectors.
Figure 2.4: Test Case 3 for ACE (solid) and MF (dashed). $P_D$ vs. fill fraction for fixed $P_{FA}$.
2.5 Test Case 4: Effect of Tail Heaviness

In this section the effect of tail heaviness on the performance of the MF and ACE is examined. We replace the MVN background with a MVT background (see Appendix B.2) and use the hypotheses:

\[
H_0 : \mathbf{x} \sim t_\nu(\mu_b, \Sigma_b) \\
H_1 : \mathbf{x} \sim \mathcal{N}(s, \Sigma_b)
\]

where \( \nu \) is the tail-heaviness parameter. Setting the dimension \( p = 5 \) and VSNR= 5, a comparison of the MF and ACE for various \( \nu \) is shown in Figure 2.5. For \( p = 5 \), the MF detector outperforms the ACE for most choices of tail parameter, but its performance depends on \( \nu \), while the ACE detector’s performance does not.

The robustness of the ACE family of detectors can be attributed to their operation; ACE is measuring an angle, so arbitrarily scaling the data (about the background mean) does not change this angle. This is simply demonstrated by noting that \( \bar{x} \) occurs in the denominator and the numerator in ACE (Equation 1.15a); therefore, we can factor any scaling of \( \bar{x} \) from the statistic; the parameter \( \nu \) acts as this type of scaling in the t-distribution. This property makes the ACE family, constant false alarm rate (CFAR) detectors with respect to tail heaviness and arbitrary scaling about the background mean. The MF and related detectors are sensitive to the parameter \( \nu \); so when the data is closer to being t-distributed than normally distributed, using a MVT model for the data is more appropriate when considering the MF.
Figure 2.5: MF (solid) and ACE (blue dashed) detectors where the background is MVT with tail parameter $\nu$.

2.6 Test Case 5: Effect of Covariance Mismatch

Typically, since we do not know and cannot estimate the covariance of the target, we cannot know the type of mismatch that will occur. However, we can consider a simple case, to see how it affects the distribution of the output. For this test case, we have the hypotheses:

$$H_0 : x \sim t_{\nu} (\mu, \Sigma_t)$$
$$H_1 : x \sim N (\mu_m, \Sigma_m)$$

where $\Sigma_m = \alpha^2 \Sigma_t + (1 - \alpha)^2 \Sigma_b$ and $\mu_m = \alpha \mu_t + (1 - \alpha) \mu_b$. This is the general mixing model that we will be using throughout this work. In this case, we can choose the target covariance matrix $\Sigma_t$. To differentiate this experiment from previous ones, we do not want the target covariance matrix to be a scalar multiple of the background covariance matrix. Instead we
can simulate mismatch by setting

\[
\Sigma_t = \begin{bmatrix} A & 0 \\ 0 & I \end{bmatrix}
\]

where \( A = \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} \frac{\sigma}{\sqrt{2}} \)

Basically, the first two eigenvectors of the covariance matrix are at 45 degrees to the axes and are scaled. We could scale or correlate these vectors individually, but we will not explore these options. The scaling parameters selected were

\[ \sigma = \{1, 2, 3\} \]

which we expect to degrade performance. We present ROC curves for this case in Figure 2.6.

Although, we see performance loss at higher fill fractions, there is a crossover point where the performance is the same for all choices of scaling parameter \( \sigma \). Indeed, at smaller fill fractions the performance when there is mismatch is actually better than without. We expect the opposite change in performance when \( \sigma < 1 \).

Because, the target covariance matrix is not known in practice, this performance loss will be observed, but since it is more pronounced at higher fill fractions we are usually unable to predict this loss. At lower fill fractions, the background covariance dominates the mixed pixel covariance, which means for lower fill fractions our prediction and observations should be closer in agreement.
Figure 2.6: Probability of detection of ACE (colored) and MF (black) for various fill fractions when there is covariance mismatch. Only the MF curves for $P_{FA} = 10^{-3}$ are included.

2.7 Summary

In this section we explored several simple hypotheses for the target and background. The effects of dimension, tail-heaviness, fill-fraction and covariance mismatch were shown to have effects on detection performance, but the relevant parameters for the MF were different from ACE. Increased tail-heaviness decreases performance of the MF, while covariance scaling and fill fraction benefited ACE more than the MF. For most choices of parameters, when the dimension of the input was small, the MF had better performance than ACE, but the opposite was true when $p$ was large. Since, HSI data may have hundreds of dimensions, we expect the ACE to perform better than the MF, in most situations.
Chapter 3

The Geometry of Detectors and False Alarm Mitigation Techniques

The detectors discussed thus far represent a distance and an angle, but why these two measures are significant to our problem has only been briefly discussed. In this section we will expand our set of detectors, and give situations where each is appropriate. We describe a geometric interpretation of each detector which involves two orthogonal subspaces.

3.1 The Signal and the Orthogonal Subspaces

Thinking about our detection algorithms in terms of reflectance signatures, restricts our ability to visualize what we are doing. Perhaps the simplest way to think about our algorithms is by representing our $p$ dimensional space as two subspaces, the signal subspace and the “orthogonal subspace.”

In the MF and ACE we first de-mean and then whiten to get the vectors $\tilde{x}$ and $\tilde{s}$. The vector $\tilde{s}$ (assumed non-zero) actually defines a 1-dimensional linear subspace in $\mathbb{R}^p$. Any vector that lies in this subspace can be written as $a\tilde{s}$ where $a$ is a scalar (the converse is true too); we will call this subspace the “signal subspace” and will write it as $< \tilde{s}>$, following
Suppose we want to represent the vector $\tilde{x}$ in terms of $\tilde{s}$; choosing $a$ such that the distance from $a\tilde{s}$ to $\tilde{x}$ is minimized gives us one such representation. The vector $a\tilde{s}$ in this case is called the linear projection of $\tilde{x}$ on the subspace $<\tilde{s}>$. Since this projection is linear, we can represent it with a matrix $P_{\tilde{s}}$, which is known as a projection matrix; we will define this below and is defined in Appendix A. This matrix is symmetric and is said to be idempotent, meaning $P_{\tilde{s}}^2 = P_{\tilde{s}}$, which follows because if a vector $\tilde{x}$ lies in the subspace $<\tilde{s}>$ then projecting onto that subspace yields the same vector, eg. $P_{\tilde{s}}\tilde{s} = \tilde{s}$. In fact, this property holds for all projection matrices while a similar property holds for non-linear projections [29].

After, finding the linear projection $P_{\tilde{s}}\tilde{x}$, we can examine the remaining vector $\tilde{x} - P_{\tilde{s}}\tilde{x}$. This vector is said to be orthogonal to $P_{\tilde{s}}\tilde{x}$ because the inner product

$$(\tilde{x} - P_{\tilde{s}}\tilde{x})^T P_{\tilde{s}}\tilde{x} = \tilde{x}^T P_{\tilde{s}}\tilde{x} - \tilde{x}^T P_{\tilde{s}} P_{\tilde{s}}\tilde{x} = 0$$

where the last manipulation is possible because $P_{\tilde{s}}$ is idempotent. The vector $\tilde{x} - P_{\tilde{s}}\tilde{x}$ is called the orthogonal projection of $\tilde{x}$ on $\tilde{s}$. By factoring $\tilde{x}$ we have $(I - P_{\tilde{s}})\tilde{x}$ and $P_{\tilde{s}}^\perp = I - P_{\tilde{s}}$ is called an orthogonal projection matrix because the component of $\tilde{x}$ that lies in the subspace orthogonal to $<\tilde{s}>$ is $P_{\tilde{s}}^\perp \tilde{x}$; we will refer to this subspace as the orthogonal subspace and will denote it by $<\tilde{s}^\perp>$. Using these projections we can write any input $\tilde{x}$ as a sum of these two projections:

$$\tilde{x} = P_{\tilde{s}}\tilde{x} + P_{\tilde{s}}^\perp \tilde{x}$$

In Figure 3.1 this “decomposition” is illustrated in two dimensions, but the way the projection matrices have been defined in Appendix A, this representation generalizes to $p$ dimensions. These projections allow us to generalize the MF and ACE, and provide a geometric view of our algorithms.
3.1.1 Signed Detectors

By projecting $\tilde{x}$ on the signal subspace, we are finding the closest representation for $\tilde{x}$ that has the form $a\tilde{s}$. It turns out that the matched filter does just this. Recall that the matched filter vector is $h = \frac{\tilde{s}}{\sqrt{\tilde{s}^T\tilde{s}}}$ and from our definition of $P_{\tilde{s}}$ in Appendix A we have

$$P_{\tilde{s}} = hh^T$$

The projection can be rewritten in terms of the MF as

$$P_{\tilde{s}}\tilde{x} = hh^T\tilde{x} = h\, y_{\text{MF}}$$

and since $h$ is in the same direction as $\tilde{s}$ we can find the coefficient $a$ with the MF. Thus the MF tells us how much of the test pixel lies in the signal subspace, and it tells us whether the input is in the same direction, or in the opposite direction as the target vector.

In our current scenario the orthogonal subspace is $p - 1$ dimensions, which means in general we cannot visualize this component as is. Instead, by using the length of the orthogonal projection, we describe the vector $P_{\tilde{s}}^\perp$ with a single number. The length of this
projection is $||P_\tilde{s}^\perp \tilde{x}||$, where $||.||$ is called the length or “2-norm” and is defined for a vector $x$ as $||x|| = \sqrt{x^T x}$. This quantity measures how much of the test vector $\tilde{x}$ lies in the orthogonal subspace. Intuitively we expect targets to be close to the signal subspace, the orthogonal projection distance is a measure of how close the pixel is to this subspace.

Since the MF ignores this orthogonal distance, we may want to penalize inputs with a large orthogonal distance. Dividing the MF by this distance yields the cotangent and scaling by $\sqrt{p-1}$ yields the t-detector of Equation 1.15c:

$$y_t = \sqrt{p-1} \frac{\tilde{s}^T \tilde{x}}{\sqrt{\tilde{s}^T \tilde{s}} \sqrt{\tilde{x}^T P_\tilde{s}^\perp \tilde{x}}}$$

$$= \sqrt{p-1} \frac{y_{MF}}{\sqrt{\tilde{x}^T P_\tilde{s}^\perp \tilde{x}}}$$

$$\cot = \frac{y_{MF}}{\sqrt{\tilde{x}^T P_\tilde{s}^\perp \tilde{x}}}$$

which are statistically equivalent to ACE. Since, the denominator in each of these detectors is always positive, the sign of the output depends on the sign of the MF. If the test vector and target vector point in the same direction, the sign is positive, and if they point in opposite directions, the sign is negative. Hence we refer to these detectors as signed detectors.

