

23

# Time-To-Intercept Estimation for an Optically Radiating Source

by

John A. Richards

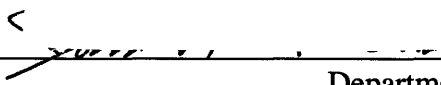
Submitted to the Department of Electrical Engineering and Computer Science  
in Partial Fulfillment of the Requirements for the Degrees of  
Bachelor of Science in Electrical Science and Engineering  
and Master of Engineering in Electrical Engineering and Computer Science  
at the Massachusetts Institute of Technology

May 28, 1996

Copyright 1996 John A. Richards. All rights reserved.

The author hereby grants to M.I.T. and Lockheed Martin IR Imaging Systems permission to reproduce and to distribute paper and electronic copies of this thesis document in whole or in part, and to grant others the right to do so.

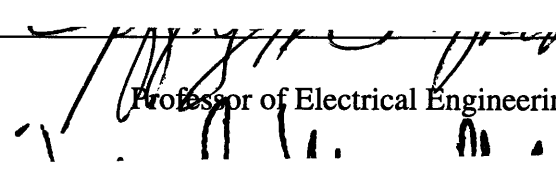
Author

  
Department of Electrical Engineering and Computer Science  
May 28, 1996

Certified by

  
Kurt Nordhaus  
Senior Engineering Fellow, Lockheed Martin IR Imaging Systems  
Thesis Supervisor

Certified by

  
Jeffrey H. Shapiro  
Professor of Electrical Engineering, Massachusetts Institute of Technology  
Thesis Supervisor

Accepted by

  
F. R. Morgenthaler  
Chairman, Department Committee on Graduate Theses

MASSACHUSETTS INSTITUTE  
OF TECHNOLOGY

JUN 11 1996 Barker Eng

## Thesis Release Letter

Professor F. R. Morgenthaler  
Department of Electrical Engineering and Computer Science  
Room 38-476, M.I.T.  
Cambridge, MA 02139

Subject: Master of Engineering Thesis of John A. Richards

Dear Professor Morgenthaler:

I have reviewed the attached thesis of John A. Richards on behalf of Lockheed Martin IR Imaging Systems. This thesis is within the scope of the thesis proposal as previously approved and does not contain any material that is objectionable to Lockheed Martin IR Imaging Systems. It is also approved for its technical content.

It is understood that the actual thesis document will be the permanent property of M.I.T. and will be placed in the M.I.T. Library within one month after the date of submission. Lockheed Martin IR Imaging Systems agrees that M.I.T. shall have the nonexclusive right to reproduce, publish, and distribute the thesis.

---

Kurt Nordhaus  
Authorized Official of Lockheed Martin IR Imaging Systems

## Acknowledgments

I owe numerous people a sincere “thank you” for their direct or indirect help in bringing this thesis to life. I have tried to give them thanks in real-time as the thesis progressed, but, in any case, they deserve some recognition here, too.

First of all, I need to thank the Threat Warning department supervisor, Tom Mascoli. He is the one that gave me a chance to work in the Signal Processing group in the first place. Without him, this thesis would not exist. Thank you, Tom, for your help back when our little corner of Lockheed Martin was still called Loral.

Second, I owe thanks to my thesis supervisors, Kurt Nordhaus and Professor Jeffrey Shapiro. Kurt, my Lockheed supervisor, offered invaluable technical advice and research suggestions throughout the project. He also allowed me the freedom to define the course of my research. His technical expertise is exceeded only by his skills as a mentor. I know of no other person who more deserves the title of “Fellow.”

Professor Jeffrey Shapiro, my MIT supervisor, offered valuable suggestions and an objective point of view during the research phase of this thesis; later, he read countless revisions of this document. His careful scrutiny and close reading made this thesis much better than it would have been otherwise. I want to thank him not simply for his editorial skills, but also for inspiring me to produce the best work I am capable of. (I also want to apologize for any long-term damage to his thumb. Professor Shapiro, I’ve taken the pin out of the voodoo doll.)

Also much deserving of praise is the entire crew of SIP Lab regulars (an oxymoron if ever there was one). Mike Bulpett, Jennifer Roy, Tim Boyd, Sheri Crowley, and Robin Williams (along with Kurt and Tom) have all offered help, moral support, and lots of good conversation and laughs throughout the past year. I owe them all thanks. I hope they’ll all help out with the “research” I plan to conduct in Building 4 this summer. I’m buying.

There are three more people I want to mention here, each of whom deserve far more thanks than I can possibly give them. Two of these people are my parents. Without their support and encouragement, I would not be at MIT. Mom and Dad, I can’t thank you enough for all you’ve done for me. You’ve made this possible.

The final person I want to thank is my fiancée, Becky. She has been supportive and understanding since day one. She has smoothed out the rough times and made the good times even better. This thesis would not have been possible without her. Becky, this one’s for you.

*For Becky*

# **Time-To-Intercept Estimation for an Optically Radiating Source**

by

**John A. Richards**

Submitted to the Department of Electrical Engineering and Computer Science

May 28, 1996

in partial fulfillment of the requirements for the degrees of  
Bachelor of Science in Electrical Science and Engineering and  
Master of Engineering in Electrical Engineering and Computer Science

## **Abstract**

Consider a detector array and a radiating source in relative motion through the earth's atmosphere. Suppose that the trajectory of the source is such that it will intercept the trajectory of the detector array. Two parameters can usefully characterize the relative source trajectory. The first is angle-of-arrival (AOA), specified in terms of azimuth and elevation from the local coordinate system of the detector array. The second is time-to-intercept (TTI), the amount of time remaining until the source and detector array are co-located. This thesis considers the problem of TTI estimation using six passive imaging (i.e., pixel array) sensors operating in the ultraviolet (UV) region of the spectrum. Atmospheric, source, sensor, and kinematic models are provided by Lockheed Martin IR Imaging Systems' (LIRIS) Threat Warning Engagement Simulator (TWES). Simplifying assumptions are made in order to facilitate the development of two TTI estimation algorithms. The first algorithm utilizes the scattered radiation received by the detector array. The output of the line-of-sight sensor is parametrized into a small-dimensional set and compared to a canonical set of parametrized sensor output vectors compiled from TWES simulations in order to obtain a TTI estimate. It is shown that the performance of this algorithm under test conditions is superior to that of previous LIRIS algorithms. The second algorithm relies on an approximation to the direct radiation received by the sensors to produce a TTI estimate. This algorithm fits integrated pixel data from successive sensor frames to an equation governing direction radiation growth. A TTI estimate is obtained by finding the minimum squared-error fit. The performance of this algorithm is shown to be inferior to that of the first algorithm, but still superior to that of previous efforts whenever it produces a convergent estimate.

Thesis Advisor: Jeffrey H. Shapiro

Title: Professor of Electrical Engineering, Massachusetts Institute of Technology

Thesis Advisor: Kurt Nordhaus

Title: Senior Engineering Fellow, Lockheed Martin IR Imaging Systems

# Table of Contents

<b>Abstract</b> .....	<b>5</b>
<b>List of Figures and Tables</b> .....	<b>9</b>
<b>1.0 INTRODUCTION</b> .....	<b>10</b>
1.1 Threat Warning .....	10
1.2 Trajectory Estimation.....	10
1.3 Our System.....	11
1.4 Overview.....	12
<b>2.0 MODELS, ASSUMPTIONS AND BACKGROUND</b> .....	<b>13</b>
2.1 Source Radiation Phenomenology.....	13
2.2 The Sensor .....	13
2.3 Radiation Propagation Phenomenology.....	16
2.3.1 Photon transmission in a vacuum .....	16
2.3.2 Photon transmission in the atmosphere.....	17
2.4 Kinematic models .....	20
2.5 The TWES .....	21
2.5.1 Source modeling .....	21
2.5.2 Sensor modeling.....	21
2.5.3 Atmospheric and kinematic modeling .....	21
2.5.3.1 Altitude .....	22
2.5.3.2 Initial range .....	22
2.5.3.3 Closing velocity .....	22
2.5.3.4 Azimuth.....	23
2.5.3.5 Elevation .....	23
2.5.3.6 Absorption.....	23
2.5.3.7 Scattering .....	23
2.6 Assumptions.....	24
2.6.1 The sensor platform is in level flight at a constant velocity .....	24
2.6.2 Atmospheric conditions remain constant throughout an engagement .....	24
2.6.3 The source approaches along a straight line at a constant velocity.....	24
2.6.4 The source AOA is [0,0].....	24
2.6.5 The source is an isotropic blackbody at constant temperature .....	25
2.7 Past Work .....	25
2.7.1 A curve-fitting approach .....	26
2.7.2 An extended Kalman filtering approach .....	26

2.7.3	A measurement-matching approach .....	27
2.8	New Conditions .....	27
<b>3.0</b>	<b>THE PARAMETER-MATCHING ALGORITHM.....</b>	<b>29</b>
3.1	Sensor Output Parametrization .....	30
3.1.1	The abundance of information .....	30
3.1.2	Pixel integration rings .....	30
3.1.3	Duration of data collection.....	33
3.2	The Canonical Set.....	34
3.2.1	Atmospheric resolution within the canonical set .....	34
3.2.2	Spatial resolution within the canonical set.....	35
3.3	The Cost Function.....	36
3.4	The Search Method .....	40
3.4.1	A coarse-to-fine search .....	40
3.4.2	Search path abort conditions .....	41
3.5	Summary .....	43
3.6	Analysis.....	44
3.6.1	The test set .....	44
3.6.2	Performance .....	45
<b>4.0</b>	<b>THE CURVE-FITTING ALGORITHM.....</b>	<b>57</b>
4.1	Phenomenology Revisited .....	57
4.2	Sensor Output Growth Parameters.....	59
4.3	The Newton-Raphson method .....	62
4.3.1	Error minimization.....	62
4.3.2	Newton-Raphson for explicit equation solving .....	63
4.3.3	Newton-Raphson for error minimization.....	65
4.3.4	Other Newton-Raphson design issues .....	67
4.3.4.1	Choice of FOV .....	67
4.3.4.2	Newton-Raphson abort conditions.....	70
4.3.4.3	Error threshold .....	70
4.3.4.4	Duration of data collection.....	71
4.3.4.5	Wavelength dependence correction factor.....	71
4.3.4.6	Initial guess .....	72
4.3.4.7	Local averaging.....	72
4.4	Summary .....	73
4.5	Analysis.....	73
4.6	Curve-Fitting Algorithm Enhancements and Modifications.....	83
<b>5.0</b>	<b>CONCLUSIONS .....</b>	<b>84</b>
5.1	Algorithm Summary .....	84

5.1.1	The parameter-matching algorithm.....	84
5.1.2	The curve-fitting algorithm .....	84
5.2	Comparison of Algorithm Performance and Applicability.....	84
5.2.1	The parameter-matching algorithm.....	85
5.2.2	The curve-fitting algorithm .....	86
5.2.3	Comparison of the two algorithms.....	87
5.3	Implementability .....	88
5.4	The Real World .....	89
5.5	Future Work .....	91
	<b>References .....</b>	<b>93</b>



## List of Figures and Tables

FIGURE 1.1:	Angle-of-arrival: azimuth and elevation .....	11
FIGURE 2.1:	Sensor block diagram .....	14
FIGURE 2.2:	Sensor output growth.....	18
FIGURE 2.3:	Sensor output profile evolution .....	20
FIGURE 3.1:	Pixel integration rings.....	31
FIGURE 3.2:	Relative ring contribution to total sensor output .....	32
TABLE 3.1:	Canonical set atmospheres .....	35
FIGURE 3.3:	Distance between vectors in two dimensions .....	37
FIGURE 3.4:	Family of source radiant intensities.....	38
FIGURE 3.5:	Distance between vector and ray in two dimensions.....	39
FIGURE 3.6:	Canonical set angular deviation surface .....	42
FIGURE 3.7:	Cross-sections of canonical set angular deviation surface .....	43
FIGURE 3.8:	Parameter-matching algorithm: mean TTI estimate vs. TTI.....	47
FIGURE 3.9:	Effect of high ozone concentration atmospheres on bias .....	48
FIGURE 3.10:	Effect of ozone concentration on three-ring parametrization.....	49
FIGURE 3.11:	Parameter-matching algorithm: TTI estimate rms error vs. TTI.....	51
FIGURE 3.12:	Effect of high ozone concentration atmospheres on rms error.....	52
FIGURE 3.13:	Parameter-matching algorithm: mean TTI estimate vs. TTI, atmosphere subsets ( $v = 500$ m/s, $J = 50$ mW/sr).....	53
FIGURE 3.14:	Parameter-matching algorithm: TTI estimate rms error vs. TTI, atmosphere subsets ( $v = 500$ m/s, $J = 50$ mW/sr).....	55
FIGURE 4.1:	Family of $c_1$ curves.....	60
FIGURE 4.2:	Family of $c_2$ curves.....	61
FIGURE 4.3:	Family of $c_3$ curves.....	61
FIGURE 4.4:	Log sensor output growth.....	62
FIGURE 4.5:	Newton-Raphson iteration in two dimensions .....	64
FIGURE 4.6:	Newton-Raphson iteration: incorrect convergence .....	64
FIGURE 4.7:	Effect of FOV on mean TTI estimate.....	68
FIGURE 4.8:	Effect of FOV on isocount contours.....	69
FIGURE 4.9:	SNR and scattering thresholds.....	69
TABLE 4.1:	Curve-fitting algorithm: convergence percentage .....	74
FIGURE 4.10:	Fraction of convergent TTI estimates vs. TTI.....	75
FIGURE 4.11:	Curve-fitting algorithm: mean TTI estimate vs. TTI.....	76
FIGURE 4.12:	Curve-fitting algorithm: TTI estimate rms error vs. TTI.....	78
FIGURE 4.13:	Curve-fitting algorithm: mean TTI estimate vs. TTI, atmosphere subsets ( $v = 500$ m/s, $J = 50$ mW/sr) .....	80
FIGURE 4.14:	Curve-fitting algorithm: TTI estimate rms error vs. TTI, atmosphere subsets ( $v = 500$ m/s, $J = 50$ mW/sr).....	82

# 1.0 INTRODUCTION

## 1.1 Threat Warning

In a modern combat environment, military personnel often rely on detection and estimation systems to warn them of approaching potential threats [1]. An ideal threat warning system would distinguish between threats and non-threats while providing information about the trajectory of an approaching object.

Threat warning systems can be categorized into two broad classes: active systems and passive systems. Active threat warning systems, such as radar, detect and track an object by emitting radiation. Some of this radiation is reflected back to the detection system by the approaching object, providing information about the location, size, and trajectory of the object [2]. Active detection systems are often undesirable, because their radiation emission provides information that could be useful to an approaching threat.

Passive threat warning systems do not emit radiation. Instead, they use radiation emitted by the approaching object to estimate its trajectory [3]. For this reason, passive systems are often preferable to active systems, especially in a combat environment. Lockheed Martin IR Imaging Systems (LIRIS) is a leading designer of passive threat warning systems. LIRIS's threat warning systems rely on electro-optic sensors to detect radiation emitted by an object and to estimate the object's position and trajectory.

In this thesis, we will be concerned with developing a passive threat warning algorithm to provide information about the trajectory of an approaching object.

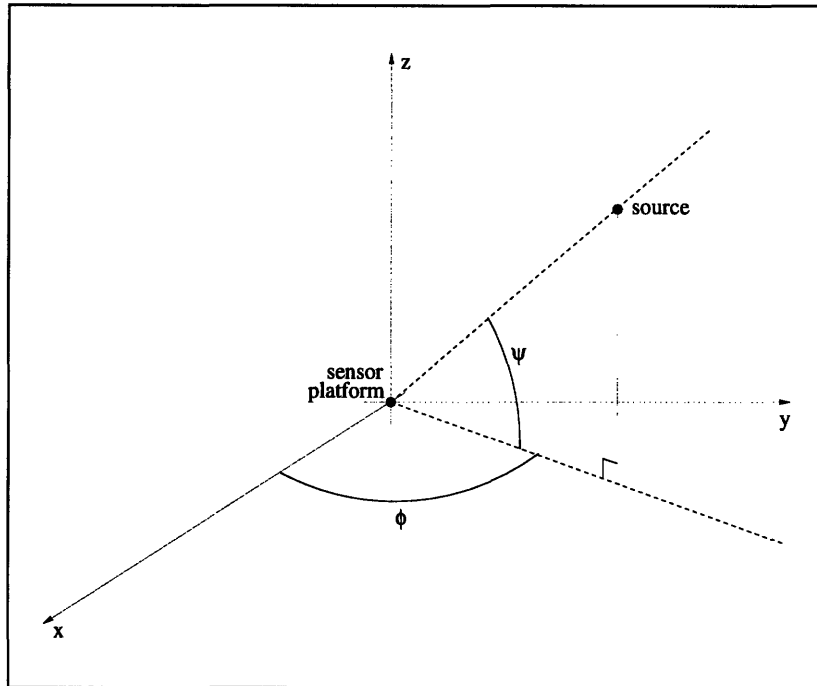
## 1.2 Trajectory Estimation

Two parameters can usefully parametrize a trajectory. The first is angle-of-arrival (AOA), usually specified in terms of azimuth and elevation from the local coordinate system of the sensor platform. AOA is the direction from which the radiant source is approaching. Figure 1.1 depicts a representation of AOA. Azimuth in this figure is denoted as  $\phi$ ; elevation is denoted as  $\psi$ . The vector  $[\phi, \psi]$  comprises AOA.

The second parameter of interest is time-to-intercept (TTI). This is the amount of time remaining until the source and sensor platform are co-located. TTI and AOA estimates can be used to coordinate anti-threat procedures that can significantly increase the survival probability of the sensor platform. This, of course, is the ultimate goal of any threat warning system.

With today's high-resolution imaging (i.e., pixel array) electro-optic sensors, AOA estimation is relatively simple for a single-source engagement [1]. Since the pixel resolution of a modern imaging sensor is much finer than the minimum acceptable granularity for the AOA estimate, a more-than-acceptable estimate can be produced simply by identifying the brightest pixel on the sensor. The sophisticated AOA estimation algorithms that were used with single-pixel, non-imaging sensors in the past are no longer needed.

TTI estimation, on the other hand, remains a formidable problem. Though imaging sensors provide more information than their single-pixel ancestors, there is still no reliable procedure for obtaining an accurate TTI estimate. The increased spatial resolution which greatly simplifies the AOA problem does not solve the TTI problem.



**FIGURE 1.1: Angle-of-arrival: azimuth and elevation**

### 1.3 Our System

This thesis will consider the problem of passive TTI estimation using six imaging missile warning system sensors. These sensors operate in the ultraviolet (UV) region of the spectrum. They are mounted on the sensor platform on mutually orthogonal axes. The sensors used in our system are high-resolution imaging sensors; the precise resolution and pixel array size, however, are classified.

One advantage to using UV sensors is the reduction of background clutter in comparison to that seen in other regions of the spectrum (i.e., infrared or visible). There are few natural sources of UV radiation in the earth's atmosphere [1]. This lack of ambient UV radiation enables UV sensors to detect and track a source even during the day. One important consideration when using UV sensors is the relatively large amount of scattering that occurs during UV radiation propagation through the atmosphere [4]. Because our sensors have a wide field-of-view (FOV), we are able to detect any scattered radiation that reaches our sensor platform. Scattering significantly affects the TTI estimation problem.

Our sensor platform is a military aircraft. The extreme limitations imposed on the size and weight of on-board systems deployed on such aircraft constrain the design of our algorithm. It should be as simple as possible, in order to minimize the computational resources that need to be brought on-board; an algorithm that requires significant computational power or storage will require more space and mass than one that does not require such resources. Additionally, an on-board algorithm must produce a reliable TTI estimate in real-time. These constraints suggest that an algorithm should be as computationally simple as possible. We must design our algorithms with this goal in mind.

Another goal of our TTI estimation work is robustness. A missile-aircraft engagement can take many different forms. The signal that reaches our sensor platform might vary widely with missile range and closing velocity, or with atmospheric conditions. A useful TTI estimation algorithm must be tolerant to a broad selection of possible engagement parameters.

## **1.4 Overview**

Chapter 2 of this thesis describes the models and assumptions underlying the two algorithms for TTI estimation that will be developed herein. This chapter outlines the models for every relevant facet of our system: the source, the atmosphere, the sensor, and the kinematic relationships governing missile approach. LIRIS's Threat Warning Engagement Simulator (TWES) is described. Chapter 2 contains an explanation of the assumptions made in order to facilitate algorithm development and testing. It concludes with a discussion of past TTI estimation efforts.

Chapter 3 outlines the development of the first of two TTI estimation algorithms. This first algorithm attempts to relate scattered radiation data to physical parameters, such as source-to-sensor platform range and closing velocity, in order to produce a TTI estimate. A scheme for parametrizing the sensor output into a useful form is presented. Limiting factors in the design process of this algorithm are discussed. Finally, the performance of this algorithm is characterized by running it on a test set meant to mirror a broad sampling of possible real-world scenarios.

Chapter 4 describes the second TTI estimation algorithm. This algorithm uses a curve-fitting approach to relate sensor output growth explicitly to physical parameters. An intensity growth function explicitly dependent on TTI is derived, and this function is used as the basis for the curve-fit. An overview of the numerical processes underlying the curve-fitting is presented. Specific design issues are discussed. Finally, the same test set used for evaluation of the first algorithm is used to evaluate the curve-fitting algorithm.

Chapter 5 concludes the thesis. It summarizes the findings of the previous chapters. It returns to some of the issues discussed in previous chapters and discusses the algorithms' applicability in the real world. Finally, this chapter presents some suggestions for further work in the area of TTI estimation.

## 2.0 MODELS, ASSUMPTIONS AND BACKGROUND

This chapter describes the phenomenology underlying source radiation, atmospheric photon transmission, sensor operation and source-sensor platform kinematics. This chapter also describes the models and assumptions underlying the two TTI estimation algorithms described in this thesis. It describes LIRIS's Threat Warning Engagement Simulator (TWES). It contains an explanation of the assumptions made concerning the atmosphere, source trajectory and source characteristics upon which our algorithms are based. This chapter concludes with an examination of past work in the area of TTI estimation.

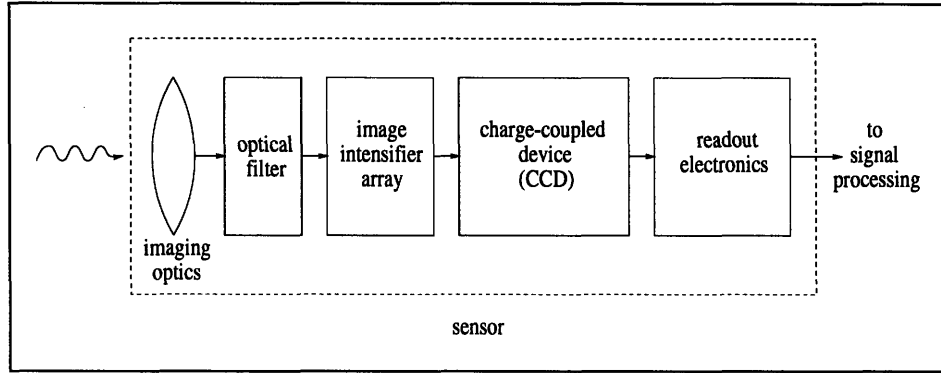
### 2.1 Source Radiation Phenomenology

Any ultraviolet-radiating object in the atmosphere is a source as far as the sensor platform is concerned. In fact, for our purposes, the only important source characteristics are those which influence the emission of radiation. We can describe a source's radiation characteristics by specifying its spectral radiance. Spectral radiance has units  $[\text{W m}^{-2} \text{sr}^{-1} \mu\text{m}^{-1}]$  and has a physical interpretation as the power radiated per unit of source area per solid angle per unit bandwidth (in wavelength units) [5]. We shall denote spectral radiance as  $N(\lambda)$ . Spectral radiance allows us to describe a source that radiates over a wide range of frequencies. Most sources exhibit a spectral radiance function that has a strong dependence on wavelength; [6] contains a number of spectral radiance functions for different sources as examples.

In this thesis, we shall consider a constant temperature isotropic blackbody source. An isotropic source radiates power equally in every direction, and thus looks the same to an optical receiver when viewed from any direction. A blackbody is a perfect absorber and emitter of radiation. The spectral radiance of a blackbody as a function of temperature is well-known; see, e.g., [7]. The spectral radiance of an isotropic blackbody can be completely characterized by specification of only its surface area and temperature, or alternately, its radiant intensity. Radiant intensity is the integration of spectral radiance over surface area and wavelength, and thus has units  $[\text{W sr}^{-1}]$ . We shall use the symbol  $J$  to denote source radiant intensity.

### 2.2 The Sensor

Our system utilizes a generic threat warning spatial sensor which incorporates the fundamental functional components LIRIS has used to design several types of spatial sensors. Our sensors detect radiation in the UV region of the spectrum. Our sensor has five basic components: imaging optics, an optical filter, an image intensifier array, a charge-coupled device (CCD), and a readout electronics stage (see figure 2.1 for a block diagram). The imaging optics on the sensor face admit and focus radiation energy. The optical filter influences the spectral response of the sensor by determining what wavelengths of light reach the image intensifier array. The optical filter is designed to allow UV radiation to pass and to attenuate the energy at other wavelengths as much as possible.



**FIGURE 2.1: Sensor block diagram**

The image intensifier array provides signal amplification. The image intensifier acts essentially as a photo-multiplier with an optical output; a single photon on an image intensifier input will result in many photons at the image intensifier output. The image intensifier array thus provides amplification to radiation incident on the sensor.

The image intensifier array output illuminates a CCD. A CCD changes optical input (photons) into electrical output (electrons). Each photon that reaches the CCD generates an electron that shows up at the output of the CCD as a discrete packet of charge. The readout electronics stage counts these packets of charge to generate a photon count as output. The readout electronics stage of the sensor contains a multiplexer to enable transmission of the array of individual pixel outputs as a stream of serial data to the signal processor.

The first three components of the sensor determine the sensor's conversion efficiency,  $\kappa(\lambda)$ . Conversion efficiency has units  $[\text{counts m}^2 \text{J}^{-1}]$  and depends on various parameters, such as incident radiation wavelength and sensor temperature and surface area. We will assume that all sensor parameters are fixed, and thus  $\kappa$  is a function only of wavelength. Our sensor exhibits a very narrow spectral response; that is, its conversion efficiency peaks in the UV region of the spectrum and is nearly zero in other regions. Incident radiation in the narrow band in the UV spectrum will produce marked sensor response, while radiation in, say, the visible or infrared regions will not.

We learn from [8] that for a source of surface area  $A_s$  and a sensor of area  $A_r$ , separated by range  $r(t)$  (note that the time dependence is due to the evolving source-sensor platform kinematic relationship), the power  $P(t)$  incident on the sensor face is given by:

$$P(t) = \int N(\lambda) \frac{A_r}{r^2(t)} A_s d\lambda \quad \text{Equation 2.1}$$

which allows us to determine the photon count rate  $\Phi(t)$ :

$$\Phi(t) = \int N(\lambda) \frac{\kappa(\lambda)}{r^2(t)} A_s d\lambda \quad \text{Equation 2.2}$$

since  $\kappa$  incorporates the sensor area.  $\Phi(t)$  has units of [counts  $s^{-1}$ ]. If we assume a spectral response that is nonzero (and nearly constant) over only a narrow bandwidth  $B$ , then we may write:

$$\Phi(t) = \frac{A_s N \kappa B}{r^2(t)} \quad \text{Equation 2.3}$$

where  $N$  is the spectral response in the optical filter's passband.

At this point, let us momentarily step aside from our sensor description in order to describe the Poisson probability distribution, which is important to the sensor phenomenology. Recall that for a Poisson variable  $x$  with mean  $\mu$ , the probability of observing  $M$  counts obeys [9]:

$$Pr \{x = M\} = \frac{\mu^M e^{-\mu}}{M!} \quad (M = 0, 1, 2, \dots) \quad \text{Equation 2.4}$$

Recall also that the variance  $\sigma^2$  of a Poisson variable is equal to its mean, so that the signal-to-noise ratio (SNR) of a Poisson variable is:

$$SNR \equiv \frac{\mu}{\sigma} = \frac{\mu}{\sqrt{\mu}} = \sqrt{\mu} \quad \text{Equation 2.5}$$

The Poisson distribution is important to us because the arrival of photons at a sensor is not deterministic: it is a Poisson process [3]. The number of counts received at the sensor during a period of time of duration  $T$  is a Poisson-distributed random variable with mean:

$$\mu(t) = \int_t^{(t+T)} \Phi(\tau) d\tau \quad \text{Equation 2.6}$$

From equation 2.3, we see that the only time-dependent quantity in  $\Phi(t)$  is  $r(t)$ , the source-to-sensor platform range. If we assume that the change in  $r(t)$  over the interval  $T$  is small compared to  $r(t)$ , then  $\Phi(t)$  is nearly constant over the interval  $T$ , and we may write:

$$\mu(t) = T\Phi(t) \quad \text{Equation 2.7}$$

or, expanding:

$$\mu(t) = \frac{A_s N \kappa B T}{r^2(t)} \quad \text{Equation 2.8}$$

By choosing an integration period  $T$  we can thus influence the magnitude of our sensor output (and its SNR). The inverse of the integration period is also known as the sensor frame rate, because a new sensor output (or set of pixel outputs) is produced every  $T$  seconds.

## 2.3 Radiation Propagation Phenomenology

We are concerned with the propagation of radiation from a source to the sensors on our sensor platform. This section provides a phenomenological description of this process. We begin with the simple, but unrealistic, case of vacuum propagation.

### 2.3.1 Photon transmission in a vacuum

In the absence of an atmosphere, every photon emitted by the source will travel uninhibited in a straight line [10]. Some photons will then be intercepted by the sensor platform. However, only those sensors which are in a direct line-of-sight from the source will experience any photon flux; the outputs of these sensors would be governed by a simple inverse-square law relationship. Thus, if the source is located in the FOV of sensor 1 only, the Poisson mean of the sensor output vector (assuming, for notational simplicity, a set of six single-pixel sensors) is:

$$\mathbf{h} = \begin{bmatrix} \frac{I_0}{r^2(t)} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \text{Equation 2.9}$$

where:

$$I_0 = A_s N \kappa B T \quad \text{Equation 2.10}$$

which gives  $\mathbf{h}$  units of simply [counts]. (The symbol  $\mathbf{h}$  is used to denote the Poisson mean sensor output, or “noiseless sensor output” by LIRIS convention.) Note that equation 2.9 depends on the assumption of a source of constant spectral radiance, a narrow spectral response for the sensor, and a nearly constant range over the integration period.

Let us consider a source of constant radiant intensity that is approaching the sensor platform at a constant velocity.  $I_0$  will remain constant while  $r(t)$  decreases linearly. As this



source approaches the sensor platform, the output of sensor 1 increases, while sensors 2-6 continue to produce zero outputs. A simple curve-fit applied to the sensor 1 output will predict the time at which the intensity will reach infinity, corresponding to the co-location of the source and sensor platform (i.e.,  $r(t) = 0$ ). The duration between the current time and the time of co-location is TTI.

### 2.3.2 Photon transmission in the atmosphere

Our system does not operate in a vacuum. Our system operates in the earth's atmosphere. This is a lossy, scattering medium, due to the random interactions between individual photons and atmospheric molecules and aerosols [8]. Photons emitted by the source are variously scattered and absorbed on their way to the sensor platform. The phenomena of absorption and scattering are known in ensemble as extinction.

Absorption results in a reduction in the power received at the sensor platform. The amount of reduction is dependent on the path length between the source and the sensor platform, or simply the range  $r$ . The chance that a photon will be absorbed en route to the sensor platform increases as the range between the source and sensor platform increases.

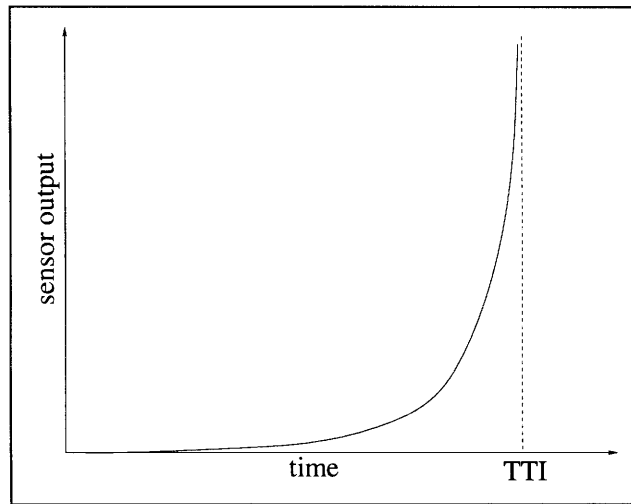
The possibility of absorption forces a modification of equation 2.6. In a homogenous atmosphere, the noiseless sensor outputs are now governed by:

$$\mathbf{h} = \begin{bmatrix} I_0 e^{-k_a r(t)} \\ r^2(t) \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \text{Equation 2.11}$$

where  $k_a$  is known as the "absorption coefficient" with units [ $\text{m}^{-1}$ ] (and with a wavelength dependence) and  $I_0$  is defined as before. This equation is again dependent on a constant source spectral radiance, a narrow spectral response and a nearly constant range. This equation is also dependent on the assumption that the absorption coefficient is nearly constant over the sensor's spectral bandwidth. Note also that in an inhomogenous atmosphere, equation 2.11 may not hold strictly due to the fact that  $k_a$  will be variable along the path from source to sensor platform, due to variable atmospheric parameters such as aerosol concentration.

The exponential term in equation 2.11 also complicates the curve-fitting mentioned above. A curve-fit now requires estimation of an additional parameter,  $k_a$ . It is still possible, though more difficult, to predict TTI from observation of  $\mathbf{h}$  in an absorptive atmosphere. Figure 2.2 depicts the growth of the sensor output with respect to time in equation 2.11.

The deflection of photons, known as scattering, is responsible for by far the biggest problem facing TTI estimation. A photon may be scattered once, or multiple times, or not at all on its journey from source to sensor platform. Scattering reduces the amount of direct radiation that reaches the sensor platform; scattering, like absorption, can be accommodated analytically into our model with the introduction of a path length coefficient  $k_s$ . Unlike the absorption coefficient  $k_a$ , however, the scattering coefficient creates a sensor output that is, in general, nonzero everywhere. The possibility of scattering means that a photon can be received at the sensor platform from any direction, and thus the output of all sensors is nonzero.