### 3.1.2 Subspace Detectors

Until now, we have only considered cases where there is a single target vector $\tilde{s}$. However, if we have multiple target vectors or signatures, we have two options: use a detector for each target or use a detector for all targets at once. If we choose the first option, then we can use the MF or ACE for each signature; if we choose the second option, we have to generalize our detectors to accommodate extra target vectors. This is easily accomplished with our notion of subspaces.

In this scenario, instead of a single signature $s$ we have a matrix $S$ of $M$ signatures

$$S = \begin{bmatrix} s_1 & s_2 & \ldots & s_M \end{bmatrix}$$
After whitening and de-meaning each signature, as before, we obtain \( \tilde{S} = [\tilde{s}_1 \ldots \tilde{s}_M] \), which is a \( p \times M \) matrix; for simplicity we assume that \( M < p \) and that these vectors form an \( M \) dimensional subspace (ie. they are linearly independent). In this situation, the projection matrices have a similar form as before

\[
P_{\tilde{S}} = \tilde{S}(\tilde{S}^T\tilde{S})^{-1}\tilde{S}
\]

\[
P_{\tilde{S}}^\perp = I - \tilde{S}(\tilde{S}^T\tilde{S})^{-1}\tilde{S}
\]

but now the projections \( P_{\tilde{S}}\tilde{x} \) and \( P_{\tilde{S}}^\perp\tilde{x} \) are \( M \) and \( p - M \) dimensional vectors. As a generalization of the MF, using the squared-length of the projection \( P_{\tilde{S}}\tilde{x} \) we obtain the matched subspace detector (MSD) of [26]:

\[
y_{\text{MSD}} = \tilde{x}^T P_{\tilde{S}}\tilde{x}
\]

When the input is \( \tilde{x} \sim \mathcal{N}(0, I) \), the output of this detector is chi-squared distributed:

\[
y_{\text{MSD}} \sim \chi^2_M
\]

where \( M \) is the degrees of freedom parameter of the distribution. As with the MF, this detector ignores the component of the input that lies in the orthogonal space. To include this component, as we did for the MF previously, we can divide the MSD by the squared orthogonal distance to obtain

\[
\cot^2 = \frac{\tilde{x}^T P_{\tilde{S}}\tilde{x}}{\tilde{x}^T P_{\tilde{S}}^\perp\tilde{x}}
\]

or equivalently the F-detector

\[
F = \frac{p - M}{M} \frac{\tilde{x}^T P_{\tilde{S}}\tilde{x}}{\tilde{x}^T P_{\tilde{S}}^\perp\tilde{x}}
\]

and the \( \beta \)-detector

\[
y_{\beta} = \frac{\tilde{x}^T P_{\tilde{S}}\tilde{x}}{\tilde{x}^T \tilde{x}}
\]

which is the standard definition of ACE found in the literature and is monotonically related to \( \cot^2 \) as \( y_{\beta} = \frac{\cot^2}{\cot^2 + 1} \). The F-detector and \( \beta \)-detector follow F and \( \beta \) distributions respectively.
when \( \mathbf{x} \sim \mathcal{N}(0, \mathbf{I}) \), we have

\[
\begin{align*}
y_{\beta} & \sim \beta_{M/2, (p-M)/2} \\
F & \sim F_{M, p-M}
\end{align*}
\]

which are the central versions of these distributions; when the target is present and the mean of \( \mathbf{x} \) lies in the signal subspace\(^1\) the MSD, F and \( \beta \) detectors follow “non-central” distributions with non-centrality parameter \( \lambda \) where \( \lambda = ||\mathbf{P}_{\mathbf{S}} \mathbf{S} \mathbf{a}||^2 \); this \( \lambda \) is just a generalization of SNR defined in 1.11b to multiple target signatures. For definitions of these distributions see [10].

Finally, we note that for the special case where the target subspace is a single vector, the MSD and \( y_{\beta} \) are simply the MF squared (Adaptive Matched Filter/AMF) and ACE squared respectively.

### 3.1.3 Detector Summary

These detectors can be put into three categories: signed, squared and subspace. The signed detectors are the MF, ACE, and cotangent; to get a sign-insensitive detector squaring the output of these detectors yields the squared detectors; finally, to detect multiple signatures at once, the subspace detectors are appropriate. This categorization is summarized in Table 3.1 while the names and abbreviations for each detector are summarized in Table 3.2. Although the nomenclature is somewhat confusing, and some of the detectors are redundant, they are listed here for completeness.

Given the large number of detectors, it may be difficult to pick which one is best for a particular application. Recall from Section 1.3 that the appropriate detector to use, depends on our assumptions about the target signal. Given one set of assumptions we obtained the MF and for a slightly different set of assumptions we obtained ACE. The assumptions about the target signal and the corresponding detector to use are summarized in Table 3.3. Basically, when the scaling \( a \) can be positive or negative, a squared detector is more appropriate,

\(^1\)That is when \( \mathbf{x} \sim \mathcal{N} \left( \mathbf{S} \mathbf{a}, \mathbf{I} \right) \) with \( \mathbf{a} \in \mathbb{R}^M \).
but when \( a \) is positive we want a signed detector. Although, it seems that the MF, AMF, and MSD are better for certain assumptions, as discussed in Chapter 2, the ACE and \( \beta \)-detector are less sensitive to data parameters than the MF, and generally do better when the dimension of the input is high.

<table>
<thead>
<tr>
<th>Name</th>
<th>Signed</th>
<th>Squared</th>
<th>Subspace s</th>
</tr>
</thead>
<tbody>
<tr>
<td>Matched Filter</td>
<td>( \frac{\tilde{x}^T \tilde{s}}{\sqrt{\tilde{s}^T \tilde{s}}} )</td>
<td>( \frac{(\tilde{x}^T \tilde{s})^2}{\tilde{s}^T \tilde{s}} )</td>
<td>( \tilde{x}^T P \tilde{s} \tilde{x} )</td>
</tr>
<tr>
<td>Cosine</td>
<td>( \frac{\tilde{x}^T \tilde{s}}{\sqrt{\tilde{s}^T \tilde{s}} \sqrt{\tilde{x}^T \tilde{x}}} )</td>
<td>( \frac{(\tilde{x}^T \tilde{s})^2}{\tilde{s}^T \tilde{s} (\tilde{x}^T \tilde{x})} )</td>
<td>( \tilde{x}^T P \tilde{s} \tilde{x} )</td>
</tr>
<tr>
<td>Cotangent</td>
<td>( \frac{\tilde{x}^T \tilde{s}}{\sqrt{\tilde{s}^T \tilde{s}} \sqrt{\tilde{x}^T P_{\perp} \tilde{x}}} )</td>
<td>( \frac{(\tilde{x}^T \tilde{s})^2}{\tilde{s}^T \tilde{s} (\tilde{x}^T P_{\perp} \tilde{x})} )</td>
<td>( \tilde{x}^T P_{\perp} \tilde{s} \tilde{x} )</td>
</tr>
</tbody>
</table>

**Table 3.1:** Summary of detectors.
<table>
<thead>
<tr>
<th>Category</th>
<th>Name</th>
<th>Shorthand</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Signed</td>
<td>Matched Filter</td>
<td>MF</td>
<td>$\frac{x^T s}{\sqrt{s^T s}}$</td>
</tr>
<tr>
<td></td>
<td>Cosine</td>
<td>ACE/cos</td>
<td>$\frac{x^T s}{\sqrt{s^T s} \sqrt{x^T x}}$</td>
</tr>
<tr>
<td></td>
<td>Cotangent</td>
<td>cot</td>
<td>$\frac{x^T s}{\sqrt{s^T s} \sqrt{x^T P_s^\perp x}}$</td>
</tr>
<tr>
<td></td>
<td>t-detector</td>
<td>t</td>
<td>$\sqrt{p-1} \frac{x^T s}{\sqrt{s^T s} \sqrt{x^T P_s^\perp x}}$</td>
</tr>
<tr>
<td>Adaptive</td>
<td>Matched Filter</td>
<td>AMF</td>
<td>$\frac{(x^T s)^2}{s^T s}$</td>
</tr>
<tr>
<td>Squared</td>
<td>Cosine Squared</td>
<td>$\beta/cos^2$</td>
<td>$\frac{(x^T s)^2}{(s^T s)(x^T x)}$</td>
</tr>
<tr>
<td></td>
<td>Cotangent Squared</td>
<td>cot$^2$</td>
<td>$\frac{(x^T s)^2}{(s^T s)(x^T P_s^\perp x)}$</td>
</tr>
<tr>
<td>Matched</td>
<td>Subspace Detector</td>
<td>MSD</td>
<td>$\frac{x^T P_s \hat{x}}{x^T \hat{x}}$</td>
</tr>
<tr>
<td>Subspace</td>
<td>CFAR MSD</td>
<td>$\beta/cos^2$/CFAR MSD</td>
<td>$\frac{x^T P_s \hat{x}}{x^T \hat{x}}$</td>
</tr>
<tr>
<td></td>
<td>F-Detector</td>
<td>F</td>
<td>$\frac{p-M}{M} \frac{x^T P_s \hat{x}}{x^T \hat{x}}$</td>
</tr>
</tbody>
</table>

**Table 3.2:** Detectors by type with names, shorthand and definitions.

<table>
<thead>
<tr>
<th>$a &gt; 0$</th>
<th>$a \neq 0$</th>
<th>Subspace $s$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Known $\sigma^2$</td>
<td>MF</td>
<td>AMF</td>
</tr>
<tr>
<td>Unknown $\sigma^2$</td>
<td>ACE/t</td>
<td>$\beta$</td>
</tr>
</tbody>
</table>

**Table 3.3:** Parameter assumptions and resulting detectors when $\mathbf{x} \sim \mathcal{N}(a \mathbf{s}, \sigma^2 \Sigma_b)$. 

56
3.2 The Geometry of False Alarm Mitigation Techniques

Throughout our development and discussion of detection algorithms and the geometry involved, we have used two quantities: the MF and the orthogonal projection distance. Using them in a ratio results in ACE or $y_\beta$, while ignoring the orthogonal component is just the MF. Because the MF ignores the orthogonal component it generally has worse performance than other detectors in practice. The MF with False Alarm Mitigation (MF-FAM) of [4] is a variation on the MF, in that a threshold is set for both the MF and the orthogonal projection $||\tilde{P}_s\tilde{x}||$.

Another way to potentially do false alarm mitigation is to threshold both the MF and the ACE statistic. This technique is called the Adaptive Sidelobe Blanker (ASB) of [15] and [24]. The original definitions are in terms of the AMF and $\beta$-detectors, but the ideas are completely analogous for the signed versions of the detectors.

To understand how these algorithms work to mitigate false alarms we will use the MF-FAM as a starting place. Plotting $||\tilde{P}_s\tilde{x}||$ with respect to $y_{MF}$, gives us a 2-D representation of each detector. As other authors have shown, the MF-FAM, MF, ASB, and ACE can all be easily visualized in this plot. For example, setting a threshold for ACE or equivalently cotangent so that $\cot = 0.5$, implies that

$$\cot = \frac{y_{MF}}{||\tilde{P}_s\tilde{x}||} = 0.5$$

$$y_{MF} = 0.5||\tilde{P}_s\tilde{x}||$$

which defines a line that passes through the origin as shown in Figure 3.2 for some synthetic data; in the figure, the green points represent a confuser material, and would be false alarms for the matched filter, but could be excluded by the MF-FAM, ACE and also the ASB. For each detector, points to the lower right on the plot will be declared targets.

A major difficulty when using any algorithm is threshold selection. While, angle-based
detectors are CFAR with respect to tail heaviness, this is not true for the matched filter or its relatives. For dual-threshold algorithms, it is even less clear how to set a proper threshold. By relating the MF output to the ACE output, or by finding the relationship between the MF and $|P_s \tilde{x}|$ we hope to make steps in this direction.

![Graph](image)

**Figure 3.2:** Matched filter and orthogonal projection distance for some synthetic data. Decision boundaries at various thresholds for the MF, MF-FAM, and cosine detectors are overlaid.

### 3.3 Relation of Fill Fraction to ACE

Until now our discussions of data modeling and our detection algorithms have been somewhat separate. For the purpose of modeling, we have the fill fraction of the pixel, while for detectors we have VSNR or SNR. However, the SNR metric can only be used in certain circumstances and does not capture the full effect of fill fraction. Therefore, our goal is to find the distribution of pixels at a particular fill fraction and then determine the expected
response for that fill fraction. In general the fill fraction for a particular pixel is unknown, but we can estimate it using the FFE or $y_{FFE}$ of Equation 1.9. Using this estimate we can find the expected response of a target at that fill fraction and compare the expectation with reality.

By choosing several fill fractions and computing the CDF of ACE for each, we create a profile of how the output statistics change with fill fraction. Since $P_D$ is determined by the CDF of the output when the target is present, this tells us how well we can detect a target at a particular fill fraction. This principle is demonstrated in Figure 3.3, where the CDF of ACE is plotted for several fill fractions. The pink lines denote CDF values of 0.1, 0.5 and 0.9; the 0.5 mark should be the median of the data at that fill fraction. Computing the CDF of ACE and the $\beta$-detector for any fill fraction is the topic of Chapter 4 and Chapter 5. The model predicts that most of the data should lie between the upper and lower curves. Generally the expected responses will differ from the observed because the true parameters are unknown, but this technique can be used to help choose a threshold for a FAM technique.

It is interesting to note that plotting ACE versus the MF (or fill fraction) contains the same information as plotting the orthogonal component versus the MF. For example, setting an ACE threshold resulted in a line through the origin in Figure 3.2, but it is obvious that such a threshold would be a horizontal line if we plot ACE versus the MF. We will examine this property further for real data in Section 6.4.
Figure 3.3: Illustration of how the CDF of ACE changes with fill fraction. The pink lines are the 10, 50 and 90 percent CDF points for each fill fraction.

3.4 Summary

The decomposition of a vector into two orthogonal components is central to a variety of detection algorithms and false alarm mitigation techniques. How these two components are used, whether in a ratio or independently determines the resulting algorithm and decision boundary for the algorithm. For HSI the matched filter component is proportional to pixel fill fraction; relating the matched filter (or fill fraction) to detector output, allows us to determine how well various detectors and false alarm mitigation techniques should perform for targets of various sizes. This idea can be applied to any of the algorithms discussed in this section and will be examined further in subsequent chapters.
Chapter 4

The Distribution of ACE with Unconstrained Covariance Mismatch

In this chapter we cover techniques for computing the CDF of ACE, the cotangent, the $\beta$-detector and the F-detector. In previous sections we presented these detectors and gave their distributions under ideal circumstances. This ideal circumstance occurs when the whitened input $\tilde{x}$ has identity covariance matrix. In general, whitening of the input data is not perfect, and so the data is non-white after whitening. In the case of the MF, the output is easily determined because the detector is linear. However, we need special consideration for ACE and the $\beta$-detector because their CDFs (and hence the $P_D$ and $P_{FA}$) do not follow standard distributions except for special cases of mismatch. Here, we present computational methods that do not constrain the type of mismatch but require that the dimension $p$ is large.
4.1 The CDF of ACE and the $\beta$-detector with Covariance Mismatch

The two detectors that we will study here are the ACE and $\beta$ (cosine-squared) detectors. These detectors are based on the angle between the whitened target vector and the whitened vector under test as shown in Figure 4.1. Recall that the detectors are

$$y_{\text{ACE}} = \frac{\tilde{x}^T \tilde{s}}{\sqrt{\tilde{x}^T \tilde{x} \sqrt{\tilde{s}^T \tilde{s}}}}$$
$$y_{\beta} = \frac{\tilde{x}^T P \tilde{s} \tilde{x}}{\tilde{x}^T \tilde{x}}$$

where the whitened vectors $\tilde{s}$ and $\tilde{x}$ are $\tilde{s} = \Sigma_b^{-1/2} s$ and $\tilde{x} = \Sigma_b^{-1/2} x$ while the whitened subspace is $\tilde{S} = \Sigma_b^{-1/2} S$ (Assuming that all vectors have been de-meaned). Again, ACE includes the matched filter:

$$y_{\text{MF}} = \frac{\tilde{x}^T \tilde{s}}{\sqrt{\tilde{s}^T \tilde{s}}}$$

while $y_{\beta}$ includes the MSD

$$y_{\text{MSD}} = \tilde{x}^T P \tilde{s} \tilde{x}$$

The geometry of these detectors and their definitions is summarized in Figure 4.1. Although not displayed in the figure, remember that the vector $x$ follows some statistical distribution, in this section, we simply assume that it is MVN:

$$x \sim \mathcal{N}(\mu, \Sigma)$$

with no constraints put on $\mu$ or $\Sigma$.

Since, $\Sigma$ may be different from $\Sigma_b$, the covariance matrix of $\tilde{x}$ may not be the identity matrix. This means that the components of $\cot(\theta)$ and $\cot^2(\theta)$ may not be independent. To solve this problem we will use conditional probability for $\cot(\theta)$ and a clever matrix manipulation for $\cot^2(\theta)$. These two statistics are easier to handle than ACE of $\beta$ since they are ratios of orthogonal components. In contrast, ACE and $y_{\beta}$ are ratios of correlated...
components. However, because of the statistical equivalence of these pairs of detectors, by computing the CDF for one of the pair, we get the CDF of the other. This is the strategy we employ herein.

\[ \text{Single Input Vector } \tilde{s} \]

\[ \text{Subspace Input } \tilde{S} \]

\[ y_{\text{ACE}} = \cos(\theta) = \frac{y_{\text{MF}}}{||\tilde{x}||} \]

\[ \cot(\theta) = \frac{y_{\text{MF}}}{||P_{\tilde{s}}\tilde{x}||} \]

\( (a) \)

\[ y_{\text{MF}} \]

\[ y_{\beta} = \cos^2(\theta) = \frac{||P_{\tilde{S}}\tilde{x}||^2}{||\tilde{x}||^2} \]

\[ \cot^2(\theta) = \frac{||P_{\tilde{S}}\tilde{x}||^2}{||P_{\tilde{S}}\tilde{x}||^2} \]

\( (b) \)

**Figure 4.1:** Illustrations and definitions of detection geometry when (a) there is single target vector and (b) there are several target vectors. \( \theta \) is the angle between the vector \( \tilde{x} \) and the subspace formed by \( \tilde{s} \) or \( \tilde{S} \).

### 4.2 Computing the CDF of ACE

In practice ACE is appealing because of its simplicity, but its CDF is not readily computable and does not follow a well known distribution; therefore, we will transform the problem into one using the orthogonal components \( \tilde{s}^T\tilde{x} \sqrt{\tilde{s}^T\tilde{s}} \) and \( ||P_{\tilde{s}}\tilde{x}|| \). From these components we obtain the t-detector which we redefine as

\[ y_t = \sqrt{p - 1} \frac{\tilde{s}^T\tilde{x}}{\sqrt{\tilde{s}^T\tilde{s}}} \frac{1}{\sqrt{\tilde{x}^T P_{\tilde{s}}^{-1} \tilde{x}}} \]

(4.1)
where $p$ is the dimension of the vectors $\mathbf{x}$ and $\mathbf{s}$. Using trigonometric relations it can be shown that the t-detector and ACE are related by the following equations:

$$y_{ACE} = \frac{y_t}{\sqrt{(n-1) + y_t^2}}$$  \hspace{1cm} (4.2)

$$y_t = \sqrt{n-1} \frac{ycos}{\sqrt{1-y_{cos}^2}}$$  \hspace{1cm} (4.3)

These relationships allows us to map the CDF of one detector to the other; specifically as a function of threshold $\eta$, the probability that ACE is less than a threshold $\eta$ is

$$\Pr\{y_{ACE} \leq \eta\} = \Pr\left\{y_t \leq \sqrt{n-1} \frac{\eta}{\sqrt{1-\eta^2}}\right\}$$  \hspace{1cm} (4.4)

where $\Pr\{\cdot\}$ is read “probability that”. This means that with the CDF for the t-detector the CDF of the ACE comes free. Our task then is to find the CDF of the t-detector.