**FIGURE 2.2: Sensor output growth**

Scattering forces another modification to the sensor output equation. Again assuming atmospheric homogeneity, a narrow spectral response, a constant source spectral radiance, nearly constant absorption and scattering over the sensor's narrow spectral bandwidth, and a nearly constant range over the integration period, we can write:

$$\mathbf{h} = \begin{bmatrix} \frac{I_0 e^{-kr(t)}}{r^2(t)} + \Theta_1(t) \\ \Theta_2(t) \\ \Theta_3(t) \\ \Theta_4(t) \\ \Theta_5(t) \\ \Theta_6(t) \end{bmatrix} \tag{Equation 2.12}$$

where:

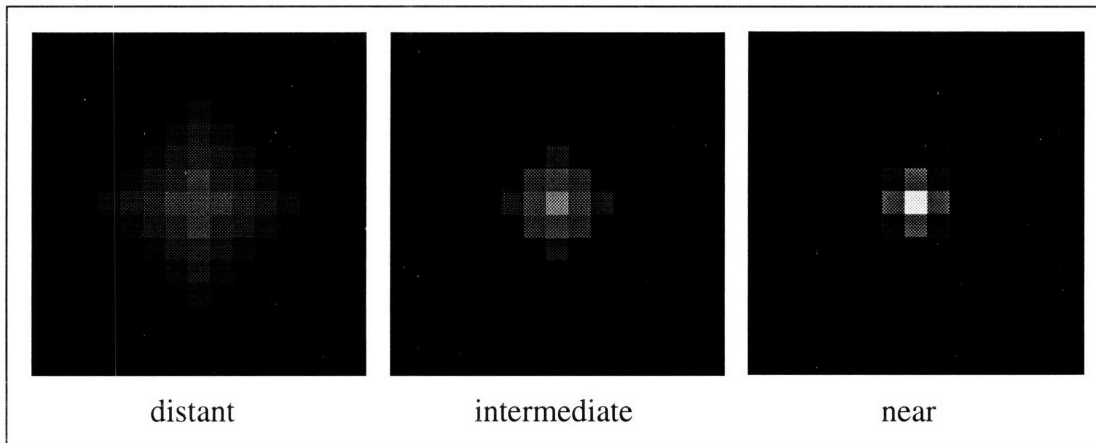
$$k = k_a + k_s \quad \text{Equation 2.13}$$

is the extinction coefficient. The  $\Theta_i(t)$  in equation 2.12 are defined as the contribution of scattered radiation to the output of the  $i$ th sensor at time  $t$ . In general, there is no closed-form representation for the  $\Theta_i(t)$  in the above equation: scattering varies widely with atmospheric conditions, source-to-sensor platform range, and even AOA. Given a set of atmospheric conditions, it might be possible to compute a precise or approximate value of  $\Theta_i(t)$  for each sensor using a specific scattering model; however, it is difficult to generalize this computation to variable atmospheric conditions. This problem has given rise to a voluminous amount of literature. There have been many attempts to characterize and predict scattering in different atmospheric environments, or to solve what is known as the “multiple scattering problem” [5]. The reader is directed to [11] and [12] for a more complete examination of scattering and characterization attempts than can be given here.

From a TTI estimation standpoint, scattering is greatly detrimental. The simple curve-fit that is straightforward in a vacuum, or even reasonably straightforward in an absorptive atmosphere, becomes intractable in a scattering environment. To perform a curve-fit in a scattering environment, we require an equation relating the  $\Theta_i(t)$  in equation 2.12 directly to physical parameters such as range and closing velocity. As noted above, a simple general analytical solution to the scattering problem is not available if we wish to operate in a wide range of atmospheric conditions.

Though it greatly complicates the TTI estimation problem, atmospheric scattering is not completely arbitrary, and several useful statements can be made about its behavior in a homogenous atmosphere. The scattering profile seen at the sensor platform is roughly symmetric about AOA since the scattering probability distribution is symmetric [8]. The relative magnitude of the scattered radiation compared to the direct radiation increases with range. The intensity profile of a nearby source will be very peaked and centralized. A distant source, on the other hand, will appear more blurry, and will result in a less peaked response. Figure 2.3 provides an illustration of a typical change in the sensor output profile as a source approaches an imaging sensor. Each frame in this figure has been normalized so that the total sensor output for each frame is constant.

Scattering leaves us two potentially viable alternatives for TTI estimation. The first of these is to attempt to divine some information about TTI from the observed shape of the scattering profile by empirical observations. This is the basis of the TTI estimation algorithm proposed in chapter 3 of this thesis. The basis for the other algorithm is as follows: with a high-resolution sensor, it might be possible to constrict FOV and to reduce the contribution of scattering (the  $\Theta_i(t)$  in equation 2.12) to the point where equation 2.11 becomes a good approximation to equation 2.12 and curve-fitting once again becomes a valid approach. The remaining chapters of this thesis describe, develop, and evaluate these two algorithms.



**FIGURE 2.3: Sensor output profile evolution**

## 2.4 Kinematic models

We have described the models used to simulate the transmission of photons from source to sensor and the translation of photon flux to sensor output. We will now provide a description of how the relationship between the source and the sensor platform changes with time.

Most missile trajectories evolve in three stages [1]. The first stage is ignition. This stage commences when the missile is fired. During the ignition stage, the missile accelerates to a constant closing velocity, using a disproportionately large amount of its propulsion materials. This stage ends as the missile's acceleration drops to zero and it reaches a constant velocity.

The second stage of missile flight (for most missiles of concern) is known as proportional navigation. In proportional navigation, the missile maintains a nearly constant velocity and makes only minor adjustments to its trajectory in order to keep its target at a fixed azimuth and elevation. The proportional navigation stage lasts for most of the engagement. The reason for employing proportional navigation is twofold. First of all, proportional navigation minimizes the need for trajectory changes (which increase drag) and thus conserves fuel and velocity. Secondly, much less propulsion is needed to change the approach vector significantly at long range than at close range, since the magnitude of the vector change is inversely proportional to the range. Any such adjustments should thus be made when the missile is as far from the aircraft as possible.

The aircraft's survival is typically determined during the proportional navigation stage. If the missile is successfully detected and tracked by a threat warning system, then the pilot can employ decoying operations in an attempt to mislead the missile. These countermeasures stand a much greater chance of success if they are coordinated by an accurate TTI estimate.

If the pilot is not successful in warding off the missile during its proportional navigation stage, whether due to a poor TTI estimate or simple bad luck, then the missile will enter its final approach. During this stage, the missile approaches the aircraft as closely as possible before detonating. During this stage, minor changes in position or velocity of either the missile or the aircraft result in drastic changes to the approach vector. If the engagement reaches this point, the aircraft will be lucky to survive.

## **2.5 The TWES**

We shall use the TWES to model the phenomena described in this chapter. The TWES is an integrated simulation package used to simulate all aspects of a missile-aircraft engagement. The TWES was originally developed at LIRIS over a decade ago and has been continually revised in order to improve performance and accommodate advances in the technology. One of the most useful features of the TWES is its user interface. Given only a handful of parameters, the TWES is able to model an engagement very successfully. These parameters are described below.

### **2.5.1 Source modeling**

LIRIS's Threat Warning Engagement Simulator (TWES) is able to model the radiation characteristics of a variety of sources. We briefly described source radiation in section 2.1; we also stated that we would model our source as a constant-temperature isotropic blackbody. To model a constant-temperature isotropic blackbody source, the TWES requires specification only of radiant intensity.

### **2.5.2 Sensor modeling**

The TWES provides users the capability to model different sensors. The TWES allows the user to specify the number and placement (in azimuth and elevation) of these sensors on the sensor platform. The TWES also requires specification of the sensor frame rate. The precise details of the sensor modeling are unimportant to this thesis.

### **2.5.3 Atmospheric and kinematic modeling**

Much of the utility of the TWES resides in its atmospheric modeling capabilities. The TWES atmospheric models, like many of the atmospheric models used in this field (especially in the defense industry) are derived from the Department of Defense's LOWTRAN code [13]. LOWTRAN is a software simulation package that models atmospheric phenomena. LOWTRAN also contains sets of atmospheric standards. These standards were established in 1976 and have been periodically updated ever since.

The TWES atmospheric models require specification of several parameters; these are described below in conjunction with the kinematic parameters, which describe how the engagement geometry evolves. The TWES is capable of simulating each stage of missile flight. We are primarily concerned with the proportional navigation stage of the engage-

ment. Thus we will describe only the TWES parameters for kinematic description of the proportional navigation stage of the engagement.

### **2.5.3.1 Altitude**

As our sensor platform altitude increases, we notice several things. First of all, the atmosphere becomes thinner. This means that photons emitted from a source encounter fewer and fewer atmospheric molecules en route to the sensor. Second, ground effects become more and more negligible. The TWES models the ground plane as a perfect absorber. Any photon that comes into contact with the ground plane disappears forever from the simulation. This TWES modeling assumption (of admittedly dubious validity) was motivated by the extreme complexity of modeling different kinds of terrain. This ground model is somewhat inaccurate at low altitudes but becomes more accurate at higher altitudes. In any case, as altitude increases, the atmosphere becomes more and more similar to a vacuum and thus atmospheric effects decrease.

### **2.5.3.2 Initial range**

The distance from the source to the sensor platform obviously affects atmospheric phenomenology. As range increases, a photon emitted from the source encounters more atmospheric molecules on its journey to the sensor platform. An increase in the probability of molecular encounters results in a higher probability of absorption or deflection. An increase in range thus results in more pronounced absorption and scattering.

In the TWES, initial range is the range at which we begin accumulating sensor data. That is, it is the source-sensor platform range at the first frame of sensor data. Initial range is denoted by the symbol  $r_0$ .

### **2.5.3.3 Closing velocity**

Missile velocity is nearly constant during the proportional navigation stage of missile flight. Aircraft velocity is also typically constant or nearly constant during the missile's proportional navigation. Closing velocity is defined as the relative velocity at which the missile approaches the aircraft. This velocity is constant as long as the missile and aircraft velocity are both constant. Closing velocity is different from the ground velocity of the missile; closing velocity for a missile at a fixed ground speed would be higher for a head-on shot than for a tail shot, for instance.

The range at the beginning of an engagement is specified by initial range  $r_0$ ; the time between sensor frames is defined by the sensor frame rate. Specification of the closing velocity thus defines the range at every subsequent frame of the engagement. Closing velocity is denoted by the symbol  $v$ .

#### 2.5.3.4 Azimuth

Azimuth refers to the horizontal angle between the body-centered sensor platform reference axes and the source approach vector. Azimuth can be envisioned as a longitude measurement of the source with respect to a level sensor platform (see figure 1.1). Recall that due to proportional navigation, azimuth varies little during an engagement. We shall use the symbol  $\psi$  to denote azimuth.

The TWES assumes that at a given altitude, the atmosphere is homogeneous. This is not strictly true; clouds and high- and low-pressure regions are examples of intra-altitude atmospheric inhomogeneities. On a scale commensurate with a typical missile-aircraft engagement, however, horizontal homogeneity is a reasonable assumption that simplifies simulations significantly. Thus, to the TWES, azimuth is unimportant. Azimuth will become important in other contexts, however.

#### 2.5.3.5 Elevation

Elevation refers to the vertical angle between the sensor platform reference axes and the source approach vector. Elevation can be visualized as a latitude measurement of a source with respect to a level sensor platform (again, see figure 1.1). Thus an elevation of 0 degrees refers to a source approaching from a point on the equator; an elevation of -90 degrees corresponds to a source approaching from directly below the platform. Elevation, like azimuth, varies little during an engagement. We shall use the symbol  $\phi$  to denote elevation.

Elevation, unlike azimuth, has an effect on atmospheric phenomenology. This is due to the inhomogeneity of the atmosphere in altitude. If we look down from our sensor platform, we definitely see a different atmosphere than the one we see if we look up from our platform. Photons emitted from a source at -90 degrees will encounter more atmospheric molecules than photons emitted from a source at +90 degrees due to the denser atmosphere at lower altitudes. The TWES takes this into account when modeling radiation propagation.

#### 2.5.3.6 Absorption

The  $k_a$  in equation 2.11 is a measure of atmospheric absorption. In a vacuum,  $k_a = 0$ ; in the earth's atmosphere,  $k_a$  is nonzero. The absorption coefficient  $k_a$  is not specified directly as a TWES input. Measurements of various constituent concentrations (primarily ozone concentration, since our system operates in the UV regime) are used to compute a value of  $k_a$  internally in the TWES.

#### 2.5.3.7 Scattering

Scattering is also specified to the TWES indirectly. Scattering levels are inferred from the same types of measurements used to calculate  $k_a$ . Two parameters are generally specified

to the TWES in place of  $k_s$ : visibility and atmospheric arena. Visibility and atmospheric arena are used as indices specifying an empirical multiple scattering model to be used by the TWES. Possible TWES atmospheric arenas include desert, rural, urban, maritime, and tundra.

## **2.6 Assumptions**

We will employ a number of simplifying assumptions to make the TTI estimation problem more tractable. These assumptions are based on judicious observations of the real world and are intended to be as unrestrictive as possible. We want an algorithm that is able to operate in widely varying circumstances and able to produce reliable TTI estimates in a wide range of conditions. Most of the following assumptions are imposed in order to fix a specific physical parameter (e.g., closing velocity or source radiant intensity) for the duration of an engagement. This greatly simplifies the development of our algorithms. These assumptions will be re-evaluated in chapter 5 of this thesis.

### **2.6.1 The sensor platform is in level flight at a constant velocity**

Aircraft spend much of their time in nearly level flight at a nearly constant velocity. Assuming that our sensor platform remains at the same altitude for the duration of the engagement frees us from the inhomogeneity concerns imposed by a layered atmosphere. The assumption of constant velocity is reasonable because aircraft, like missiles, typically will not waste fuel zigging and zagging when a threat is still quite far away.

### **2.6.2 Atmospheric conditions remain constant throughout an engagement**

The atmosphere changes with altitude. As noted in section 2.3, the atmosphere also changes horizontally at any given altitude. However, the distance covered by an aircraft during a typical engagement is small compared to the scale of horizontal atmospheric inhomogeneities. Thus we can assume that atmospheric conditions remain constant during an engagement.

### **2.6.3 The source approaches along a straight line at a constant velocity**

For most of the duration of a typical engagement, a missile is in the proportional navigation stage of its flight. During the proportional navigation stage, the missile's velocity and attitude remain nearly constant; the missile makes minor corrections to its trajectory to compensate for minor deviations and losses. Coupled with our first assumption, this means that closing velocity and AOA are constant throughout an engagement.

### **2.6.4 The source AOA is [0,0]**

This assumption is completely artificial and has no real-world motivation. In the real world, a missile could approach from any direction. Limiting elevation to zero (in conjunction with the atmospheric assumptions described above) constrains missile altitude to



remain constant throughout an engagement, thus ensuring that the missile-aircraft environment is homogenous throughout the engagement. Further constraining azimuth to zero removes concerns of non-uniform response over the sensor face. Practical size and complexity issues discussed in chapter 3 forced this assumption's institution. Chapter 5 contains a discussion about how well this assumption can be reconciled with real-world concerns, and what happens when this assumption is violated.

### **2.6.5 The source is an isotropic blackbody at constant temperature**

Section 2.1 contains a description of source radiation characteristics. In reality, a missile is not an isotropic constant-temperature blackbody. These assumptions enable us to ignore concerns of spectral radiance variations within an engagement by fixing radiation and thus allowing us to describe a missile completely in terms of its radiant intensity  $J$ .

The surface area and temperature used in the atmospheric simulations were  $1 \text{ m}^2$  and  $3000 \text{ K}$ , respectively. These values result in a source that is several orders of magnitude brighter than any missile we might expect to encounter. The radiant intensity of this source is approximately  $40 \text{ W/sr}$ , while typical missile radiant intensities range roughly from  $5 \text{ mW/sr}$  to  $500 \text{ mW/sr}$ . The noiseless sensor outputs produced by this  $40 \text{ W/sr}$  source were thus scaled down to mimic those produced by a missile of a desired radiant intensity. Although this scaling was done externally to the TWES, it is equivalent to reducing the surface area of the blackbody: the spectral dependence of the source radiation emission retains its shape, but decreases in magnitude by a constant factor.

Overall, the assumption that our source is a constant-temperature isotropic blackbody is of dubious validity. Although a missile generally remains at a near-constant temperature during proportional navigation, it is neither isotropic nor a blackbody. However, this assumption is necessary to our analysis, since considering the effects of different source geometries and radiation patterns is beyond the scope of this thesis.

## **2.7 Past Work**

Numerous attempts have been made to develop and implement a successful TTI estimation algorithm. Several attempts have been made at LIRIS within the past few years [14-16]. These attempts have met with varying degrees of success. In general, TTI algorithms produced in the past have provided TTI estimates with low mean errors but with significant root-mean-squared (rms) errors. Typically the rms error has been sufficiently high to preclude use of the algorithm as a reliable source of a TTI estimate. Thus many systems make no TTI estimate at all.

A discussion of three relevant passive-sensor TTI estimation attempts follows. All three of these efforts have taken place at LIRIS in the last several years.

### 2.7.1 A curve-fitting approach

The possibility of using a curve-fit to determine TTI was mentioned in previous sections. It was noted that the addition of the scattering term to an otherwise simple analytical formula greatly complicates the curve-fitting. If there is no analytical form for the sensor output growth, then there is no function to which to fit data.

In 1987, a team of LIRIS engineers working with a single-pixel sensor system designed a TTI estimation algorithm based on curve-fitting. This attempt was based on the assumption that the total incident radiation growth over the FOV of an entire sensor is similar to the direct radiation growth. Essentially, this team decided to neglect the scattering and model the sensor output as in equation 2.11, i.e., as it would be in a non-scattering atmosphere.

The discouraging results of this attempt are outlined in a LIRIS technical document [14]. Mean error is near zero, but rms error in most cases is nearly as high as TTI itself. Disregarding the scattering terms  $\Theta_i(t)$  in equation 2.12 is catastrophic. Scattered radiation across the FOV of an entire sensor typically outweighs direct radiation unless the source is relatively near the sensor platform. The scattered radiation growth across the FOV of a single-pixel sensor also dominates the total radiation growth, in essence indicating that observed data from a single-pixel sensor does not behave according to equation 2.11 except at close ranges.