### 4.2.1 The CDF of the t-detector

We wish to obtain the CDF of the t-detector by using conditional probability rules. Consider that if the MF statistic is known then what remains is the orthogonal component; for Gaussian distributions, conditional distributions are also Gaussian. In our current situation, assuming that the MF is known, yields the conditional distribution of the orthogonal distance, and vice versa. First, the CDF of the t-detector may be written as

$$F_y(\eta) = \Pr\{y_t \leq \eta\}$$  \hspace{1cm} (4.5)

$$= \Pr\left\{\sqrt{n-1} \frac{\tilde{s}^T \tilde{x}}{\sqrt{\tilde{s}s}} \frac{1}{\sqrt{x^T P_\perp \tilde{x}}} \leq \eta\right\}$$  \hspace{1cm} (4.6)

and to simplify notation we define $h = \tilde{x}^T P_\perp \tilde{x}$ and note that $y_{MF} = \frac{\tilde{s}^T \tilde{x}}{\sqrt{\tilde{s}s}}$. Substituting these values into (4.6) and moving $\sqrt{h}$ to the right hand side yields

$$F_y(\eta) = \Pr\left\{\sqrt{n-1} y_{MF} \leq \eta \sqrt{h}\right\}$$  \hspace{1cm} (4.7)
Moving everything but the matched filter to the right, we have

\[
F_y(\eta) = \Pr \{ y_{\text{MF}} \leq \eta' \} \quad (4.8)
\]

\[
\eta' = \frac{1}{\sqrt{n-1}} \eta \sqrt{h} \quad (4.9)
\]

At this point we have an expression in terms of two random variables \( y_{\text{MF}} \) and \( h \). However, if the value of \( h \) is known, then \( \eta' \) is known. This means that after conditioning on \( h \), Equation 4.7 is simply the CDF of the MF. Luckily we know the distribution of the MF is

\[
y_{\text{MF}} \sim \mathcal{N}(\mu_{\text{MF}}, \sigma_{\text{MF}}^2) \quad \text{where,}
\]

\[
\mu_{\text{MF}} = \hat{s}^T \Sigma_b^{-1/2} \mu
\]

\[
\sigma_{\text{MF}}^2 = \hat{s}^T \Sigma_b^{-1/2} \Sigma \Sigma_b^{-1/2} \hat{s}, \quad \text{and}
\]

\[
\hat{s} = \frac{\tilde{s}}{\sqrt{s^T \tilde{s}}}
\]

Therefore given \( h \), Equation 4.7 is simply the CDF of a normally distributed random variable, which is readily computable and well known.

Obtaining the final output still requires integrating out \( h \). Because \( h \) is always positive the CDF becomes

\[
F_y(\eta) = \int_0^\infty \Pr \{ \sqrt{n-1} y_{\text{MF}} \leq \eta \sqrt{h} \} f_h(h) \, dh \quad (4.10)
\]

where \( f_h(h) \) is the PDF of \( h \) and \( \Pr \{ . | h \} \) denotes conditioning on \( h \). Recall that \( h = \tilde{x}^T P_{\tilde{s}}^\perp \tilde{x} \) and that \( P_{\tilde{s}}^\perp \tilde{x} \) consists of \( p-1 \) components of \( \tilde{x} \). Since \( \tilde{x} \) is MVN distributed, the components \( P_{\tilde{s}}^\perp \tilde{x} \) are also MVN distributed. The squared distance of these components \( \tilde{x}^T P_{\tilde{s}}^\perp \tilde{x} \) is known as a quadratic form (see Appendix C.1). A general quadratic form is \( x^T A x \), which ends up being a sum of squared elements. For example if \( A \) is diagonal, then \( x^T A x \) is \( \sum x_i^2 A_{ii} \), where \( x_i \) is the \( i \)th element of \( x \) and \( A_{ii} \) is the \( i \)th diagonal element of \( A \). Since, the components of the orthogonal projection are MVN distributed, the quadratic form is in terms of normal random variables. Finally, using the results in section C.2.1 we can compute \( f_h(h) \) and numerically evaluate the integral of Equation 4.10.

65
4.3 Computing the CDF of the $\beta$-detector

We would like a simple way to compute the CDF of the $y_\beta$, but again, because its components are correlated, we will reduce it to an easier form before proceeding. Recall that the F-detector is a scalar multiple of $\alpha$ and is statistically equivalent to $\cot^2$. The F-detector for subspace input is

$$y_F = \frac{p - M}{M} \frac{\tilde{x}^T P \tilde{x}}{\tilde{x}^T P_s \tilde{x}}$$  \hspace{1cm} (4.11)

where $M$ is the number of vectors in $S$ and $p$ the dimension of $x$. The $\beta$-detector and the F-detector are related by

$$y_\beta = \frac{y_F}{(p - M)/M + y_F}$$  \hspace{1cm} (4.12)

$$y_F = \frac{p - M}{M} \frac{y_\beta}{1 - y_\beta}$$  \hspace{1cm} (4.13)

As before, we will use (4.13) to map probabilities from one detection statistic to the other. It can be shown that the CDFs of the two detectors are related by

$$\Pr\{y_\beta \leq \eta\} = \Pr\left\{y_F \leq \frac{p - M}{M} \frac{\eta}{1 - \eta}\right\}$$  \hspace{1cm} (4.14)

All we need to do now is find the CDF of the F-detector and we get the CDF of the $\beta$-detector for free.

4.3.1 The CDF of the F-detector

We have introduced the F-detector because it is more commonly used than the cotangent, but now we consider only the right hand side of the detector and define $y$ as

$$y = \cot^2 = \frac{\tilde{x}^T P_s \tilde{x}}{\tilde{x}^T P_s \tilde{x}}$$  \hspace{1cm} (4.15)

which is a ratio of quadratic forms in normal variables (see Appendix C.1). To proceed we wish to get our current CDF in a form like Equation C.5. To do so we define two matrices
B and D as

\[
B = \Sigma_{\text{b}}^{1/2} \Sigma_{\text{b}}^{-1/2} P_{\text{s}} \Sigma_{\text{b}}^{-1/2} \Sigma_{\text{b}}^{1/2} \tag{4.16}
\]

\[
D = \Sigma_{\text{b}}^{1/2} \Sigma_{\text{b}}^{-1/2} P_{\text{s}}^\perp \Sigma_{\text{b}}^{-1/2} \Sigma_{\text{b}}^{1/2} \tag{4.17}
\]

These matrices allow us to rewrite the ratio in terms of a new vector \( \tilde{z} \) that is \( \tilde{z} \sim \mathcal{N}(\mu, I) \), where \( \mu = \Sigma^{-1/2} \mu \). Since, \( \tilde{z} \) has covariance matrix I, its components are independent, and sums of its components are sums of independent random variables. In terms of the matrices B and D and the vector \( \tilde{z} \), a statistically equivalent way to write \( y \) is

\[
y \overset{d}{=} \frac{\tilde{z}^T B \tilde{z}}{\tilde{z}^T D \tilde{z}}
\]

where \( \overset{d}{=} \) means “is statistically equivalent to”. We can show statistical equivalence by substituting in Equation 4.15. Following the techniques in [3] we can rewrite the CDF of \( y \) as

\[
F_y(\eta) = \Pr \left\{ \frac{\tilde{z}^T B \tilde{z}}{\tilde{z}^T D \tilde{z}} \leq \eta \right\} = \Pr \left\{ \tilde{z}^T (B - \eta D) \tilde{z} \leq 0 \right\} \tag{4.18}
\]

It can be shown that \( B - \eta D \) is symmetric and has a decomposition \( U \Lambda(\eta) U^T \), with U unitary and \( \Lambda(\eta) \) diagonal. Defining another random vector \( z \) as \( \tilde{z} = U^T z \) we can rewrite the quadratic form in Equation 4.18 as \( z^T \Lambda(\eta) z \). The CDF that we seek then becomes

\[
F_y(\eta) = \Pr \left\{ z^T \Lambda(\eta) z \leq 0 \right\}
\]

and \( \tilde{z} \sim \mathcal{N}(d, I) \) and \( d = U^T \mu \) which is equivalent to the quadratic form in Equation C.5. This means that we can apply the results of Section C.2.2 to get the desired CDF \( F_y(\eta) \) and use the transformation in (4.14) to get the CDF of the \( \beta \) detector as desired.

### 4.4 Handling MVT Distributed Input

Thus far in this chapter we have assumed that the input was MVN distributed, but in our model we may have MVT distributed input. Note that ACE and the \( \beta \)-detector are generally unaffected by tail-heaviness, but we include the model here for completeness. As in Equation
B.1, an MVT distribution can be defined in terms of MVN vector $\mathbf{z}$ and a chi-squared variable $s$ as

$$
\mathbf{x} \sim \mathcal{N}(\mu, \nu \Sigma)
$$

where $s \sim \chi^2_{\nu}$, $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \Sigma)$, $\Sigma$ is the covariance matrix of $\mathbf{z}$ and $\mu$ is a constant vector. By assuming that the value of $s$ is known, the vector $\mathbf{x}$ becomes Gaussian distributed. This means that conditioned on $s$ we can apply all previous results that applied only to MVN input. The difficulty is that to find the total response we have to integrated out $s$. Conditioned on $s$ we have

$$
\mathbf{x} \sim \mathcal{N}(\mu + \mathbf{z}(s/\nu)^{-1/2})
$$

which we can use to find the CDF of any given detector, which we denote $F_y(\eta|s)$. Using conditional probability rules, the total CDF is

$$
F_y(\eta) = \int_0^\infty F_y(\eta|s)f_s(s)ds
$$

where $f_s(s)$ is the PDF of $s$ and is a chi-squared PDF. In general this integral can be numerically integrated by selecting a range of values of $s$ where the integrand is non-negligible. We note that the ratio $\frac{s}{\nu}$ has mean 1 and variance $2/\nu$ and for large $\nu$ is approximately Gaussian. Again, we can use this technique for ACE and other angle-based detectors, but these detectors are insensitive to changes in $\nu$, so in practice the difference between the MVN and MVT models is small. This is not the case for the MF and MSD which are strongly affected by tail heaviness. Since the MF is linear MVT distributed input yields univariate t-distributed output.

### 4.5 Algorithm Verification

One method to verify that these methods produce the correct values is by Monte Carlo simulation. The idea is simple, generate samples from a specific distribution and then put
them through our detection system. At the output, estimate the probability of exceedance, of Section 1.2.3, of the pixels and compare that estimate with the theoretical probability of exceedance. For demonstration we use the input distribution

\[ x \sim \mathcal{N}(s, \Sigma_b) \]

which, according to Equation 1.16b, makes the distribution of the t-detector:

\[ y_t \sim t_{p-1}(\text{VSNR}, 1) \]

and we can select \( s \) to yield a specific VSNR. Samples can be generated from the desired distribution in MATLAB using \texttt{mvnrand}. Those samples are then put through the t-detector to obtain an estimate of the probability of exceedance. The theoretical probability of exceedance is \( 1 - F(y_t) \), where \( F(y_t) \) is the CDF of a t-distribution. Using the approach described in this chapter, we can compute this CDF with numerical integration. In MATLAB there is also a built-in function to compute the CDF of the t-distribution to high numerical precision. In Figure 4.2 the probability of exceedance with the Monte Carlo simulation estimate, MATLAB built-in function, and numerical integration technique are presented. We see that there is good agreement among the techniques to a tolerance of \( 10^{-8} \). For most detection scenarios this is acceptable for determining \( P_D \) and \( P_{FA} \).

69
Figure 4.2: Univariate t-distribution CDF using our technique (blue), Monte Carlo simulation (red) and MATLAB’s built-in function (green) with various non-centrality parameters and dimension $p = 20$.

4.6 Summary

In this chapter we developed two computational methods: one for computing the CDF of ACE and one for computing the CDF of the $\beta$ detector. These two methods work well when the dimension $p$ is large, which is usually the case for HSI. Unlike other methods and standard results like the F-distribution and t-distribution, these techniques can be applied when there is strong covariance mismatch, and without performing Monte Carlo simulations. These methods can be used for any of the hypothesis tests discussed so far, and will be used for the multimodal model too.
Chapter 5

Performance of the MF and the ACE in Multimodal Clutter

Thus far we have assumed that hyperpsectral backgrounds are well described by a single mean vector and covariance matrix. Yet, hyperspectral images typically have highly variable backgrounds consisting of multiple modes. As an example, we can expect that if a scene only has grass then it will be well modeled by a single Gaussian distribution, but if the same scene had a road going through it, two distributions would be more appropriate. The goal of this section is to develop a method to estimate detection performance for any background composition, target, and fill fraction. To reach this goal, we use a Gaussian Mixture Model (GMM) or t-Mixture Model for the background components, and the linear mixture model as discussed in Section 1.2.1 for full-pixel and subpixel targets. The response of the background determines the $P_{FA}$ for every threshold, while the response of target-background mixtures determine the $P_D$ for every threshold. These two quantities characterize performance for any fill fraction of interest.
5.1 Outline of the Prediction Model

The prediction model is basically the combination of a background model and a target or mixed pixel model. Theses statistical models are passed through our detection algorithms to obtain an output distribution for both the background and the assumed target. Thus, after modeling, the primary concern is determining the CDF or distribution of the output of the detector. We discuss the model in the following order

1. background modeling;
2. background covariance matrix;
3. matched filter responses; and
4. ACE responses.

5.2 Background Mixture Model and Detector Response

A generic mixture model is formed as a weighted sum of densities. Using the notation of McLachlan [21] the mixture parameters and total density are

\[ \psi_{MVN} = \{\pi_i, \mu_i, \Sigma_i\}_{i=1}^N \] (5.1)

\[ \psi_{MVT} = \{\pi_i, \mu_i, \Sigma_i, \nu_i\}_{i=1}^N \] (5.2)

\[ f(x) = \sum_{i=1}^N \pi_i f_i(x; \psi) \] (5.3)

where \( \psi \) is either \( \psi_{MVN} \) or \( \psi_{MVT} \) and the densities \( f_i(.) \) are MVN or MVT. A simple example of this model in 2D is illustrated in Figure 5.1. Applying the statistical results of Sections 1.3.2 and 1.3.4 to this mixture model may seem daunting, but since the individual components are independent, they pass through the detector independently, are weighted and summed.

Suppose that the MF response (PDF) for the \( i \)th component is \( f_{MF}(\eta; i) \), with respect to a threshold \( \eta \). Since each background component is independent, the total response of the
Figure 5.1: Illustration of the background mixture model

background is

\[ f_{MF}(\eta) = \sum_{i=1}^{N} \pi_i f_{MF}(\eta; i) \]

So, to find the total response we just need to find the response of each component as we did in Chapter 2.

To find the response of the target, we use the LMM of Section 1.2.1 where mixing occurs with only one component of the background. Recall that we describe a mixed pixel as

\[ x_m \sim \mathcal{N}(\mu_m, \Sigma_m) \] with

\[ \mu_m(\alpha) = \alpha \mu_s + (1 - \alpha) \mu_i \]

\[ \Sigma_m(\alpha) = \alpha^2 \Sigma_t + (1 - \alpha)^2 \Sigma_i \]

where \( \mu_i \) and \( \Sigma_i \) are the parameters of the \( i \)th background component. For a fixed fill fraction \( \alpha \) we can use these parameters to find the detector response and thus the performance of the detector. The incorporation of the LMM with a simple target is illustrated in Figure 5.2.
5.2.1 Background Covariance Matrix $\Sigma_b$

Given the background mixture of B, assuming Gaussian components, the parameters of the individual clusters and the overall parameters will differ. It can be shown that the “global” background mean $\mu_b$ and covariance $\Sigma_b$ are

$$\mu_b = \sum_{i=1}^{N} \pi_i \mu_i \quad (5.4)$$

$$R_i = \Sigma_i + \mu_i \mu_i^T \quad (5.5)$$

$$R_b = \sum_{i=1}^{N} \pi_i R_i \quad (5.6)$$

$$\Sigma_b = R_b - \mu_b \mu_b^T \quad (5.7)$$

where $\Sigma_i$ are the covariances of the components, and $R$ denotes a correlation matrix. In our algorithms, the global mean is used to de-mean the data and the signature, while the global covariance matrix is used for whitening (appearing as $\Sigma_b^{-1}$ in each detector). Since the covariance and mean of the detector will differ from the mean and covariance of the individual classes, whitening will not yield “white” components. For the distributions of the Cosine and $\beta$ detectors this means that the techniques of Chapter 4 will be needed, while the output CDF of the matched filter remains trivial.

5.2.2 Matched Filter Response

Since, the Matched Filter is a linear operator, Gaussian distributed input has Gaussian distributed output, and MVT distributed input has univariate t-distributed output. This is summarized here:

$$f_{MF}(\eta; i) = \mathcal{N}(\eta; \mu_i, \sigma_i^2) \text{ or } t_\nu(\eta - \mu_i; 0, \sigma_i^2) \quad (5.8)$$
Figure 5.2: Illustration of background mixture model and target at fill fractions \( \alpha = 0.5, 0.75, \) and 1.0.

where

\[
\mu_i = h^T \tilde{\mu}_i, \quad (5.9)
\]

\[
\sigma^2_i = h^T \tilde{\Sigma}_i h, \quad (5.10)
\]

and

\[
\tilde{\mu}_i = \Sigma_b^{-1/2} (\mu_i - \mu_b), \quad (5.11)
\]

\[
\tilde{\Sigma}_i = \Sigma_b^{-1/2} \Sigma_i \Sigma_b^{-1/2} \quad (5.12)
\]

The components \( \tilde{\mu}_i \) and \( \tilde{\Sigma}_i \) are the whitened (and de-meaned) versions of the background components.

Similarly, for a target-background mixture, we have whitened components

\[
\tilde{\mu}_m(\alpha) = \Sigma_b^{-1/2} (\mu_m - \mu_b), \quad \text{and}
\]

\[
\tilde{\Sigma}_m(\alpha) = \Sigma_b^{-1/2} \Sigma_m \Sigma_b^{-1/2}
\]

which are used in Equation 5.8 to get the mixed pixel response. A simple example in 2D is illustrated in Figure 5.3. The MF responses are simply projections of the distributions onto the vector connecting the background mean and target vector.
5.2.3 ACE Response

Although, each component of the mixture passes through the detector independently, the response of ACE to each component of the mixture is non-trivial. Recall from Section 1.3.2 that if $\Sigma_b = \Sigma_i$ and the input is MVN distributed then the output of the t-detector is t-distributed. However, when $\Sigma_b \neq \sigma^2 \Sigma_i$, for some $\sigma > 0$, we use the technique in Section 4.1 to find the CDF of the output for each component. Once, we have the CDF for each component $F(\eta; \psi, i)$, we weight and sum as before to find the total response:

$$F_{ACE}(\eta) = \sum_{i=1}^{N} \pi_i F_{ACE}(\eta; \psi, i)$$

where $\psi$ is either the set of parameters for a Gaussian mixture ($\psi_{MVN}$) or a mixture of t-distributions ($\psi_{MVT}$).

We state here the method for finding the distribution of the cotangent $y_{\text{cot}}$. Recall in
equation 1.15b the cotangent is defined as

\[
y_{\cot} = \frac{s^T P_s \bar{x}}{\sqrt{s^T s} \sqrt{\bar{x}^T P_s \bar{x}}}
\]

Noting that \( s^T P_s \bar{x} / \sqrt{s^T s} \) is actually the matched filter vector \( h \) of Equation 1.8, we can write the cotangent as

\[
y_{\cot} = \frac{y_{MF}}{\sqrt{\bar{x}^T P_s \bar{x}}}
\]

This is a ratio of two possibly dependent random variables, but the numerator and denominator are orthogonal and the input is elliptically contoured. The marginal and conditional distributions are also elliptically contoured, and when the input is Gaussian, both conditional and marginal distributions are Gaussian [22].