### 2.7.2 An extended Kalman filtering approach

Mark Luetzgen, a LIRIS VI-A student, dealt with the problem of TTI estimation in his 1990 Master's thesis [15]. Luetzgen, like the team before him, worked with a system consisting of single-pixel sensors. Luetzgen took a different approach to the problem. He used an extended Kalman filter (EKF). The EKF takes initial estimates of relevant physical parameters (including TTI) and a matrix-valued physical parameter error covariance estimate  $C$  as "seeds" for an iterative estimation process [17]. The EKF uses a state-space atmospheric propagation model to estimate TTI and  $C$  at every new data frame. With a good initial guess and well-behaved data, both the TTI estimate and the elements of  $C$  should approach bounded steady-state values; the latter estimates the error covariance of TTI and other state-space physical parameters.

Luetzgen's results were more promising than those seen several years earlier. His estimates had a lower rms error and near zero mean error. However, his approach had other problems and limitations. Local minima in parameter space often cause convergence to a false TTI. Additionally, there is no guarantee that the EKF algorithm will converge at all. Convergence is dependent on well-behaved data and good seeds. The TTI seed can be based on a sensor measurement, but there is no easy way to guarantee a sufficient level of accuracy in the initial estimate of  $C$  to ensure accurate convergence of the EKF. Because the error covariance matrix  $C$  is used to calculate the gain terms at each step in the EKF process, the accuracy and reliability of the algorithm are affected greatly by the initial esti-

mate [18]. Thus an algorithm run that might converge with a small initial error covariance matrix estimate might not converge with a larger covariance matrix.

### **2.7.3 A measurement-matching approach**

Craig Andera made the most recent approach to the TTI problem at LIRIS, while working on his 1995 M.Eng. thesis [16]. Andera's system is virtually identical to those already mentioned; it comprises six single-pixel sensors. Andera's approach is different from those taken before him. He creates a master table of sensor outputs based on a subsampling of a wide span of all physical parameters. Given a set of observed sensor outputs during an algorithm run, Andera checks his table to see which entry produces the minimum mean-squared deviation from the sensor data. His TTI estimate is calculated from the range and velocity corresponding to this entry's physical parameter values.

Andera's algorithm successfully estimates AOA, which was a significant problem for single-pixel sensors. However, the algorithm falls short of the mark when it comes to TTI. His results again show near-zero mean error but large rms error. His rms errors are approximately equal to Luetzgen's for similar atmospheric and source characteristics.

One important difference between Andera's algorithm and Luetzgen's algorithm is that Andera's is guaranteed to produce an estimate, whereas Luetzgen's algorithm may not stabilize to a fixed estimate. Another important difference is that due to the extensive size of Andera's master table and the related search effort inherent in finding the best estimate, Andera's algorithm is more computationally intensive than Luetzgen's. Luetzgen reports an execution time of 2 seconds for his algorithm when it is run on a DEC VAXstation; Andera reports an execution time of 20 seconds on a significantly faster DEC Alpha.

## **2.8 New Conditions**

All three approaches to the TTI problem described above suffered from large rms error. It appears that in general there is not enough information present in a single-pixel sensor system to do much better than in the three cases listed above. The total scattered radiation received by single-pixel sensors overwhelms direct radiation at all but close ranges. Most of the information about TTI we can hope to glean from the scattered radiation is inherent in its distribution across the sensor face, not in its overall magnitude. However, a single-pixel sensor does not have the resolution to see the shape of the scattering function; it sees only an integration of the scattering function over its entire FOV. Trying to guess TTI from an integration this coarse is similar to trying to describe a person's physical appearance given only his weight. Any inferences about the shape of the scattering function in a single-pixel sensor system have to be made from six very coarse integrations of that function. Since most of the variation of the scattering function occurs at small angles from AOA, whereas the large-angle behavior contains relatively little information, the scattering information most relevant to TTI is lost in a single-pixel sensor system.

Our system has one incredibly important difference from those used in the past. Our sensors are imaging sensors. We have sufficient resolution to see a much finer spatial integra-

tion of the scattered radiation. We thus have the small-angle, or “near-field,” scattering data available to us. This provides two viable options, both of which utilize the near-field scattering data. The first algorithm, the parameter-matching algorithm, uses the shape of the sensor output profile to divine information about TTI. The second algorithm, the curve-fitting algorithm, looks to a very narrow region of the near-field as an approximation of the direct radiation received by the sensor. We shall discuss the parameter-matching algorithm first.

### 3.0 THE PARAMETER-MATCHING ALGORITHM

This chapter provides a description and performance assessment for the first of our two algorithms, the parameter-matching algorithm. The parameter-matching algorithm is somewhat similar in concept to Andera's 1995 algorithm, i.e., it compares sensor data to entries in a large lookup table and returns a TTI estimate corresponding to the minimum-error entry. Let us define a noisy sensor output vector  $\mathbf{z}$  as a six-dimensional vector whose elements are statistically independent and Poisson distributed with mean values given by the corresponding values of  $\mathbf{h}$ :

$$\bar{\mathbf{z}} = \mathbf{h} = \begin{bmatrix} \frac{I_0 e^{-kr(t)}}{r^2(t)} + \Theta_1(t) \\ \Theta_2(t) \\ \Theta_3(t) \\ \Theta_4(t) \\ \Theta_5(t) \\ \Theta_6(t) \end{bmatrix} \quad \text{Equation 3.1}$$

Note that equation 3.1, like the others preceding it, assumes a set of six single-pixel sensors for notational simplicity. For an imaging sensor,  $\mathbf{h}$  and  $\mathbf{z}$  contain an element for each pixel.

Andera's master table contains the  $\mathbf{h}$ 's obtained at different points in the physical parameter space. We can define  $\mathbf{h}$  directly in terms of these physical parameters:

$$\mathbf{h} = F(r_0, \nu, I_0, k, \Theta, \psi, \phi) \quad \text{Equation 3.2}$$

If the table of these  $\mathbf{h}$  values is indexed by the physical components that are arguments of  $\mathbf{h}$ , then it is simple to recover these physical parameters given the location of a specific  $\mathbf{h}$  in the master table.

Andera's algorithm uses several frames of sensor data to build its sensor output vector  $\mathbf{z}$ . He uses data from all six single-pixel sensors. Our intention is not to use every pixel value from every sensor in every frame; we would like to concentrate on the small-angle scattering over all available frames, as our hypothesis is that this will provide relevant information about TTI.

The parameter-matching algorithm development will progress in several steps. First, we shall develop a method to provide a useful, but small-dimensional, description of the full sensor output image. Second, we will construct a master table that spans a wide range of physical parameters. Third, we shall select a cost function that can be used to relate this sensor data to the noiseless entries in the master table. Finally, we will provide a method for efficiently and quickly searching the master table to find the best match to the data.

## 3.1 Sensor Output Parametrization

Andera's single-pixel system produces six data points per sensor frame: one value for each single-pixel sensor. The imaging sensors in our system provide us with many more than six data points per frame. We can imagine developing an algorithm that utilizes the full set of data provided by multiple image frames. However, it is in our computational interest to find a way to reduce this large data set into a compact, but still meaningful, form.

### 3.1.1 The abundance of information

A typical imaging sensor used on-board a military aircraft contains anywhere from a hundred to ten thousand or more pixels on its face [3]. A typical frame integration rate might range from ten to one hundred frames per second [3]. A six-sensor platform using high-resolution imaging sensors and a high frame rate could, by this reckoning, amass over six million data points in a single second! Essentially this implies that if we use every data value, then our sensor output vector  $z$  contains not six elements, but as many as six million. Clearly any algorithm that attempts to use every pixel value at every frame in such a system would be quite computationally expensive (both in terms of storage space and computation time). Hence, we see that too much information is not necessarily a good thing.

To streamline a TTI algorithm, we need to reduce the complete set of pixel values obtained during observation to a much smaller set of parameters. We would like this reduced parameter set to have two characteristics. First, it should be as small as possible. The fewer elements in the set, the quicker and more efficient the algorithm will be. Second, and just as important, we would like the small, reduced set of parameters to preserve most of the information in the original, unreduced set. It is not productive to disregard information that is useful for estimating TTI, even if the result is a more "efficient" algorithm. A streamlined, efficient algorithm that produces a consistently poor TTI estimate is of no use.

It is in fact possible to reduce an entire frame of data to a scant handful of parameters and still preserve a good deal of the information in the full set of pixel values. We will achieve this by spatially integrating pixel outputs.

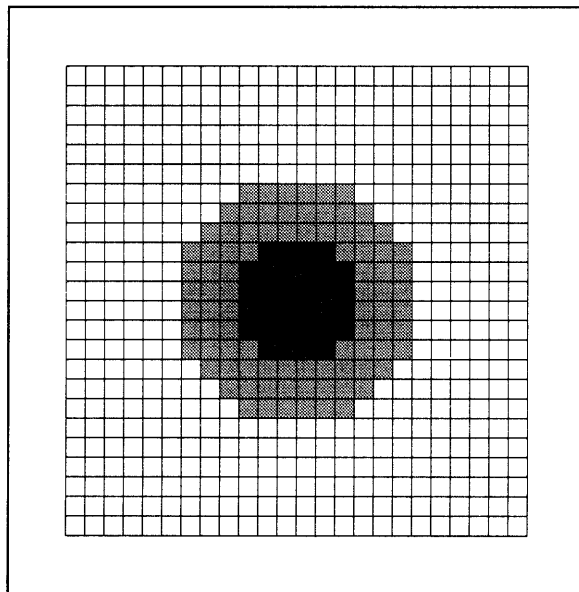
### 3.1.2 Pixel integration rings

Atmospheric scattering blurs a point source into a diffuse glow [4]. This makes TTI estimation complicated, but not hopeless. If we can find a way to relate scattering to TTI, then we can use scattered radiation data to obtain an estimate of TTI. In this section, we will demonstrate that with an accurate AOA estimate (which is easy to obtain with our high resolution sensors) we can characterize the scattered radiation received at the sensor platform by looking only at the output of the line-of-sight sensor.

We made several generalizations about scattering in section 2.3.2. Recall that a distant source will appear more blurry and diffuse than a nearby source (see figure 2.3). Addi-

tionally, due to forward-peaked scattering, most of the scattered radiation received at our sensor platform is contained at small angles around AOA. Finally, in a homogenous atmosphere, the scattered radiation received by the sensor is radially symmetric about AOA. These properties suggest a simple approach for reducing the data set. To preserve the most information, we want to concentrate on the areas where scattering is greatest—the pixels in the vicinity of the source AOA.

In a homogenous atmosphere, scattering falls off symmetrically in every direction. Thus all pixels at, say, a ten-degree separation from AOA will have roughly the same value in a noiseless system. These pixels thus provide roughly the same information. We can imagine integrating the pixel outputs in annular regions surrounding AOA. For instance, we could add up the pixel outputs in each of the gray-tone regions in figure 3.1 to reduce the dimension of the output vector to 3. This procedure is straightforward as long as we have an AOA estimate to use as the origin for our radial integration.



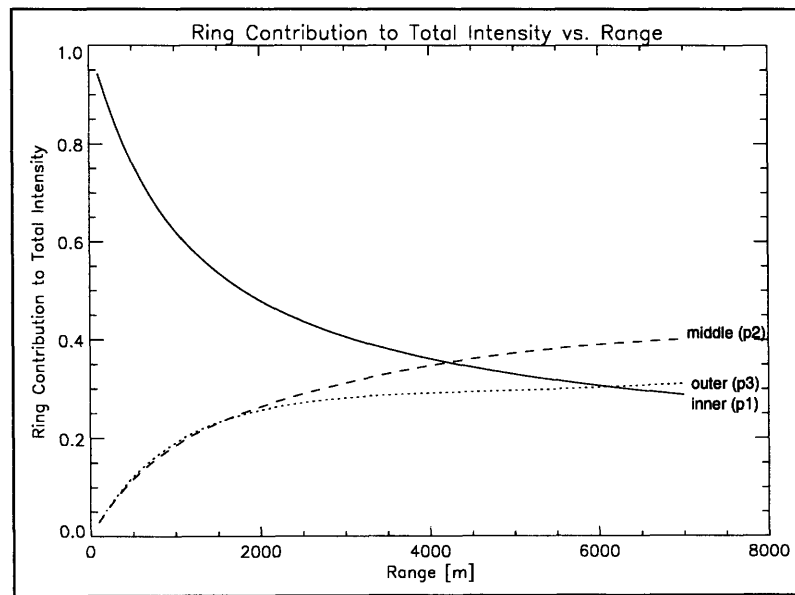
**FIGURE 3.1: Pixel integration rings**

This parameter set reduction approach has several advantageous features. First, it is a fairly intuitive way to describe the sensor output. Second, it is computationally simple, requiring only addition operations equal in number to the amount of integrated pixels. Finally, integration of pixels will generally improve the SNR of the data set.

The choice of the number and size of the rings is subject to a number of practical considerations and limitations. Ring radii should be as small as possible to maximize resolution, but several other factors must be taken into account. First of all, it is desirable to have the integrated ring intensities be tolerant to subpixel AOA deviations, since our system generally will not have available an AOA estimate below the pixel level. Because the granularity of the radial integration is a single pixel, it is not directly possible to shift the integration origin for different AOAs within a single pixel. This requires an interpolation

or dithering procedure. In order to avoid this altogether, we want different AOAs within a single pixel to result in similar integrated values for each ring. This criterion sets a lower bound on the center integration radius; testing shows that this bound is approximately 2.5 degrees. Thus the outputs of all pixels with centers within 2.5 degrees of AOA are summed to form the inner ring parameter.

Having chosen a size for the inner ring, we face other constraints for choosing the other ring radii. It is desirable for all rings to have similar SNR characteristics, so that whatever cost function is chosen is equally sensitive to noise variations in all rings. Thus we should choose each successive ring radius to give an integration value that is similar to that of the center ring in a typical atmosphere, at least at ranges that are far enough away that the inner ring intensity does not completely dominate the others. Though based on TWES simulations, this choice is somewhat subjective; we choose a parameterization into three rings with radii of 2.5 degrees, 14 degrees and 45 degrees. Integrating past 45 degrees is of little value because in most cases scattering at this angle is negligible. Figure 3.2 illustrates the relative output contribution from each ring, averaged over 27 representative atmospheres. (These atmospheres will be discussed in section 3.2.)



**FIGURE 3.2: Relative ring contribution to total sensor output**

Let us briefly examine the phenomenology underlying figure 3.2. The inner ring, because its integration includes AOA, will capture all of the direct radiation received from the source. At close ranges, the contribution from the direct radiation will be large and we will observe relatively little scattered radiation. Most of the radiation that is received at the sensor has arrived along the line-of-sight, and not along a scattered path. As the path length increases, however, relatively fewer photons reach the sensor along the direct path. Not only does a more distant source result in a higher probability of scattering, but more



radiation along the direct path is absorbed before it reaches the sensor platform. Thus scattering becomes a more important contributor to total radiation.

Note that the inner ring integration does not consist entirely of direct radiation. Some scattered radiation is collected in this region. The scattering distribution is forward-peaked, so most scattered radiation is deflected only slightly off its original course. Larger deflections are, by comparison, much rarer, and a large enough deflection will prevent a photon from reaching the sensor platform at all unless it is rescattered. Most of the scattered radiation observed in the inner ring consists of photons that have been deflected only once en route to the sensor platform. Conversely, many of the photons received in the outer rings may have been scattered multiple times. As we move further from AOA, the probability that an observed photon has been scattered only once decreases.

The possibility of multiple scattering increases with range. A distant source will cause more multiple-scattered radiation to be observed at the sensor platform. Thus as range increases, the relative contributions of the outer rings will also increase. This is observed in figure 3.2.

Forming rings of intensity enables us to reduce a data set of thousands or millions of data points per frame to a parameter set containing only three points. Sensor outputs can thus be reduced by simple addition into a much smaller set that is nearly as informative.

### 3.1.3 Duration of data collection

The choice of data collection duration, or the number of sensor frames to collect, presents a trade-off. We would like our algorithm to collect data for as long as possible; the more data frames we have, the more information should be present to predict TTI. However, while the algorithm is priming itself by collecting data, the source is getting closer and closer; the time in which the algorithm has to produce an estimate and the pilot has to react are getting shorter and shorter. Since time is quite a premium here, there needs to be a point when the algorithm stops collecting data and begins processing it to formulate an estimate.

We set a duration of 2 seconds for data collection. This choice is somewhat arbitrary but is motivated in part by Andera's choice in [16]. A standard frame integration rate is 100 Hz; in the interest of reducing the size of our data set, we choose a slower frame rate of 10 Hz. This gives us twenty frames of data. Twenty frames of three values (ring intensities) each gives us 60 data points for an algorithm run. This is a reduction by a factor of thousands from the unreduced data set. We will continue to use the notation  $\mathbf{h}$  to denote a parametrized noiseless output vector and  $\mathbf{z}$  for a parametrized observed sensor output vector of the three-ring, twenty-frame reduced data set, i.e.,  $\mathbf{z}$  and  $\mathbf{h}$  will be sixty-dimensional vectors.

## 3.2 The Canonical Set

Implementation of a TTI estimation algorithm requires knowledge—whether analytical, empirical, or modeled—of how TTI influences the sensor outputs. If an analytical relationship between TTI and sensor output existed, a complex algorithm would be unnecessary. However, no such analytical relationship is available, so we must rely on the empirically determined models of the TWES to relate sensor data back to TTI.

We would like some kind of a mapping relating points in sensor output space to points in physical parameter space. All the maps we have are unidirectional; given a point in physical parameter space, these maps can calculate the location of the corresponding noiseless point in sensor output space. The complexity of our models and the presence of noise prevent us from explicitly mapping in the other direction. We can, however, construct a bidirectional map by tabulating the results of TWES runs. By providing a set of physical parameters completely spanning the pertinent region of physical parameter space, we can obtain sensor outputs completely spanning the reachable sensor output space. These sensor outputs can be reduced into rings of intensity exactly as described in section 3.1.2. This process produces a map from the reduced data space to physical parameter space. We shall call this map the canonical set.

The resolution of the canonical set can be made as fine as desired; the sensor output space resolution is determined by the physical parameter space resolution. As the resolution of the physical parameter space approaches continuity, the resolution of the reachable sensor output space will approach continuity [19].

### 3.2.1 Atmospheric resolution within the canonical set

As described in section 2.5.3, the TWES contains atmospheric models based on the LOWTRAN atmospheric models. These models are governed internally by a multitude of parameters which describe the concentration, particulate size, and behavior of numerous atmospheric constituents and aerosols. Specification of all these parameters is generally not done explicitly. Rather, the user typically specifies three external parameters which are used to calculate the internal model parameters: ozone concentration, visibility, and atmospheric arena. Visibility and atmospheric arena specifications are used to infer the concentrations of numerous atmospheric and aerosol concentrations, based on empirical models. Because our system operates in the UV, and is especially sensitive to the presence of ozone, we specify this concentration explicitly. Visibility is measured in the visible range of the spectrum, and cannot be used to infer ozone concentration as accurately as our system demands. Ozone concentration is the primary determinant of the absorption coefficient  $k_a$  in the UV regime.

In order to explore a wide range of possible atmospheres, three values were chosen for each of the three external atmospheric parameters. Table 3.1 lists the permutations of parameters chosen for each atmosphere. Low, nominal, and high values of ozone concentration were chosen. Likewise, low, nominal and high visibilities were chosen. Three atmospheric arena choices, designated “rural,” “maritime,” and “desert,” were chosen to

provide a wide span of possible atmospheric settings. The permutations of these three parameters provide a set of 27 different atmospheres. This set of 27 atmospheres spans the realm of typical atmospheres in which the algorithm would operate.

This choice of 27 different atmospheres stands in stark contrast to the atmospheres used in the past works cited in section 2.7. Luetgen used two atmospheric models; he was limited by the fact that he had to develop these models himself. (As an aside, it should be noted that he was quite successful in developing these models; he published his results in [20].) Andera used only two TWES atmospheres. Our algorithm, because of the much broader range of atmospheric conditions that it includes, should be more robust than these prior works.

**TABLE 3.1: Canonical set atmospheres**

<b>ATM</b>	<b>ozone</b>	<b>visibility</b>	<b>arena</b>	<b>ATM</b>	<b>ozone</b>	<b>visibility</b>	<b>arena</b>
1	low	low	rural	15	high	nominal	desert
2	nominal	low	rural	16	low	high	desert
3	high	low	rural	17	nominal	high	desert
4	low	nominal	rural	18	high	high	desert
5	nominal	nominal	rural	19	low	low	maritime
6	high	nominal	rural	20	nominal	low	maritime
7	low	high	rural	21	high	low	maritime
8	nominal	high	rural	22	low	nominal	maritime
9	high	high	rural	23	nominal	nominal	maritime
10	low	low	desert	24	high	nominal	maritime
11	nominal	low	desert	25	low	high	maritime
12	high	low	desert	26	nominal	high	maritime
13	low	nominal	desert	27	high	high	maritime
14	nominal	nominal	desert				

### 3.2.2 Spatial resolution within the canonical set

The source’s approach is defined by its initial range and closing velocity. The canonical set should span all possible ranges and velocities at a very fine resolution, since TTI is simply range divided by velocity.

By convention, initial range in our algorithm shall refer to the range at the time when the algorithm has finished data collection, i.e. the range at the final data frame. (Note that this convention is different from the TWES range convention described in section 2.5.3.2.) Thus, when we speak of a TTI estimate, we shall also be referring to the time-to-intercept from the instant when the last frame was collected.

The brightest missiles are trackable only out to a range of several kilometers [1]. Coincidentally, this is also approximately the maximum range at which a missile would be launched at an aircraft. At ranges beyond about 5000 m, the probability of successful

engagement by passive infrared heat-seeking missiles is very low. Most engagements involving this type of missile are thus initiated well within the 5000 m range. We choose the ranges between 100 m and 4500 m as the span of the canonical set ranges; we will use a resolution of 25 m between points. This gives the canonical set 177 range points.

When in level flight, aircraft on which our system might be used typically travel at 50 to 250 m/s. Typical missile velocities range from around 300 to 800 m/s. This implies that the maximum closing velocity (for a head-on shot) would be 1050 m/s, and minimum closing velocity (for a tail shot) would be 50 m/s. The canonical set spans velocities from 200 m/s to 1100 m/s. (A missile closing at a low velocity has a low probability of reaching its target, so velocities below 200 m/s are not included.) The resolution on closing velocity is 25 m/s. This results in 37 velocity points in the canonical set.

Typical missile radiant intensities are in the range of 5 mW/sr to 500 mW/sr. However, it is possible to construct the canonical set with a single intensity value. The precise reasons for this are discussed in section 3.3; they are motivated by the fact that noiseless sensor outputs scale linearly with source radiant intensity.

Given the atmospheric and spatial resolution above, the canonical set already has  $27 \times 37 \times 177 = 176823$  points. Each one of these “points” actually represents a 60-dimensional reduced sensor output vector. Thus, even with data reduction, the canonical set quickly becomes unwieldy. Storage requirements and set traversal time both increase with canonical set size.

The inclusion of azimuth and elevation in the canonical set would add two more high-resolution physical parameter dimensions. This would result in a multiplication by a factor of hundreds or thousands to the size of the already large canonical set. This is not practical, from either a storage or a search standpoint.

This limit motivated the  $[\psi, \phi] = [0, 0]$  assumption. This is admittedly a fairly limiting case, though this very assumption has been made by everybody mentioned in section 2.7 with the exception of Andera. We can imagine obtaining an AOA estimate and using this estimate as the origin for our pixel integration; however, as elevation deviates from zero the scattering profile becomes less and less symmetrical. This is due both to atmospheric altitude inhomogeneity and to different optical response at different points on the sensor face. To represent these effects explicitly we would need to include nonzero AOA in the canonical set.

The necessity of imposing the  $[\psi, \phi] = [0, 0]$  condition, in order to construct a usable canonical set, is a serious practical limitation. Its implications are discussed further in chapter 5.

### 3.3 The Cost Function

A cost function provides a criterion for judging which entry in the canonical set is optimal given a noisy sensor output vector. The cost function should be intuitively and computationally simple, and should reflect some perception of what it means to say that two sensor

data vectors are “similar.” The performance of the algorithm is dependent on our choice of cost function, since any number of different TTI estimates could be considered “optimal” under different cost functions.

Probably the most common choice of cost function in all estimation algorithms is mean squared error (MSE). A MSE cost function is usually tractable and assigns high penalties to outlying data points. The MSE cost function is widely used—in part for the reasons just stated, but also because in many cases MSE minimization coincides with maximum likelihood (ML) or maximum a posteriori (MAP) estimation [21]. Specifically, in the presence of additive Gaussian noise, MSE minimization algorithms have numerous characteristics that make MSE a natural choice for cost function.

Sensor noise is not Gaussian; it is Poisson. However, the Gaussian and Poisson distributions are nearly identical for large means. Even at very low means (around 20 or 30 counts) the Poisson distribution is almost identical to the (discretized) Gaussian distribution [22]. Thus it appears that MSE might be a natural choice for a cost function.

Let us define  $S$  to be the squared error between a parametrized observed sensor output vector  $z$  and a given canonical set entry  $h$ . Thus:

$$S = \sum_{i=1}^{60} (z[i] - h[i])^2 \quad \text{Equation 3.3}$$

where  $z[i]$  and  $h[i]$  denote the  $i$ th elements of  $z$  and  $h$ , respectively. Equation 3.3 can be interpreted as a distance in 60 dimensions (see figure 3.3 for a two-dimensional representation). Thus squared error minimization over the entire canonical set is equivalent to choosing the canonical set entry  $h$  that minimizes  $S$  in equation 3.3, or choosing the minimum-distance  $h$  from  $z$ .

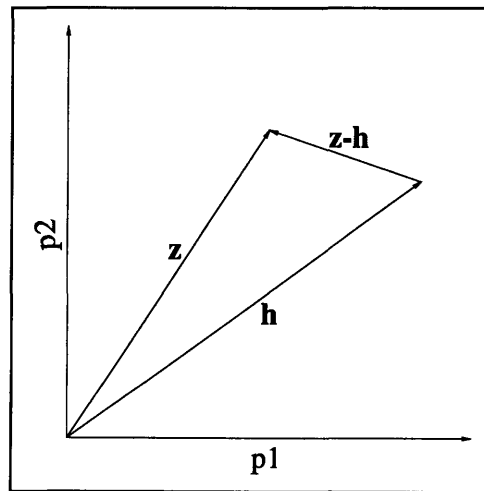
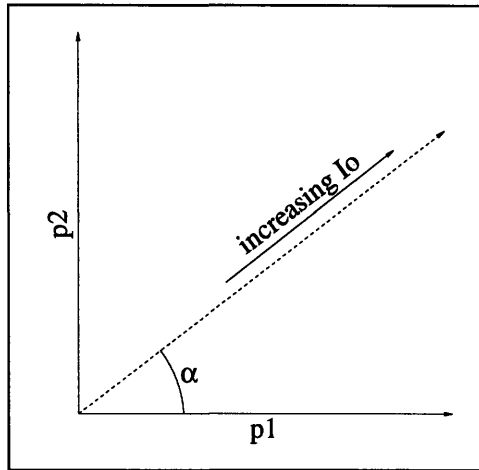


FIGURE 3.3: Distance between vectors in two dimensions

Recall that the canonical set does not include a source radiant intensity dimension. This is because the noiseless sensor output scales linearly with source radiant intensity. Scaling  $N$  in equation 2.8 results in an equivalent scaling of  $\mu$ . In a noiseless environment, a doubling of source intensity will result in a doubling of the output of every pixel. If we express  $\mathbf{h}$  as a function of  $I_0$ , we may write:

$$\mathbf{h}(\beta I_0) = \beta \mathbf{h}(I_0) \quad \text{Equation 3.4}$$

Source radiant intensity is, by assumption, constant throughout an engagement. Consider a family of output vectors created by holding all physical parameters constant except for source radiant intensity (or equivalently,  $I_0$ ), which is allowed to vary from zero to infinity. Plotting this family of points in two dimensional space produces the dotted line in figure 3.4. This family is a ray extending at a fixed angle  $\alpha$  from the origin. This generalizes to any number of dimensions. A family of radiant intensity curves in sixty-dimensional sensor output space is a ray extending from the origin in sixty-dimensional space.



**FIGURE 3.4: Family of source radiant intensities**

We can represent the ray in figure 3.4 with a single point, since the origin and a single point define the angle  $\alpha$ . More generally, we can define a ray in  $n$ -dimensional space (and its corresponding set of angles  $\alpha_1, \alpha_2, \dots, \alpha_{n-1}$ ) with a single point. We can thus represent every entry in the canonical set as a single point defining a ray in sixty-dimensional space. Each entry in the canonical set defines a different point in this space, and thus each one of these rays extends from the origin at a different angle. This allows us to construct a canonical set without a source radiant intensity dimension.

We are now interested in finding the best possible canonical set entry  $\mathbf{h}$  and  $I_0$  value to match an observed sensor output vector  $\mathbf{z}$ . This is a slightly different scenario from that depicted in figure 3.3, in which  $I_0$  is an explicit canonical set parameter. Instead of minimizing the distance between discrete points as in equation 3.3 and figure 3.3, we want to

minimize the distance between a point and a line, as depicted in figure 3.5. The canonical set entry defines the angle of the line by specification of all physical parameters except  $I_0$ ; the precise point along this line where distance is minimized corresponds to a specific value of  $I_0$ . Note that for any canonical set entry  $\mathbf{h}$ , the distance between  $\mathbf{z}$  and  $\mathbf{h}$  is minimized at the projection of  $\mathbf{z}$  onto  $\mathbf{h}$ . Note also that as  $\omega$  in figure 3.5 decreases, the distance between  $\mathbf{z}$  and  $\mathbf{h}$  decreases.

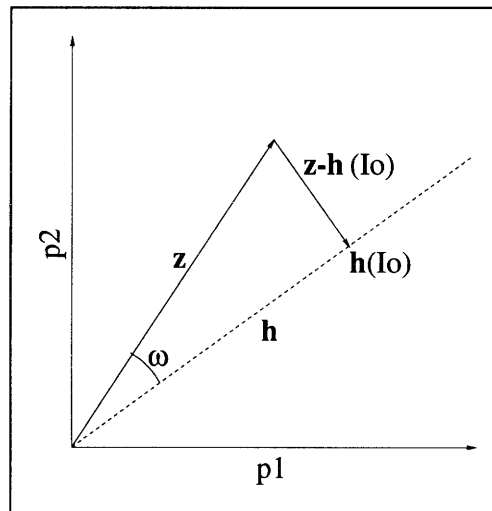


FIGURE 3.5: Distance between vector and ray in two dimensions

It is clear that the minimization of spatial distance between a vector  $\mathbf{z}$  and all canonical set entries is equivalent to the minimization of angular deviation between the sensor output vector  $\mathbf{z}$  and an entry in the canonical set. In figure 3.5, this angular deviation is denoted by  $\omega$ . This  $\omega$  is, in fact, very easy to calculate. For an observed output  $\mathbf{z}$  and a canonical table entry  $\mathbf{h}$ :

$$\mathbf{h} \cdot \mathbf{z} = |\mathbf{h}||\mathbf{z}| \cos \omega \quad \text{Equation 3.5}$$

Solving for  $\omega$ , we obtain:

$$\omega = \text{acos} \left( \frac{\mathbf{h} \cdot \mathbf{z}}{|\mathbf{h}||\mathbf{z}|} \right) \quad \text{Equation 3.6}$$

Minimizing  $\omega$  is thus reduced to minimizing the above expression.

All elements of every sensor output vector are non-negative. This implies that the maximum value  $\omega$  may obtain for any values of  $\mathbf{h}$  and  $\mathbf{z}$  is 90 degrees. Obviously, the minimum value for  $\omega$  is 0 degrees. On the range  $[0,90]$ , the *acos* function is strictly decreasing; thus minimizing  $\omega$  is equivalent to maximizing the quantity:

$$\omega' = \frac{\mathbf{h} \cdot \mathbf{z}}{|\mathbf{h}||\mathbf{z}|} \quad \text{Equation 3.7}$$

Furthermore, every entry in the canonical set can be normalized ahead of time to have the same magnitude, since scaling the magnitude of any point in sensor space is equivalent to changing the value of  $I_0$  in physical parameter space. Then, for any given  $z$ , maximizing the quantity in equation 3.7 becomes equivalent to maximizing:

$$\omega'' = \mathbf{h} \bullet \mathbf{z} = \sum_{i=1}^n \mathbf{h}[i] z[i] \quad \text{Equation 3.8}$$

since  $|\mathbf{h}|$  is constant for all canonical set entries, and  $|\mathbf{z}|$  is fixed for a specific  $z$ . Minimizing squared error is thus exactly equivalent to maximizing the dot product of  $\mathbf{h}$  and  $z$ . The vectors  $\mathbf{h}$  and  $z$  are sixty-dimensional; thus each comparison between the observed sensor output and a canonical table entry requires only 60 multiplications and 60 additions. The collapse of the dimensionality of the canonical set with the elimination of  $I_0$  also makes the search process quicker.

### 3.4 The Search Method

The cost function and canonical set have both been selected. The final requirement for completion of the algorithm is a search procedure. One choice of search method is a simple sequential search, in which the observed sensor output vector is compared to every canonical set entry in order. The entry that produces the minimum error after every entry has been checked would be used to produce a TTI estimate.

If time were not a concern, the “sequential search” procedure described above would be adequate and complete. Traversing every single entry in the canonical set, however, is extremely time-consuming. Recall that there are 176,823 entries in the canonical set. Calculating the error for each entry requires 60 multiplications and 60 additions. Thus traversing the entire set requires 21,218,760 calculations.

It is beneficial to reduce the number of calculations as much as possible; this will cut down on the amount of time it takes to produce a TTI estimate. Since the error computation has already been stripped down as much as possible, any additional computational savings must be achieved by modifying the search method.

#### 3.4.1 A coarse-to-fine search

One way to cut down on the search computation is to look only at a fraction of the entries in the canonical set. At first, it would appear that any algorithm that did not examine every canonical set entry would be flawed. It seems that such an algorithm could not guarantee that one of the entries which went unexamined did not have a lower error than the “best” examined entry. If the coarse search is conducted carefully and the error function is well-behaved, however, then we can be reasonably certain that the best estimate from the coarse search is in fact very close to the global best estimate [23].



The canonical set has three dimensions: range, closing velocity, and atmosphere. Each of these dimensions is discretized and limited to a finite span. We can view the canonical set entries as a function in these three dimensions. If we sweep along one dimension with the other two fixed, we see all 60 elements of  $\mathbf{h}$  evolving incrementally. Incremental changes in any physical parameter cause incremental changes in the sensor output. Thus adjacent entries (in range, velocity, or atmosphere) in the canonical set produce similar errors when compared to any observed sensor vector.

Let us modify our search so that instead of looking at every point in the canonical set, we look at every  $n$ th point in range, every  $n$ th point in velocity, and every  $n$ th atmosphere. (Generally speaking, we are not required to use the same coarseness in all dimensions; the range coarseness, for instance, could be greater than the velocity coarseness.) We obtain a “best” estimate from this coarse search. If  $n$  is reasonably small compared to the scale of local variations in  $\mathbf{h}$ , then we can be reasonably certain that the global best estimate is within  $n$  entries in all dimensions (range, velocity and atmosphere) of the coarse best estimate. Conducting a fine search around the best coarse estimate will then result in a local best estimate within the vicinity of the coarse best estimate.

The above procedure, a rough first pass followed by a fine sweep around the rough estimate, is very beneficial. It cuts down search time by a factor of  $n^3$ . Choosing  $n = 3$ , for instance, will cut the rough search time to approximately 4% of the sequential search time. This is quite significant.