In our case, the distributions of \( y_{MF} \) and \( P_s \bar{x} \) are marginal distributions of the input, and since they are orthogonal, the distribution of one conditioned on the other is simply a conditional distribution of the Gaussian input. These properties allow us to condition on the denominator, find the parameters of \( y_{MF} \) and integrate to find the distribution of the output. This procedure is described in detail in Chapter 4.

5.3 Summary

In this section we introduced a multimodal background model, and described how it can be used in the same fashion as the unimodal models described earlier. The major reason for using a multimodal model is it is more flexible; it allows the proportions of each background component to vary and therefore novel combinations of background components can be constructed. Suppose we have a grass background in one scene and a desert background in a different scene; these could be combined to statistically describe a scene with some of each component; we expect that detection performance for certain materials will change significantly depending on the types of backgrounds present.
Chapter 6

Results

Before we can explore some of the applications of this model we have to test whether, and to what degree, the model agrees with reality. First, in Section 6.1.1 we discuss the hyperspectral data and the background components that we used in our experiments. In Section 6.2, we examine a target of known fill fraction and study the effects of target covariance and target mean on our model. In Section 6.3 we consider an application of the prediction model where we determine the minimum fill fraction needed to have acceptable detection performance. Finally, in section 6.4 we use the model to compare expected ACE output at every fill fraction with the true output at the estimated fill fraction as outlined in Section 3.3. In each case, we will compare the matched filter and ACE to show the superiority of the ACE detector.

6.1 Description of Data

The hyperspectral image used in this thesis had 255 spectral bands, which was reduced to 159 after removal of low transmittance bands (ie. water vapor absorption bands). The background clusters used in further experiments are shown in Figure 6.1. In this image there are several different targets of various sizes, including one designed with 25% fill fraction.
We will use two target materials “Material A” and “Material B” in our experiments. The target materials in the scene including Material A and Material B have similar signatures.

![Cluster map of the scene with target areas and anomalies masked.](image)

**Figure 6.1:** Cluster map of the scene with target areas and anomalies masked.

### 6.1.1 Background Modeling

In creating a background model it is assumed that the ground is covered by several distinct groups or classes of materials, where the pixels within a class are spectrally similar. Natural background ground covers (grass, fields, etc.) are well modeled by a Gaussian or t-mixture model [19]; in contrast, man-made materials that occupy only a few pixels are considered anomalies instead of background covers.

To create the background model, as described in Section 5.2, a Gaussian mixture model (GMM) is fit to the data, after masking all targets and anomalies, using the Stochastic Expectation Maximization (SEM) algorithm [20]. SEM was run several times with four Gaussian components and the parameter set (mean vectors and covariance matrices) which maximized the likelihood of the data was taken as the parameter set for the data. Pixels were assigned to the cluster that maximized its likelihood. Therefore, pixels that are a mixture of two background classes are put into the one that is statistically closer, but relatively few
pixels contain two background components as these pixels occur on the boundaries of two backgrounds. The resulting class map is shown in Figure 6.1 while the clusters in principal component space are shown in Figure 6.2.

![Figure 6.2: Scatter plot of background clusters in terms of their first two principal components.](image)

Using the sample covariance and sample mean of each cluster produces a complete Gaussian mixture model for the background. To create a t-mixture model the tail-heaviness of each cluster was assessed to find the tail parameter $\nu_i$. This is accomplished via the technique outlined in [19], which fits a weighted sum of F-distributions to the Mahalanobis distance metric of each cluster. The Mahalanobis distance measures the statistical distance of a pixel from the cluster mean, defined as

$$y_{MD} = (x - \mu_i)^T \Sigma_i^{-1} (x - \mu_i)$$

where $\mu_i$ and $\Sigma_i$ are the mean vector and covariance matrix of the $i$th cluster. When $x$ is assumed to follow a t-distribution, as outlined in Section 1.2.2 and Appendix B, the Mahalanobis distance follows an F-distribution. Therefore, when using a statistical mixture
of t-distributions, the CDF of the Mahalanobis distance takes the form

$$F_{MD}(y) = wF_{p,\nu_b} \left( \frac{y}{\nu_b} \right) + (1 - w)F_{p,\nu_t} \left( \frac{y}{\nu_t} \right)$$

where the degrees of freedom $\nu_b$ and $\nu_t$ correspond to the body and the tails of the Mahalanobis distance respectively. The weighting $w$ determines the relative contribution of the body and tail distributions. Small $\nu$ means heavy tails with many outliers, while large $\nu$ means very few outliers, and we assume $\nu > 100$ means the distribution is Gaussian.

After computing the Mahalanobis distance of each pixel in a cluster, the CDF is estimated and fitting is done for each background cluster, with the results shown in Figure 6.3b. On visual inspection it seems that some of the fits are very poor in the tails; this is because there are very few points in that region. We see that in some cases the tails seem shallower than the empirical and in some they are heavier. Since the background determines the probability of false alarm of a detector, for a “worst case” scenario we may simple use a MVT distribution with this tail parameter instead of the mixture.

### 6.1.2 Background Response

The background response for each component should be different except under ideal circumstances, as in Chapter 2. We considered two materials, Material A and Material B, which have distinct signatures. Using the background clusters for the scene, an empirical response and an predicted response were computed for each background and both the matched filter and ACE. These results for the matched filter are shown in Figure 6.4a while the results for the ACE are shown in Figure 6.4b. Although the individual component responses (colors) may fit well in the tails, the empirical and modeled aggregate responses (black) are reasonably close.
Figure 6.3: Illustration of F-distribution fitting: (a) potential F-distributions used for fitting and (b) F-distributions fitted to the Mahalanobis distance ($y_{MD}$) of each background cluster; each F-component follows an $F_{159,\nu}$ distribution.
Figure 6.4: Response of the Matched Filter (a) and ACE (b) for each background component (colors) and the aggregate response (bold black). Empirical responses (solid) and model responses (dashed) are shown for each.
6.2 Results for Target with Known Fill Fraction

When the fill fraction of a target is assumed known, we may compare the observed response with the response predicted by the model, for that fill fraction. First we studied how the choice of target mean and target covariance matrix affects the predictions. In Experiment 1 we used a library signature as $s_t$ and the background covariance matrix $\Sigma_b$ as the target covariance matrix. In Experiment 2 we used two pixels from the target mask to estimate the target signature and a set of pixels to estimate the target covariance $\hat{\Sigma}_t$. The parameters used in each experiment are shown in Table 6.1.

Although the covariance of the target typically cannot be estimated because the number of target pixels is usually far less than the number needed for an estimate. In our case, there were enough pixels of one target material to get an estimate. This estimate was used for $\Sigma_t$, and is denoted “In-Scene” in Table 6.1.

The estimated target mean was determined in a similar manner. The average spectrum of several full-pixel targets was used as the estimated target mean. Since there may be a systematic error in doing atmospheric correction there may be a significant difference between the in-scene signature and the library spectrum. In this case the spectra from only two pixels was used. We denote the use of this target mean “In-Scene” in Table 6.1. This mean will not be used as the target signature, only as the target mean; so the LMM will change but the detector we design will not.

<table>
<thead>
<tr>
<th>Experiment #</th>
<th>Target Covariance ($\Sigma_t$)</th>
<th>Target Mean ($s_t$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$\Sigma_b$</td>
<td>Library</td>
</tr>
<tr>
<td>2</td>
<td>In-Scene ($\hat{\Sigma}_t$)</td>
<td>In-Scene ($\hat{s}_t$)</td>
</tr>
</tbody>
</table>

$\Sigma_b$ is the background covariance matrix

**Table 6.1:** Test cases for mixture model simulation.
6.2.1 MF and ACE Prediction Results

We found that for these parameter sets, the predicted response was generally higher than the empirical responses for both detectors at the assumed 25% fill fraction. However, for the majority of pixels a fill fraction between 15% and 20% produced a prediction that was in the same range as the empirical detection output. Perhaps the most important result is that using the in-scene signature as the target mean shifts the distribution of scores towards the left (lower scores), but the choice of target covariance has a smaller effect on the output.

In all cases, the minimum detector output observed is greater than the empirical background response in Figure 6.5 which implies perfect detection, even for a modest fill fraction of 25%. This leads us to wonder what the smallest object we can detect at a reasonable false alarm rate is. This will be explored further in Section 6.3.

The predicted responses for the ACE exhibit a similar pattern as the matched filter: a 25% fill fraction over-estimates the detector response, fill fractions from 15% to 20% fit the observed data well, and using the in-scene target mean shifts the distributions to the left. In addition, using the estimated target covariance matrix also shifts the distributions to the left; see Figure 6.6 and compare c and d). This is because ACE depends heavily on the distribution of the input in the orthogonal subspace, while the matched filter depends on variability in the signal subspace.
Figure 6.5: MF response of 25% fill fraction target (red) and predicted responses (other colors) for parameters of (a) Experiment 1 and (b) Experiment 2 of Table 6.1.
Figure 6.6: ACE response of 25% fill fraction target (red) and predicted responses (other colors) for parameters of (a) Experiment 1 and (b) Experiment 2 of Table 6.1.
6.3 Application: Determining Detection Limits

Suppose we have a material that we want to detect and we want to know how small the object can be and still be reliably detectable. In this scenario we will have some false alarm rate that is tolerable, and some probability of detection that is tolerable. In this example we will use the values \( P_D = 0.7 \) and \( P_{FA} = 10^{-5} \). So, we plan to set a threshold such that we get 1 false alarm in 100,000 pixels, and we want to detect at least 70% of the targets.

To find the minimum detectable fill fraction only the information that is typically available was used: the library target signature \( s_t \), and a background model. Again, the target covariance is generally unknown, so it was assumed to be \( \Sigma_b \). In the experiment, Material B was embedded in the grass background, and based on previous results, we selected fill fractions of

\[
1, 5, 7.5, 10, 15 \text{ and } 20\%
\]

The resulting ROC curves for the MF are shown in Figure 6.7a while the ROC curves for ACE are shown in Figure 6.7b. If the ROC curve at a particular fill fraction passes through the “acceptable performance region” then the target at that fill fraction is considered acceptable. From these figures the acceptable minimum acceptable fill fraction for ACE is around 7.5% while it is around 15% for the MF. For ACE the ROC curves for the larger fill fractions are not shown but are well within the acceptable region. Increasing the target size benefited ACE more than the MF.