The inclusion of noise in the sensor output vector makes the coarse search less reliable; it is conceivable that noise could result in an anomalous coarse best estimate. This can be counteracted. The coarse-to-fine search can be modified so that instead of maintaining one “best” point around which to conduct a fine search, the  $m$  “best” points are maintained. At the conclusion of the coarse search, a fine search is conducted around each of these  $m$  points to find what is hoped to be a global best estimate. This modification makes the coarse-to-fine search method less susceptible to noise, since a single anomalous point will be less likely to result in an incorrect TTI estimate. Also, since a fine search around each of the  $m$  best coarse points takes an insignificant amount of time compared to the overall rough search, conducting several fine searches does not increase the search time significantly.

Our implementation of the algorithm chooses  $n = 3$  and  $m = 5$ . These values are somewhat subjective choices based on testing and empirical observations. These choices cut the total search time to under 5% of the search time for a sequential search. The innovation discussed in the following section decreases the search time even further.

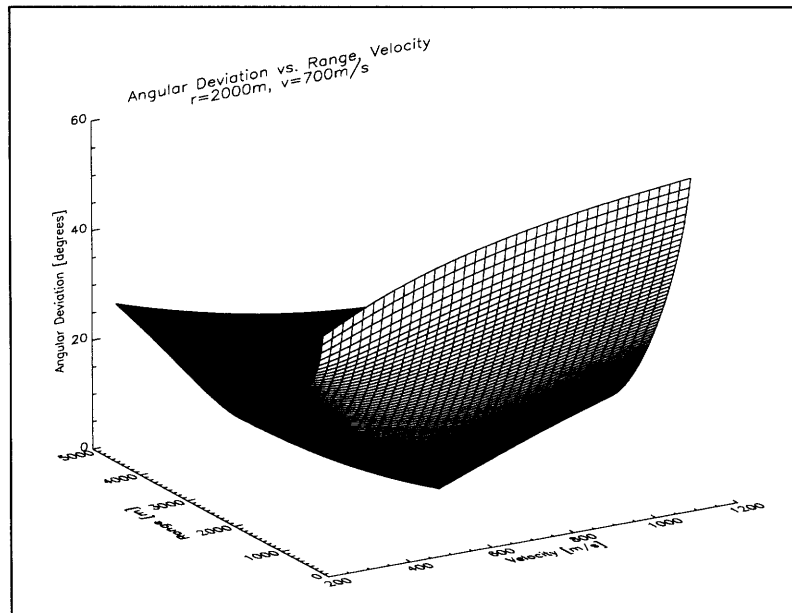
### 3.4.2 Search path abort conditions

The search can be further improved if we can identify instances when our search is clearly leading us away from a local error minimum. If error is increasing at successive points in the search, a great deal of time and computational expense could be saved by aborting the local search path and skipping ahead to a point in the search where successive errors

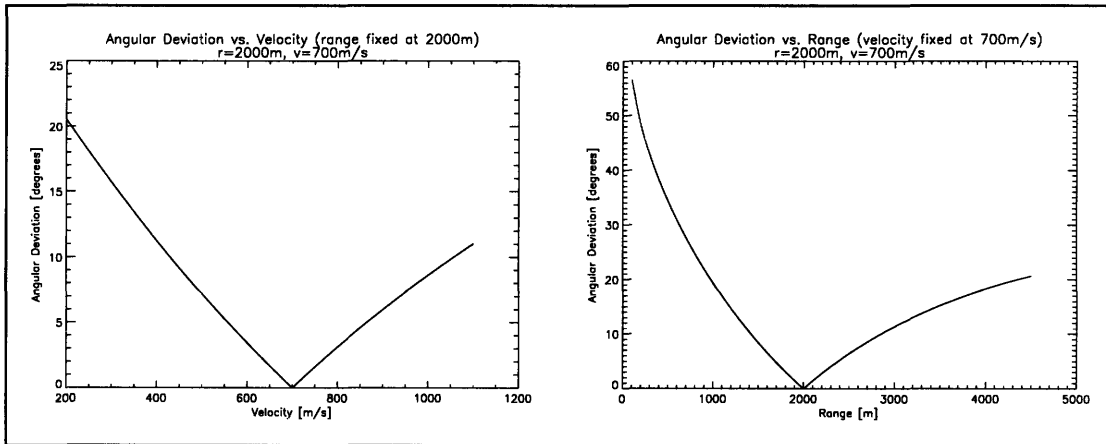
decrease. As long as error is decreasing, the search is progressing towards what is at least a local best estimate.

Let us examine the behavior of the error function for a sensor output vector obtained in a noiseless environment. Let us then compare this vector to every other canonical set entry from the same atmosphere. For such a vector obtained at range 2000 m and closing velocity 700 m/s (and in arbitrarily chosen atmosphere 5) we obtain the error surface shown in figure 3.6. Angular deviation is zero at canonical set range 2000 m and velocity 700 m/s; as we move away from this range/velocity pair, the error increases monotonically. We see that the error function is roughly bowl-shaped, with a minimum at the range/velocity pair corresponding to the source's actual physical parameters. Our search can take advantage of this fact.

Figure 3.7 shows cross-sections of the error surface shown in figure 3.6. (Note that the continuous curves in this figure are actually discretized at the level of range and velocity resolution in the canonical set—25 m and 25 m/s, respectively.) We see that sweeping along range or velocity produces a roughly V-shaped curve with a minimum at the source's actual range or closing velocity. This is the case even for fixed range or velocity not equal to the source's actual range or velocity; in these circumstances, however, the cross-section minimum is non-zero but still roughly centered about the correct parameter value.



**FIGURE 3.6: Canonical set angular deviation surface**



**FIGURE 3.7: Cross-sections of canonical set angular deviation surface**

If we can identify the point in a range sweep or velocity sweep when the error values begin increasing (i.e., when we have swept past the minimum) then we can abort the local search and move to another search path. For instance, if we are nesting range sweeps within velocity sweeps, and we see the error values increasing during a range sweep, then we should abort the range sweep, skip to the next velocity point, and begin another range sweep.

As always, the presence of noise complicates matters. Noise changes the smooth bowl in figure 3.6 to a rougher surface in which local minima and maxima abound. The same thing is true of the cross-sections in figure 3.7. Looking at two successive errors in a range or velocity sweep, we cannot be sure whether a gradient implied by their values is indicative of the true error function gradient or a noise artifact. Seeing an increase from a single error value to the next is no longer a sufficient abort condition. Seeing an increase in several successive error values might be. A monotonic error increase in, say, five successive error measurements is very likely a statement about the gradient of the noiseless error curve and not a noise anomaly. Our algorithm uses this criterion as its abort condition. These abort conditions greatly reduce the search time without reducing completeness.

In closing, we note that the bowl-shaped surface in figure 3.6 suggests a gradient descent method for finding the minimum error [24]. Though not implemented in this thesis, this innovation could conceivably reduce search time by an even greater degree.

### 3.5 Summary

The parameter-matching algorithm compares an observed sensor output vector to a canonical set of noiseless sensor output vectors to obtain a best estimate. Both the observed sensor output vector and the noiseless canonical set vectors are parametrized versions of the complete imaging sensor data set; this parametrization is essentially a coarse radial subsampling about AOA, and is necessary to reduce the sensor output vector size to a man-

ageable size. The criterion used for choosing the best canonical set entry is mean squared error, which, by using intensity normalization in the canonical set construction, is equivalent to angular deviation. The traversal of the canonical set is performed using a coarse-to-fine search method that also contains abort conditions. Together, these innovations greatly reduce the search time.

## 3.6 Analysis

This section reports on the performance of the parameter-matching algorithm. The algorithm was run on a large test set intended to span the range of possible physical parameters. Although testing and analysis was a large component of the design process, the analysis of this section is intended to represent the performance of the algorithm in its final form.

### 3.6.1 The test set

This section outlines the choice of physical parameters used to test the parameter-matching algorithm. With one minor difference, this is the same set of parameters used to test the curve-fitting algorithm described in the next chapter.

By convention, range is specified in the algorithm with respect to the end of the data window: that is, at the end of the 2 seconds of data collection. As noted in section 3.2.2, typical engagements generally begin at a range of not more than 5000 m. The maximum test set range is thus 4500 m. Since range is intrinsically non-negative, with 0 m range occurring when the source and sensor platform are co-located, the minimum test set range is 100 m. Source-sensor platform range was anticipated to have a pronounced influence on algorithm performance. For this reason, range resolution in the test set is fairly fine. The increment between range values is 100 m; thus the test set consists of ranges from 100 m to 4500 m at 100 m increments.

As was also described in section 3.2.2, most closing velocities range from 200 m/s to 1050 m/s. Four velocities were chosen to be representative of this range: 250 m/s, 500 m/s, 750 m/s and 1000 m/s. These four velocities provide a wide enough variety to span the possible set of velocities.

The radiant intensity of a typical missile is between 5 mW/sr and 500 mW/sr. Algorithm performance is expected to be quite dependent on source radiant intensity, since this parameter directly affects how many photons reach the sensor platform, and thus directly influences SNR. Five radiant intensity values were chosen to represent this test set dimension: 5, 15, 50, 150, and 500 mW/sr. This set of radiant intensities is nearly equally spaced on a logarithmic axis.

The parameter-matching algorithm uses its canonical set of sensor outputs as a representation of the atmospheres possible in the real world. The implicit assumption that comes with the canonical set is that any real-world scenario will produce sensor outputs that, in the noiseless case, mirror an entry in the canonical set. Thus the widest range of atmo-

spheres implicitly possible under the canonical set is the set of the 27 atmospheres within that set. Thus the test set was chosen to contain the same atmospheres found in the canonical set. The implications of this will be further examined in chapter 5.

In retrospect, it would have been illuminating also to test the parameter-matching algorithm on a set of atmospheres not included in the canonical set. Unfortunately, due to time limitations, this was not possible. The amount of time necessary to generate a new atmospheric set in the TWES and to perform algorithm runs, coupled with growing competition for computational resources at LIRIS as the thesis progressed, made this impossible.

A single altitude was chosen for inclusion in the test set. A change in altitude has several primary effects on the physical parameters that determine the sensor outputs. First, an increase in altitude results in a decrease in the extinction coefficient  $k$ ; this is due to atmospheric thinning. Second, with all else equal, scattering effects also decrease (for the same reason that extinction decreases). Ground effects become negligible as altitude increases sufficiently. Finally, if source altitude is fixed, a change in sensor platform altitude results in a change in elevation angle. However, we have assumed that elevation angle is zero.

The choice of numerous ozone concentration visibility and atmospheric arena permutations in the atmospheric dimension of the test set, described in the previous section, accounts for many of the physical changes that might also be encountered by varying altitude. This, coupled with the dubious validity of the TWES ground models (see section 2.5.3.1) suggests that a single test set altitude, high enough to reduce ground effects to the point of negligibility, is sufficient in the test set. An altitude of 1500 m was chosen as the nominal aircraft altitude; at this altitude, ground effects are virtually unnoticeable.

The above dimensions define the number of cases in the test set. With the choices above, the test set consists of 24300 cases. Each one of these cases represents a unique combination of specific values of physical parameters. In order to acquire a statistical basis for our results, we ran each case in the test set multiple times. This will allow us to speak of TTI estimate bias error and rms error. For each case in the test set, 50 repetitions were performed.

### 3.6.2 Performance

Reducing the massive volume of data generated by running the test set is no trivial matter. When all is said and done, we need to find a way to describe over 1 million (24300 x 50) data values in an understandable way.

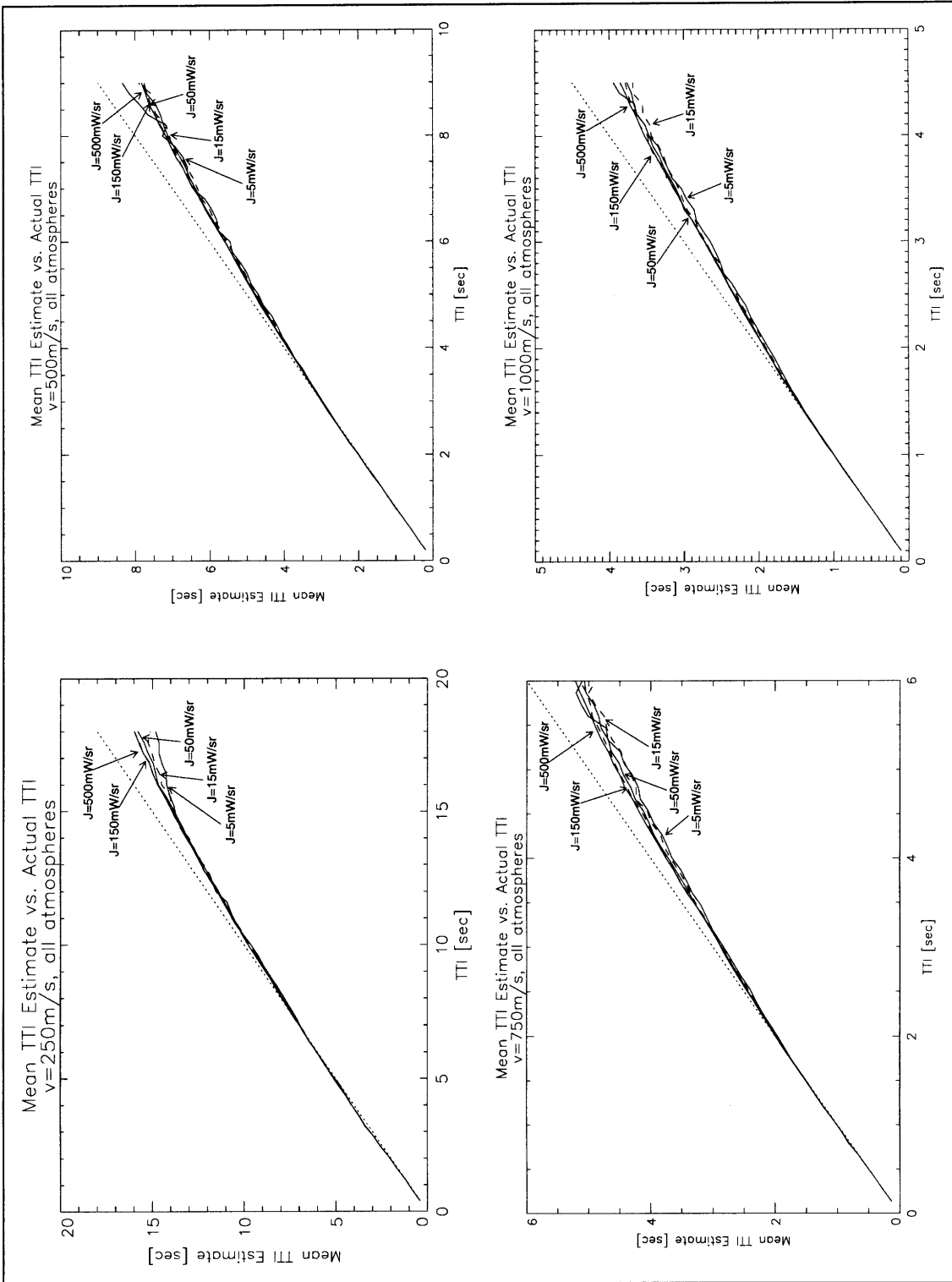
Two statistics are of primary importance in assessing the performance of a TTI estimation algorithm. The first of these is bias, or mean estimate error. The second of these is root mean-squared (rms) error. These two statistics provide a good indication of algorithm performance. Bias gives information about the average estimation error; rms error describes the spread of the estimation error. A large bias or a large rms error is generally indicative of poor algorithm performance. TTI estimation algorithms have generally had a low bias but a high rms error.

We will present bias and rms error information as an attempt to quantify the performance of each algorithm. It seems intuitively clearer in the sections that follow to present bias in terms of mean estimate, not mean estimate error. This is a trivial transformation; bias is simply TTI minus the mean TTI estimate.

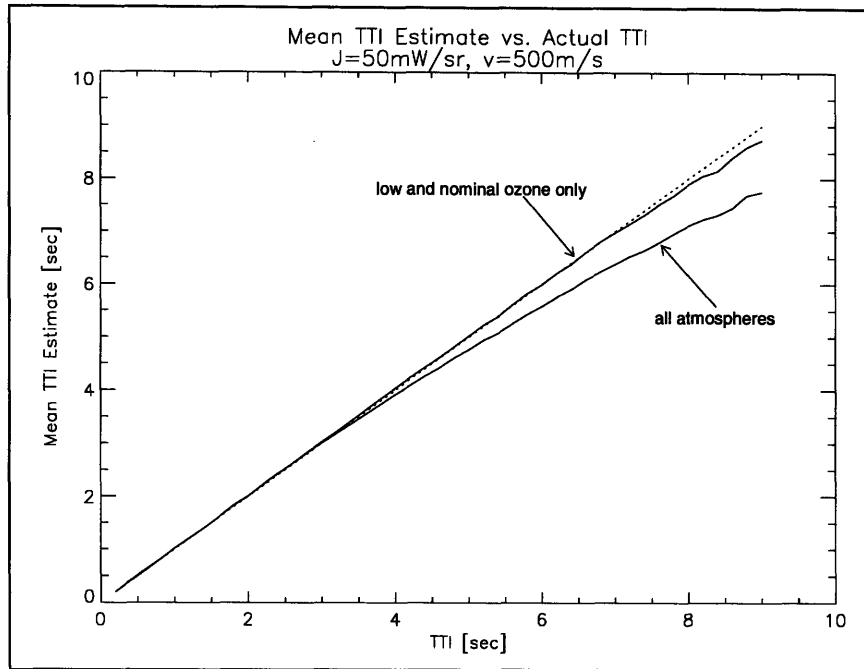
The parameter-matching algorithm produces estimates with a consistently low bias; the mean TTI estimate is nearly equal to TTI. Plots of the mean TTI estimate vs. actual TTI for the each of the four velocities in the test set are shown in figure 3.8. The format of these plots requires some explanation. Each individual plot in figure 3.8 represents the mean algorithm estimates at the entire set of ranges obtained for a single choice of velocity; the velocity corresponding to each plot is noted in the plot heading. There are five traces on each plot, corresponding to the five radiant intensity values in the test set. Each one of these traces depicts the change in mean TTI estimate, averaged over all atmospheres, as TTI (or equivalently, range) changes. Thus each trace has a minimum TTI value corresponding to the minimum range value (100 m) in the test set; likewise, the maximum TTI value in each trace corresponds to the maximum range value (4500 m) in the test set. Each plot contains a lightly dotted line representing zero bias, i.e., mean TTI estimate = TTI.

Several things are apparent from the plots in figure 3.8. First of all, the mean TTI estimate at a given range and velocity appears to be nearly constant for any source radiant intensity. That is, bias is nearly independent of source intensity, and a dim source results in the same mean estimation error as a bright source. This is not remarkably surprising. Noiseless sensor output vectors for a dim source and a bright source are scalar multiples of each other. They map to the same canonical set entry. Thus it seems plausible that observed sensor output vectors corresponding to bright and dim sources might map to the same canonical set entry. This appears to be the case.

Another thing that is immediately apparent from the plots is the deviation of mean TTI estimate from actual TTI at high TTI; i.e., the deviation of the traces from the dotted line representing zero bias. Analysis shows this deviation to be due primarily to the extremely low SNR in high ozone concentration atmospheres at distant ranges. Recall that each trace in figure 3.8 represents an average over all 27 test set atmospheres. If we exclude the high ozone concentration atmospheres from the averaging (i.e., if we average over only the 9 low and the 9 nominal ozone concentration atmospheres), we see this bias almost disappear in all cases. Figure 3.9 compares the bias plot for the 50 mW/sr - 500 m/s trace averaged over all atmospheres to that averaged over low and nominal atmospheres only. The dotted line in this figure represents zero bias. Note that the low and nominal ozone trace coincides with the dotted line even at distant range. (When we look at the bias over low ozone concentration atmospheres only, we see an even closer correspondence.) This is the case for every radiant intensity/closing velocity pair depicted in figure 3.8. The low SNR caused by the large absorption coefficient in high ozone concentration conditions heavily skews the overall bias.



**FIGURE 3.8: Parameter-matching algorithm: mean TTI estimate vs. TTI**



**FIGURE 3.9: Effect of high ozone concentration atmospheres on bias**

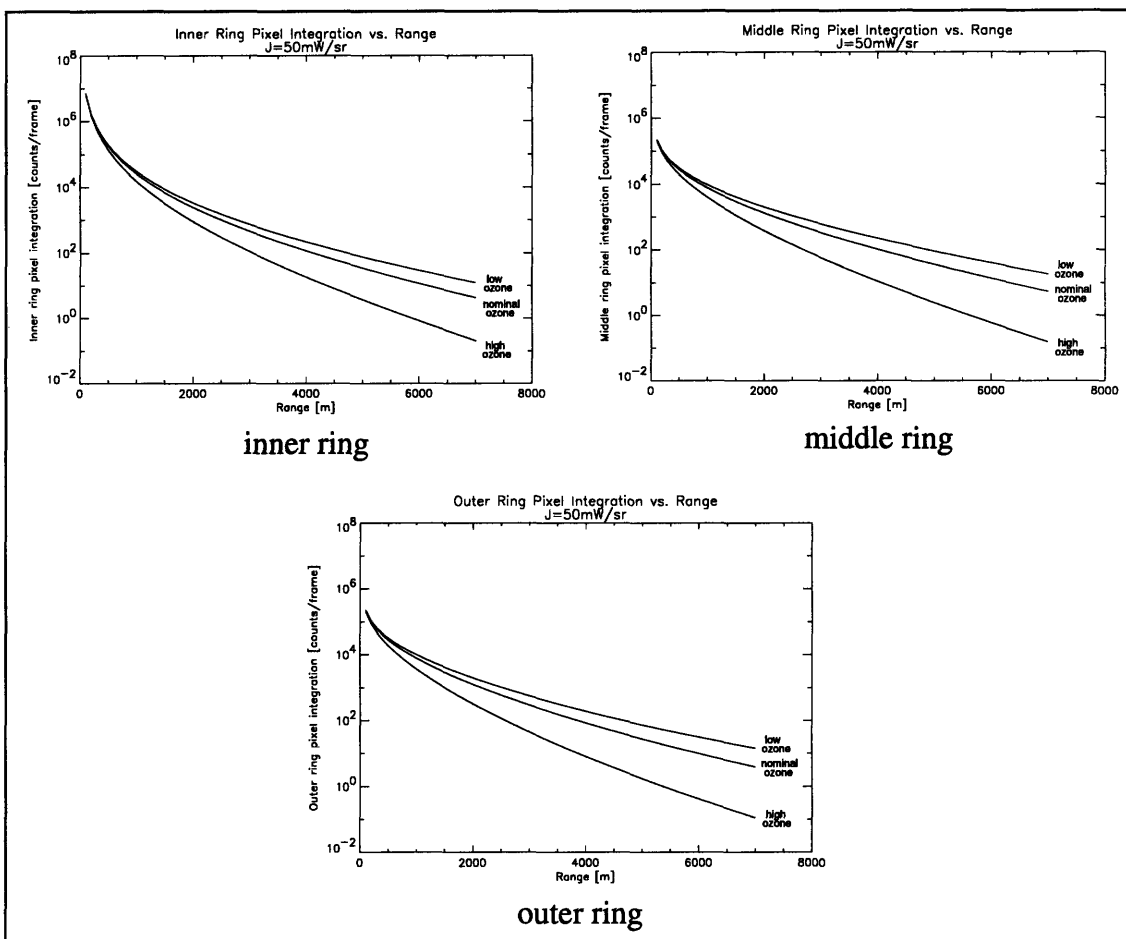
Before proceeding, let us examine exactly how ozone concentration influences SNR. Recall that absorption in the UV band is primarily determined by ozone concentration. A higher ozone concentration will thus result in a higher absorption and a lower SNR. Lowering SNR adversely affects algorithm performance. As SNR decreases, the smooth error function shown in figure 3.6 becomes very rough and local minima almost completely overwhelm the true shape of the surface.

Figure 3.10 shows how the three-ring parametrization depends on ozone concentration. The three plots in this figure correspond to the three rings (inner, middle, and outer) in the parametrization, and are representative of the effects of ozone concentration on signal level in a variety of circumstances. Each plot contains three traces, representing the pixel integration value for the ring averaged over each of the nine-atmosphere ozone concentration subsets (see table 3.1). Note that the signal level for the high ozone atmospheres is markedly worse than that for the low or nominal ozone atmospheres; at distant ranges, the high ozone concentration signal (in all rings) is roughly 20 times smaller than that in the nominal ozone atmospheres. This corresponds to a decrease in SNR (as defined in equation 2.5) by a factor of 4.5. (The traces in these plots are not evenly spaced because the low, nominal and high ozone concentrations used in the TWES simulations are not evenly spaced in parts per billion.) Clearly we will expect notably worse performance in a high ozone concentration atmosphere than in an atmosphere with a low or nominal concentration of ozone. Figure 3.9 shows that in fact this is the case.

There is a simple explanation for the uniformly negative bias in low SNR environments. Recall that the canonical set has a range dimension that contains values from 100 m to 4500 m in 25 m increments. Thus the maximum possible range estimate for any sensor



output is 4500 m. Let us consider a source approaching at 500 m/s from range 4500 m. We would expect the velocity estimates produced by the algorithm to be distributed about the true velocity of 500 m/s. The SNR would presumably determine the spread of this distribution; with a source of very high radiant intensity, nearly every velocity estimate would be 500 m/s and the spread would be very narrow.



**FIGURE 3.10: Effect of ozone concentration on three-ring parametrization**

Similarly, we would expect the range estimates to be distributed about 4500 m with a spread determined by SNR. However, the maximum canonical set range value is 4500 m—so essentially the upper half of this distribution is clipped. A range estimate can either equal 4500 m or be less than 4500 m. Thus the mean TTI estimate in this case is forced to be less than or equal to true TTI. This is why the traces in figures 3.8 and 3.9 are bowed. The maximum canonical set range alters the range estimate distribution and lowers its mean by refusing to allow range estimates over 4500 m. This becomes most apparent at high TTI corresponding to ranges near 4500 m, where more of the distribution is clipped.

The artificial introduction of a bias by the canonical set bounds could be remedied by extending the bounds of the canonical set. There are several problems with this approach. First of all, the canonical set represents a collection of physical parameters that spans as wide a range as plausible. Including source ranges of over 4500 m does not make sense because ranges of interest generally do not extend beyond this value. Bias manipulation should not be an excuse to step outside the bounds of physical plausibility. Second, the canonical set is already very large; increasing its bounds makes it even more unwieldy. Finally, extending the bounds might improve the bias, but would almost definitely worsen the rms error.

Figure 3.11 contains plots of the parameter-matching algorithm rms error averaged over all atmospheres. (Note that the vertical scale is not the same on every plot.) These plots are structured in the same way as the mean TTI estimate plots are structured. All four plots look roughly similar.

Again, several trends are immediately apparent. First of all, higher radiant intensities result in lower rms errors. This is to be expected. Higher source intensity improves SNR, thus improving algorithm performance. For an ideal, noiseless sensor output vector (corresponding to an infinite SNR) with zero bias, the rms error would be zero for any TTI. It is also apparent that rms error increases as range increases. Again, this can be explained in terms of SNR: fewer photons reach the sensor from a distant source than from a nearby source, thus the Poisson mean is lower, the SNR is higher, and the sensor output is more affected by noise.

Comparing the rms errors between plots, we see that rms error is higher at lower velocities. A quickly approaching source appears to produce a more reliable estimate than a slowly approaching source. The reason for this is not immediately apparent, but becomes clear if viewed in terms of sensor output.

A slow source results in slow sensor output growth. Over a short observation interval, the growth might appear linear or nearly constant. A fast source, on the other hand, will elicit a more pronounced sensor output growth over the same interval. Over our fixed 2-second observation interval, the sensor output growth for the fast source will show a steeper slope and more curvature than the slow source's sensor output. In a noisy environment, the steeper curve will stand out more; the flatter curve will tend to get lost in the noise. Basically, the fast source exhibits a sensor output growth with more of a visible trend, and thus it is more likely to stand out in a noisy environment and be identified by the algorithm.

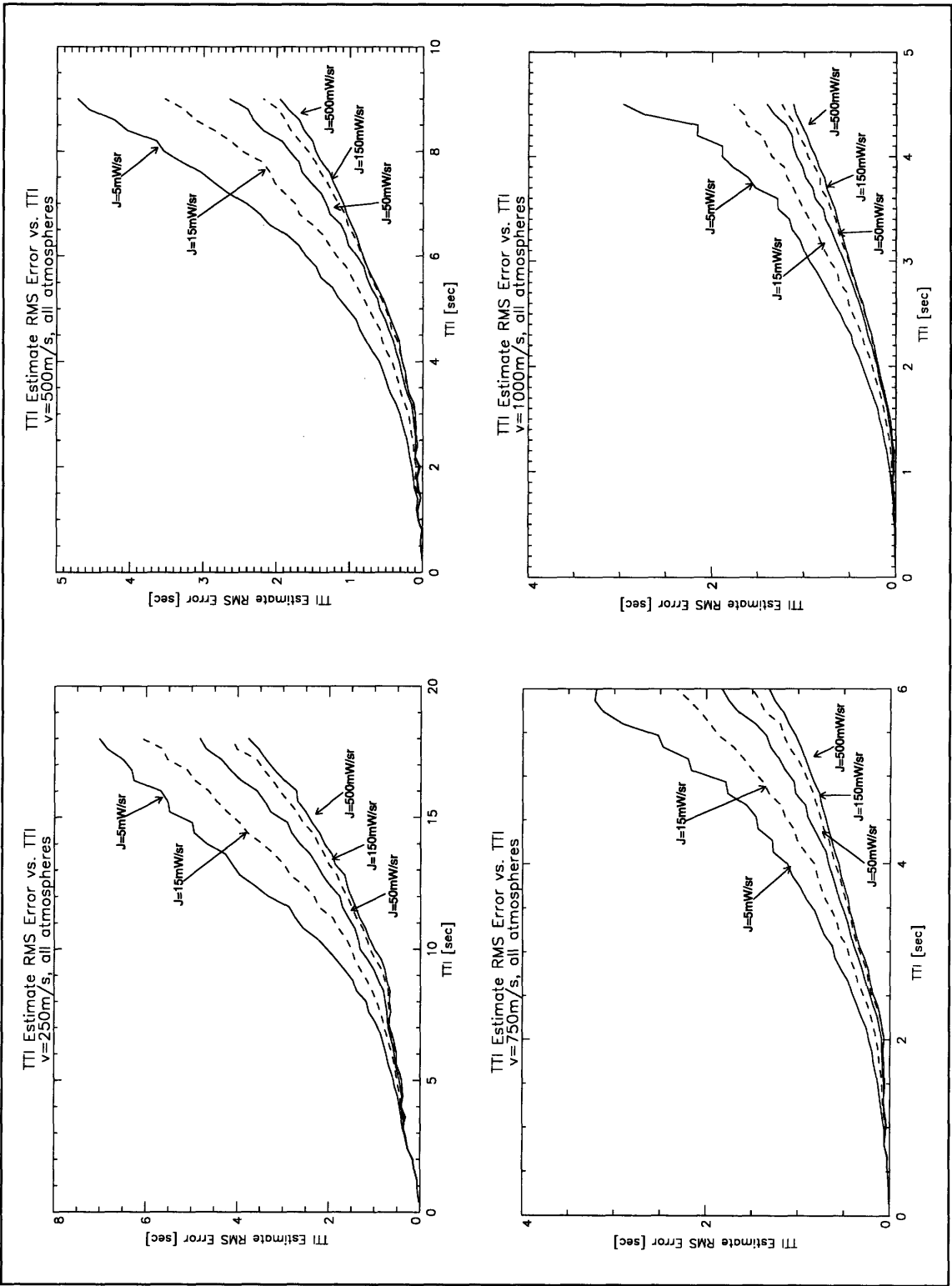
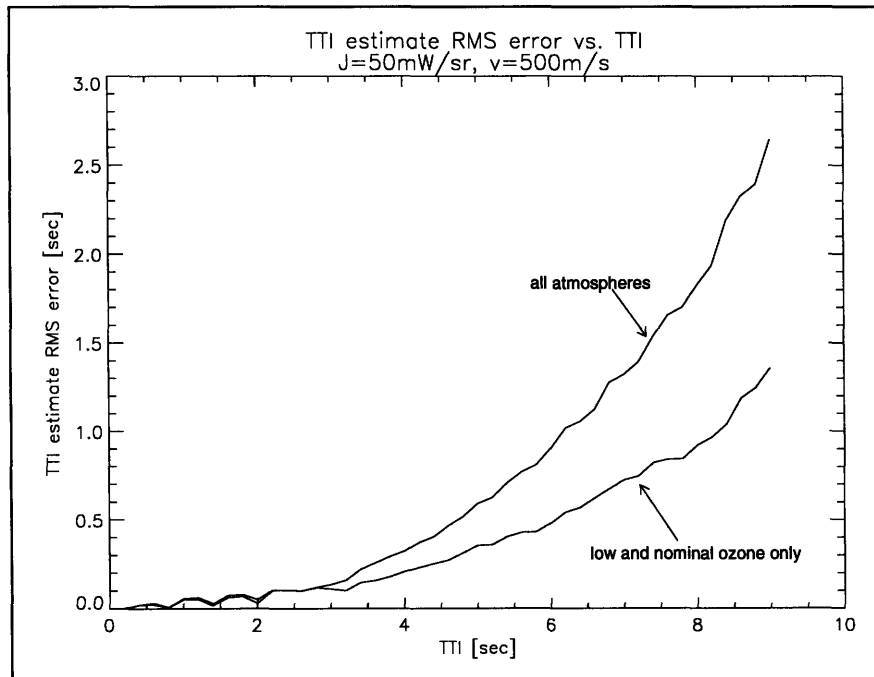