We have demonstrated that using in-scene estimates of parameters leads to lower predicted performance. In this experiment the parameters used were not estimated from the scene, so these results are rather optimistic. In practice, we can expect to see lower performance overall. Therefore, we should look at these results as an upper bound on detection performance.
Figure 6.7: Predicted ROC curves for the (a) MF and (b) ACE for several fill fractions of Material B. The acceptable performance region has a $P_{FA}$ of less than $10^{-5}$ and a $P_D$ greater than 0.7. The smallest fill fraction that satisfies these requirements for the matched filter is approximately 15%, and is approximately 7.5% for ACE.
6.4 Results for False Alarm Mitigation Techniques

The decision surface of a detector ultimately determines the weaknesses of a detector. A detector produces a score for each input vector, and this score determines whether the vector is considered a target or a non-target. Vectors with the same scores are equal for a given detector and are determined by the decision surface of the detector.

Since, the matched filter projects each pixel onto a vector (as discussed in Section 3.1.1), any pixel that shares this projection will have the same matched filter score. The decision surface for the matched filter is therefore a hyperplane; in the MF-FAM plots of Section 3.2 the decision surface is a vertical line. For ACE, pixels that form the same angle with the target vector have the same score; this results in decision surfaces that are hypercones; in the MF-FAM plots, this decision surface appears as a line that passes through the origin. One feature of this hypercone is that it gets larger as we move away from the origin. In some cases this means that pixels with high ACE values for a particular signature, actually come from a different target. This is why in some cases, using ACE alone may be problematic, the MF-FAM alleviates of Section 3.2, does not share this property and may be more selective than ACE.

In Figure 6.8a we see that similar target materials may become false alarms for the matched filter. Observe that the MF values for some groups of pixels (purple and green) are the same as the values for Material B (red), because these groups come from targets that are similar to Material B. In using both the orthogonal projection distance and the matched filter, these groups are separable. The MF-FAM thresholds, shown in the figure, can remove pixels that would otherwise be false alarms for the MF alone.

Setting a threshold for both ACE and the MF in Figure 6.8b we would be able to separate most of the confuser material pixels; this corresponds to an ASB dual threshold as mentioned in Section 3.2; however, setting the ACE threshold less than 0.6 will include most of the confuser material pixels as false alarms. The same can be said for pixels with low fill.
fractions or matched filter scores. It seems that at low fill fractions, the MF-FAM may be a better false alarm mitigation technique than the ASB.

To see how the model predictions for every fill fraction compared to the observed output, the prediction model was used to find the CDF of ACE for several fill fractions of Material B embedded in the grass background; the 10%, 50% and 90% CDF points were used to create constant CDF vs. fill fraction curves as shown in Figure 6.9a; the fill fraction of each point in the scene for Material B was estimated using the FFE, and the ACE output plotted against the FFE. The result when the target covariance was set equal to the background covariance matrix is shown in Figure 6.9b, while the results when using the estimate target covariance are shown in Figures 6.10a and 6.10b. In all cases we expect the true target pixels to lie around the 50% line (green), when the estimated covariance matrix was used there was better agreement than when background covariance matrix \( \Sigma_b \) was used as the target covariance matrix \( \Sigma_t \). Essentially, using \( \Sigma_b \) as the target covariance matrix was overly optimistic. When the estimate of the target covariance matrix \( \hat{\Sigma}_t \) was used a wider range of ACE values was predicted at high fill fractions, which agreed better with observed ACE scores. Unfortunately, in practice \( \Sigma_t \) will not be known and cannot be easily estimated, so the background covariance matrix may have to be used instead; Generally this will over-estimate detection performance at higher fill fractions, but serves as an upper bound.
Figure 6.8: (a) Orthogonal projection distance vs. MF with MF-FAM thresholds (dashed); (b) ACE vs. FFE and same MF-FAM thresholds. The true target pixels (red) in (a) have small orthogonal projections and from (b) we see that the MF-FAM would be able to remove a large number of confuser material pixels (purple, green). In contrast, an ASB threshold, (using both the MF and ACE) may still have many false alarms.
Figure 6.9: ACE vs. FFE for Material B. (a) MF-FAM threshold and predicted constant CDF curves using the background covariance matrix as the target covariance matrix ($\Sigma_t = \Sigma_b$). (b) True targets (red) and confusers (green, purple). True targets should lie between the blue and green curves.
Figure 6.10: ACE vs. FFE for Material B. (a) MF-FAM threshold and predicted constant CDF curves using the estimated target covariance ($\Sigma_t = \hat{\Sigma}_t$). (b) True targets (red) and confusers (green, purple). True targets should lie between the blue and green curves.
Chapter 7

Conclusions

In this thesis we have developed a system for predicting performance of detection algorithms in hyperspectral images. We have used a statistical model for the background, target and mixed pixels. These models are used as input to our detectors to determine performance for various parameter sets.

We have seen that for the 25\% fill fraction target the predicted output and observed output do not agree for a variety of parameter choices. Instead, using 25\% for the model produces a very optimistic result. Instead, a fill fraction of around 18\% fit the empirical results well. Why do these fill fractions differ by so much? It is possible that there is some discrepancy between the true target mean and the library signature. Or, it is possible that the true fill fraction ended up lower than the designed fill fraction. This is an area that will be explored more thoroughly in future work.

Because the predicted performance was usually better than the empirical performance, the prediction model may be best used as a “best-case” performance estimate. This was explored in Section 6.3, where the smallest acceptable fill fraction for the matched filter and ACE were found to be about 15\% and 7.5\% respectively, for the scenario that was posed. Again, this should be viewed as an upper bound, but is illustrative of how the prediction model can be used in practice.
Throughout this thesis we have only considered the performance of the matched filter and ACE; for scenarios where the squared or subspace versions of these detectors are required, the same models may be applicable but may require additional computational methods from Section 4. In the future we will see how these methods can be applied to other types of data.

Throughout this thesis, we showed that ACE outperformed the MF for a wide range of parameter sets. However, for only certain parameter sets was the agreement between the observed responses and the predicted responses qualitatively good; in particular, when target parameters were estimated from the data, we saw good agreement. However, in general these parameters will be unknown, so we should be careful how the results of the prediction model are interpreted.
Appendix A

Projection Matrices

A subspace in \( \mathbb{R}^p \) can be defined in terms of a set of vectors in \( \mathbb{R}^p \), a linear projection is a representation of a vector in the full space that lives in the subspace. Projection matrices allow us to easily express linear projections of a vector on a subspace, and to easily find the projection of a vector on a linear subspace.

A subspace \(< S >\) can be defined by a matrix \( S = [s_1, s_2, \ldots, s_m] \), compose of vector \( s_i \) (assume \( S \) has rank \( m \)). Any vector which is orthogonal to this subspace lies in the subspace \(< S^\perp >\). We state without proof that any vector in the full space \( \mathbb{R}^p \) can be written as a sum of a vector in \(< S >\) and a vector in \(< S^\perp >\). A linear projection matrix allows us to find these components.

It can be shown that the linear projection of a vector \( x \) on the subspace \(< S^\perp >\) is

\[ x_o = S(S^T S)^{-1} S^T x. \]

The matrix formed by the components on the left is known as a projection matrix and is defined as

\[ P_S = S(S^T S)^{-1} S^T \]

Now, given that the linear projection of \( x \) on that subspace is \( P_S x \), the orthogonal component is the difference, \( x - P_S x \); factoring \( x \) we get the orthogonal projection matrix

\[ P_S^\perp = I - P_S \]
and we say that the orthogonal projection of $\mathbf{x}$ on $< \mathbf{S} >$ (or the projection of $\mathbf{x}$ on $< \mathbf{S}^\perp >$) is $\mathbf{P}_S \mathbf{x}$. For a detailed discussion of these types of matrices and their statistical properties, see [25].
Appendix B

The Normal Distribution and t-Distribution

For data modeling and detector performance evaluation, two multivariate distributions are of interest, the Multivariate Normal (MVN) and the Multivariate t (MVT). Both are in the family of elliptically contoured distributions (ECD), but the MVT distribution has heavier tails [22]. Generally, HSI is fit better by MVT distributions than MVN distributions so we include both in our discussion of detectors.

B.1 Gaussian or Multivariate Normal (MVN) Distributions

The MVN distribution is probably the most common and well understood multivariate distribution. A vector \( \mathbf{x} \) that is Gaussian distributed with mean vector \( \mathbf{\mu} \) and covariance matrix \( \Sigma \), is denoted

\[
\mathbf{x} \sim \mathcal{N}(\mathbf{\mu}, \Sigma)
\]
and has the PDF [11]

\[
f(x) = \frac{1}{(2\pi)^{p/2}\vert\Sigma\vert^{1/2}} \exp \left( -\frac{1}{2}(x - \mu)^T \Sigma^{-1}(x - \mu) \right)
\]

The mean vector \( \mu \) determines the center of the distribution, while the covariance matrix \( \Sigma \) determines the shape of the distribution. It is interesting to note that the leading term is a constant, and that the exponent is a function of \((x - \mu)^T \Sigma^{-1}(x - \mu)\), which is known as the Mahalanobis distance. Vectors with large Mahalanobis distances are unlikely, while ones with small distances are more likely.

### B.2 Multivariate T (MVT) Distributions

The multivariate t-distribution is a variation on that Gaussian model, that allows for heavier tails. This distribution has a third parameter \( \nu \) that determines the heaviness of the tails. Notationally we represent this distribution as

\[
f(x) = t_\nu(x; \mu, R)
\]

and statistically we can express the MVT distribution in terms of a Gaussian distributed variable \( z \) and a chi-square variable \( s \) as

\[
x \overset{d}{=} \mu + z(s/\nu)^{-1/2}
\]

where \( s \sim \chi^2_\nu \), and \( z \) is a zero mean Gaussian distributed vector with positive definite covariance matrix \( R \). The PDF of the MVT distribution is

\[
f(x) = \frac{\Gamma((\nu+p)/2)}{\Gamma(\nu/2)(\nu\pi)^{p/2}\vert R \vert^{1/2}} \left[ 1 + \frac{1}{\nu}(x - \mu)^T R^{-1}(x - \mu) \right]^{-(\nu+p)/2}
\]

which becomes more Gaussian as \( \nu \) increases. We note that \( R \) is the covariance matrix of the underlying Gaussian vector, but the covariance matrix of the t-distribution is \( \Sigma = \frac{\nu}{\nu - 2} R \). This distribution has more flexibility than the Gaussian distribution because it allows significant outliers to occur.
B.3 A Note on the Univariate t-distribution

The univariate t-distribution has several definitions, but we will use

\[ x \sim t_{\nu}(\delta, \sigma) \]  
\[ x \overset{d}{=} \sigma z (h/\nu)^{-1/2} \]

where

\[ z \sim \mathcal{N}(\delta, 1) \]
\[ h \sim \chi^2_{\nu}(0) \]

\( \sigma > 0, \nu \geq 2, \) and \( z \) and \( h \) are independent. This is the standard definition of the univariate t-distribution [10]. The parameter \( \delta \) is known as the non-centrality parameter. Notice that for the MVT distribution of Equation B.1, the variable \( z \) is zero mean, while \( \mu \) determines the mean of the distribution. In the univariate case, the variable \( z \) may have a non-zero mean. This mean becomes the non-centrality parameter. The non-centrality parameter not only shifts the distribution, but changes its shape too.
Appendix C

Quadratic Forms of Normal Variables

Quadratic forms find a variety of uses in detection and statistics, but we have not yet discussed what they are and why they are useful. Section C.1 provides some of the statistical and algebraic manipulations that we will use later. Section C.2 introduces the characteristic function for any quadratic form. Section C.2.1 describes how the PDF of can be obtained from the characteristic function while section C.2.2 shows how the CDF can be computed. These sections describe some background material that is needed for handling ACE and the $\beta$-detector.

C.1 Reducing Quadratic Forms in Gaussian Random Variables

In this section we define what a quadratic form is and show that when the test vector is Gaussian, we can obtain a quadratic form that is statistically equivalent to the original, but is now expressed in terms of independent components. We define the general quadratic form as

$$ q = x^T A x $$  \hspace{1cm} (C.1)
where \( \mathbf{x} \) is an \( n \) dimensional vector, and \( \mathbf{A} \) is an \( p \) by \( n \) symmetric matrix. In special cases \( q \) may be a Mahalanobis distance or the length of a projection, but here we only assume that it is symmetric.

Let \( \mathbf{x} \) be a normally distributed vector:

\[
\mathbf{x} \sim \mathcal{N} (\mu, \Sigma) \tag{C.2}
\]

where we assume \( \Sigma \) is positive definite; therefore, it has a factorization of the form

\[
\Sigma = \mathbf{U} \Sigma^{1/2} \mathbf{U}^T
\]

where \( \mathbf{S} \) is a diagonal matrix and \( \mathbf{U} \) is an orthonormal matrix. We define the square root matrix \( \Sigma^{1/2} \) as

\[
\Sigma^{1/2} = \mathbf{U} \Sigma^{1/2} \mathbf{U}^T
\]

Using this matrix, a statistically equivalent way to write \( \mathbf{x} \) is in terms of a Gaussian vector \( \hat{\mathbf{z}} \):

\[
\mathbf{x} \overset{d}{=} \Sigma^{1/2} \hat{\mathbf{z}} \tag{C.3}
\]

\[
\hat{\mathbf{z}} \sim \mathcal{N} (\bar{\mu}, \mathbf{I}) \tag{C.4}
\]

with \( \bar{\mu} = \Sigma^{-1/2} \mu \). This allows any Gaussian distributed vector to be written in terms of one with identity covariance matrix; that is, its components are unit Gaussian and are independent.

Our quadratic in Equation C.1 can now be written in terms of \( \hat{\mathbf{z}} \) as

\[
q = \mathbf{x}^T \mathbf{A} \mathbf{x} = \hat{\mathbf{z}}^T \Sigma^{1/2} \mathbf{A} \Sigma^{1/2} \hat{\mathbf{z}} = \hat{\mathbf{z}}^T \mathbf{B} \hat{\mathbf{z}}
\]

where \( \mathbf{B} = \Sigma^{1/2} \mathbf{A} \Sigma^{1/2} \) and is also symmetric; since \( \mathbf{B} \) is symmetric, it has a factorization \( \mathbf{L} \Lambda \mathbf{L}^T \), where \( \mathbf{L} \) is orthonormal and \( \Lambda \) is diagonal. By definition the matrix \( \mathbf{L} \) rotates \( \hat{\mathbf{z}} \) and by defining \( \mathbf{z} = \mathbf{L}^T \hat{\mathbf{z}} \), the expression for \( q \) becomes

\[
q = \hat{\mathbf{z}}^T \mathbf{L} \Lambda \mathbf{L}^T \hat{\mathbf{z}} = \mathbf{z}^T \Lambda \mathbf{z} \tag{C.5}
\]
with $z \sim \mathcal{N}(L^T \bar{\mu}, I)$. Since $\Lambda$ is diagonal, $q$ is simply a sum of independent random variables

$$q = \sum_{i=1}^{n} \lambda_i z_i^2$$

with $\lambda_i$ the $i$th diagonal element of $\Lambda$ and $z_i$ the $i$th element of the vector $z$. Since each $z_i$ is unit Gaussian distributed, $z_i^2$ is a non-central chi-square variable with non-centrality parameter $\delta_i$ [10]; defining the vector $d$ with elements $d_i$ as $d = L^T \bar{\mu}$ the non-centrality parameters are $\delta_i = d_i^2$.

**C.2 The Characteristic Function of Quadratic Forms in Normal Random Variables**

The characteristic function is useful for us because we can obtain the CDF and PDF of a quadratic form using the characteristic function. The benefit here is that the expression for the characteristic function is relatively simple as we will see.

Consider again the quadratic form $q$, which can be written as a sum of independent random variables

$$q = \sum_{i=1}^{n} \lambda_i z_i^2 \quad \text{(C.6)}$$

From elementary probability we know that the PDF of $q$ is equal to the convolution of the PDFs of each random variable in the sum. In turn, convolution in the “time” domain (or $x$ domain) is equivalent to multiplication in the Fourier domain. The characteristic function of a random variable is by definition the complex conjugate of the Fourier transform of the PDF. Therefore, the characteristic function of $q$ is simply the product of the characteristic functions of the random variables in the sum.

Luckily, the simple and well known expression [8] [10] for the characteristic function for
each component $\lambda_i z_i^2$ is

$$
\phi_i(t) = (1 - 2j\lambda_i t)^{-1/2} \exp \left( \frac{j\delta_i \lambda_i t}{1 - 2j\lambda_i t} \right)
$$

where $j = \sqrt{-1}$. For the quadratic form $q = z^T \Lambda z$ with $\Lambda$ diagonal and $z \sim \mathcal{N}(L^T \bar{\mu}, I)$, the characteristic function is the product

$$
\phi_q(t) = \prod_i^n \phi_i(t)
$$

(C.7)

$$
\phi_q(t) = \prod_i^n (1 - 2j\lambda_i t)^{-1/2} \exp \left( \frac{j\delta_i \lambda_i t}{1 - 2j\lambda_i t} \right)
$$

(C.8)

$$
\phi_q(t) = \prod_i^n (1 - 2j\lambda_i t)^{-1/2} \exp \left( \sum_k^n \frac{j\delta_k \lambda_k t}{1 - 2j\lambda_k t} \right)
$$

(C.9)

Through this expression both the PDF and CDF of $q$ can be calculated as will be discussed in sections C.2.1 and C.2.2 respectively.

### C.2.1 Obtaining the PDF from the Characteristic Function

The relationship between the characteristic function and the PDF of a random variable is simple; the characteristic function is the complex conjugate of the Fourier transform of the PDF and is defined as

$$
\phi(t) = \int_{-\infty}^{\infty} f(x) \exp \{ jtx \} \, dx
$$

where $f(x)$ is the PDF of the random variable. In our current scenario, we are given or can compute $\phi(t)$ and would like to find $f(x)$. The classic transform [23] from characteristic function to PDF is

$$
f(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \phi(t) \exp \{ -jtx \} \, dt
$$

but may be equivalently be written in the more familiar form

$$
f(x) = \int_{-\infty}^{\infty} \frac{1}{2\pi \phi(2\pi t)} \exp \{ j2\pi tx \} \, dt
$$

which coincides with the classic definition of the inverse Fourier Transform [16].
C.2.2 Obtaining the CDF from the Characteristic Function

Since the CDF of the random variable $q$ does not yield a simple closed form expression the simplicity of characteristic function is attractive. The most common inversion formula is the Gil-Pelaez [7] formula

$$F(x) = \frac{1}{2} - \frac{1}{\pi} \int_0^\infty \frac{\Im\{\phi(t)e^{jxt}\}}{t} dt$$

where $\Im\{y\}$ denotes the imaginary component of $y$. Using the geometry of the complex plane and the characteristic function $\phi_q(t)$, given in (C.8), Imhof [8] showed that an equivalent formula is

$$F(x) = \frac{1}{2} - \frac{1}{\pi} \int_0^\infty \frac{\sin(\theta(x,u))}{u\rho(u)} du$$

(C.10)

with

$$\theta(u) = \frac{1}{2} \sum_{r=1}^m \left\{ \tan^{-1}(\lambda_r u) + \delta_r^2 \lambda_r u (1 + \lambda_r^2 u^2)^{-1} \right\} - \frac{1}{2} xu$$

$$\rho(u) = \prod_{r=1}^m (1 + \lambda_r^2 u^2)^{1/4} \exp \left\{ \frac{1}{2} \sum_{r=1}^m \frac{(\delta_r \lambda_r u)^2}{(1 + \lambda_r^2 u^2)} \right\}$$

Although, this integral is over an infinite interval, the integrand decays with $\frac{1}{u}$ and so can be approximated using only a finite interval.