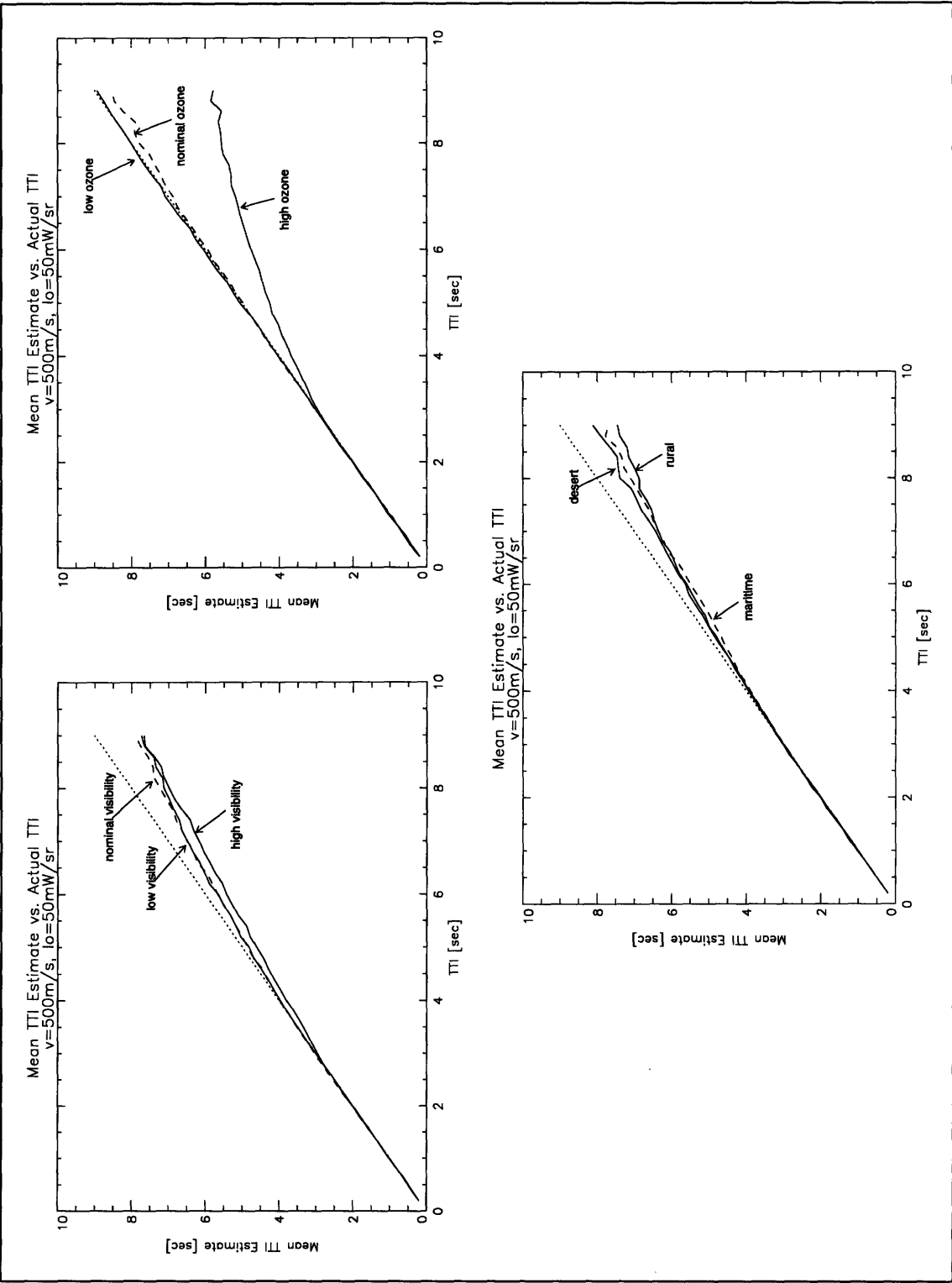
FIGURE 3.11: Parameter-matching algorithm: TTI estimate rms error vs. TTI

High ozone concentration atmospheres also influence the rms error plots in figure 3.11. If we consider only the low and nominal ozone concentration atmospheres, we see the rms error reduced from its plotted value in figure 3.11 in all cases. The reduction in rms error is generally slightly larger than the bias over all atmospheres. For example, if the bias over all atmospheres for a certain range/velocity/radiant intensity triplet is 1.5 s, then calculating the rms error over low and nominal ozone concentration atmospheres for that triplet might result in an rms error 1.6 s or 1.7 s below that shown in figure 3.11. Figure 3.12 shows the effects of excluding high ozone concentration atmospheres on the 50 mW/sr - 500 m/s trace; the results for other traces are similar.



**FIGURE 3.12: Effect of high ozone concentration atmospheres on rms error**

Figures 3.13 and 3.14 show how atmospheric effects influence the parameter-matching algorithm's performance. Figure 3.13 depicts mean TTI estimate vs. TTI for varying ozone concentrations, visibilities, and atmospheric arena. These traces were obtained at fixed closing velocity (500 m/s) and radiant intensity (50 mW/sr) in order to facilitate presentation; for other velocity/intensity pairs, the results are similar. The traces in these plots are labeled by atmospheric characteristic; each trace represents a value obtained by averaging over the nine atmospheres in table 3.1 that have the identified trait. The dotted line in the plots of figure 3.13 corresponds to zero bias.



**FIGURE 3.13: Parameter-matching algorithm: mean TTI estimate vs. TTI, atmosphere subsets ( $v = 500\text{m/s}$ ,  $J = 50\text{mW/sr}$ )**

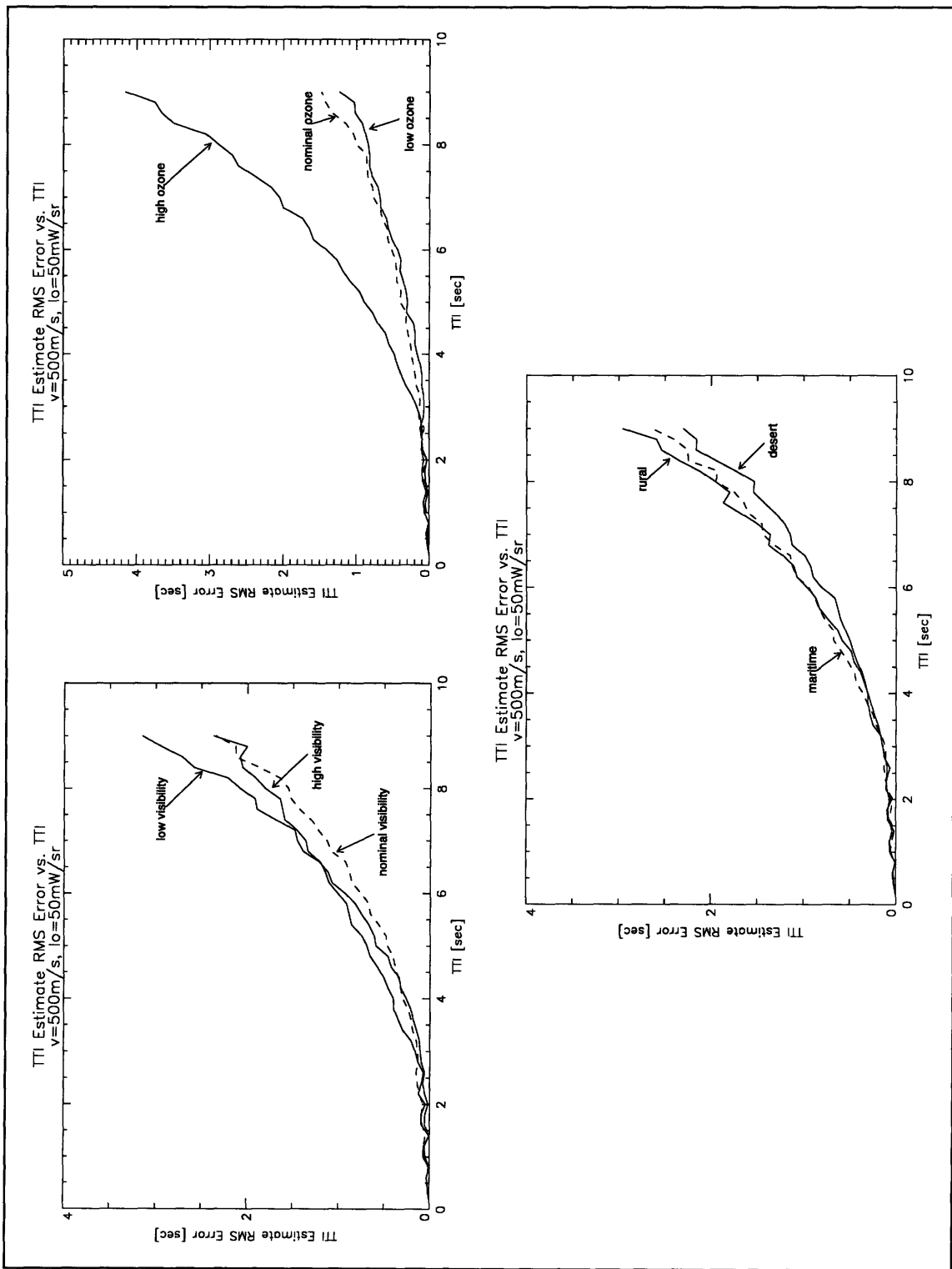
The results depicted by the ozone concentration plot in figure 3.13 are by now familiar. Again, we see the profound effect of ozone concentration on algorithm performance. The mean TTI estimate depends heavily on ozone concentration. As ozone concentration increases, SNR decreases, and algorithm performance suffers considerably. Note that the low ozone concentration algorithm runs exhibit almost no bias, and the nominal ozone concentration algorithm runs show very little bias. As shown in figure 3.10, the SNR for these atmospheres is much higher than it is in the high ozone concentration atmosphere. Recall that as SNR decreases, local minima proliferate in the error function and the global minimum corresponding to the actual TTI might be obliterated. In this case we might see an essentially random “best” estimate that varies with the exact characteristics of the noise in the sensor output vector.

Note that the visibility and atmospheric arena traces show non-zero bias at high TTI, just as those in figure 3.8. The explanation for this is the same as it was in conjunction with figure 3.8. Each of the traces in the visibility and atmospheric arena plots was obtained by averaging results over nine atmospheres; three of these nine atmospheres have a high ozone concentration. (See table 3.1.) If the three high ozone concentration atmospheres are excluded from the average, a near-zero bias results in all cases.

Visibility and atmospheric arena do not have much effect on performance; this is because these parameters primarily affect scattering and have a relatively minor effect on absorption. Thus they do not markedly affect SNR. Since the canonical set spans the rural, maritime and desert arena and low, nominal and high visibilities all with equal resolution, there is little reason to expect different visibilities or arenas to affect parameter-matching algorithm performance differently.

Figure 3.14 shows the rms error performance of the parameter-matching algorithm under different atmospheric conditions. Once again, the most readily apparent characteristic of this figure is the outlying high ozone concentration atmosphere trace. The rms error in this atmospheric subset is much higher than the rms error under any other conditions. Again, this is due to the excessively low SNR in this environment. As ozone concentration decreases to a nominal or low level, we see a dramatic reduction in rms error.

Again, visibility and atmospheric arena have relatively little effect on rms error. We do notice that a high visibility seems to produce a higher rms error than a nominal visibility atmosphere; this is unexplained. Roughly speaking, however, the rms errors for permutations of visibility and atmospheric arena are nearly constant. This implies that changes in scattering (determined primarily by visibility and atmospheric arena in the TWES) have little effect on the rms error. Again, excluding high ozone concentration atmospheres from the averaging process results in lower rms errors in all cases.



**FIGURE 3.14: Parameter-matching algorithm: TTI estimate rms error vs. TTI, atmosphere subsets ( $v = 500\text{m/s}$ ,  $J = 50\text{mW/sr}$ )**

Figures 3.13 and 3.14 show that algorithm performance is strongly dependent on ozone concentration, which manifests itself by way of SNR. Changes in scattering, however, are relatively insignificant in their effect on algorithm performance. Thus the algorithm is relatively tolerant to atmospheric effects that do not appreciably affect SNR. (Recall that this was one of the stated design goals of a TTI estimation algorithm.)

Generally speaking, the performance results described in this section seem to indicate that the parameter-matching algorithm is superior to any of the TTI estimation algorithms cited in section 2.7. The parameter-matching algorithm produces rms errors that are much lower than those of these previous efforts. A comparison of these results to past work is undertaken in chapter 5.



## 4.0 THE CURVE-FITTING ALGORITHM

The parameter-matching algorithm discussed in chapter 3 is implicitly limited by the scope and resolution of its canonical set. Its interpretation of the physical world is limited to a set of explicitly simulated conditions. If this algorithm performs in an environment with conditions similar to those found in this set, it should do well. However, if this algorithm is used in an environment that is foreign to the canonical set, there is no way to predict exactly how it will behave. It could produce incorrect or misleading estimates.

These robustness concerns are one reason that we are interested in developing a different algorithm. Another major reason is the size of the canonical set. The spatially limited canonical set ( $\psi$  and  $\phi$  both constrained to 0) described in the previous chapter is unwieldy, and further expansion of this set seems impractical. We would prefer an algorithm that is unaffected by storage limitations.

The parameter-matching algorithm operates by brute force, comparing observed data to a large set of simulated data. A different approach to the TTI estimation problem is to use information in the observed sensor output vector to calculate relevant physical parameters explicitly. There exist good physical models for the processes that map physical parameters to sensor outputs. Even if there is not an inverse model to derive physical parameters directly from sensor outputs, we can imagine making educated guesses of the physical parameters to choose a set that results in a sensor output vector similar to the observed output vector. This is what the curve-fitting algorithm does.

### 4.1 Phenomenology Revisited

The relationship governing the noiseless sensor output in the absence of scattering is:

$$\mu_i(t) = \frac{I_0 e^{-k_a r(t)}}{r^2(t)}, \quad i = 1 \quad \text{Equation 4.1}$$

$$\mu_i(t) = 0, \quad i \neq 1 \quad \text{Equation 4.2}$$

where  $\mu_i(t)$  is the output of pixel  $i$  at time  $t$ , and where pixel 1 is in the direct line-of-sight from the source—that is, pixel 1 encompasses AOA. (We shall assume for the remainder of this chapter that we have available an estimate of AOA; recall that such an estimate will generally be easy to obtain with our imaging sensors.) All other quantities are defined as in section 2.3.2. Note that we are continuing to assume constant source radiant intensity, narrow spectral response, nearly constant range over the integration period, atmospheric homogeneity, and nearly constant extinction coefficient over the sensor's narrow bandwidth, as was done in chapter 2.

The existence of scattering modifies the noiseless sensor output relationship to:

$$\mu_i(t) = \frac{I_0 e^{-kr(t)}}{r^2(t)} + \Theta_i(t), \quad i = 1 \quad \text{Equation 4.3}$$

$$\mu_i(t) = \Theta_i(t), \quad i \neq 1 \quad \text{Equation 4.4}$$

where  $\Theta_i(t)$  represents the contribution from scattered radiation at pixel  $i$  at time  $t$ , and where pixel 1 is again in the direct-line-of-sight from the source. Furthermore, recall that we do not in general have an explicit form for  $\Theta_i(t)$ . Scattering thus complicates the relationship between physical parameters and sensor output.

In the absence of scattering, i.e. when equations 4.1 and 4.2 hold strictly, it is a simple matter to derive TTI from observation of the evolution of the sensor vector. As discussed in section 2.3.2, intercept time corresponds to the instant when sensor output approaches infinity. A simple curve-fit applied to data of the form of equation 4.1 would predict TTI. Scattering wreaks havoc on this approach. Extracting a general dependence of  $\Theta_i$  on the wide range of possible physical parameters is extremely hard, and implementing a curve-fit based on such a dependence could prove to be extremely difficult.

Let us define:

$$U(t) = \sum_{i \in \mathcal{A}_\rho} \mu_i(t) \quad \text{Equation 4.5}$$

where  $\mathcal{A}_\rho$  is the region of all pixels within radius  $\rho$  of AOA.  $U(t)$  is thus an integration of pixel outputs similar to that described in section 3.1.2 and depicted in figure 3.1. We can re-express equation 4.5 as:

$$U(t) = \frac{I_0 e^{-kr(t)}}{r^2(t)} + \sum_{i \in \mathcal{A}_\rho} \Theta_i(t) \quad \text{Equation 4.6}$$

Thus it is apparent that by modifying  $\rho$  we are including more scattered radiation in  $U(t)$ .

In a single-pixel sensor system,  $\rho$  is fixed; it is impossible to change the FOV of a sensor that stares at a very large, fixed region of space. Any single-pixel sensor will typically collect a significant amount of scattered radiation. This is why curve-fitting efforts based on the direct radiation growth in equation 4.1 failed in single-pixel sensor systems.

The high resolution of our imaging sensor allows us to circumvent the scattering problem. Recall the sensor output parametrization scheme described in section 3.1.2. Pixels were integrated over three annular regions. The inner region (represented by the darkest gray-tone in figure 3.1) was dominated by direct radiation. At distant range, scattered radiation is more likely to reach the sensor in one of the outer two rings. If we constrict our FOV by reducing  $\rho$ , we will admit less scattered radiation. The summation term in equation 4.6 will contribute less and less to  $U(t)$ , and the direct radiation term will dominate. In these

circumstances, it might be possible to use  $U(t)$  as an approximation to the direct radiation received by the sensor.

For a good approximation, it seems desirable to restrict  $\rho$  as much as possible in order to mask out as much scattered radiation as possible. (Note that even with  $\rho$  reduced to the point where  $\mathcal{A}_\rho$  contains only the direct line-of-sight pixel, there will still be a scattered intensity component in  $U(t)$ ; some photons are scattered but still intercepted at AOA.) However, there are practical problems with decreasing  $\rho$  too far. Pixel resolution defines the minimum possible (non-zero) value of  $\rho$ . More importantly, sensor optics and readout multiplexer crosstalk result in a blurring of pixel data onto adjacent pixels. This results in direct radiation showing up not in a single pixel but in a small set of adjacent pixels [25]. Despite these non-idealities, the high resolution of our sensor potentially provides us with the means to develop a useful TTI estimation algorithm based on curve-fitting for the first time. By choosing an appropriate  $\rho$  and using values of  $U(t)$  from successive data frames, we hope to obtain a sequence of data that behaves approximately according to equation 4.1.

## 4.2 Sensor Output Growth Parameters

We are interested in fitting observed integrated pixel data to a curve of the form:

$$U(t) = \frac{I_0 e^{-kr(t)}}{r^2(t)} \quad \text{Equation 4.7}$$

This equation has three parameters:  $I_0$ ,  $r$ , and  $k$ . Unfortunately, the parameter of interest, TTI, does not appear in this equation. We would like to be able to produce an estimate of TTI directly; that is, we would like TTI to be an explicit parameter in the sensor output growth equation.

It is possible to rewrite equation 4.1, so that TTI is an explicit parameter. Under the assumption of constant closing velocity, we can write:

$$r(t) = r_i - vt \quad \text{Equation 4.8}$$

where  $r_i$  is range at the beginning of the engagement, and  $t$  is elapsed time from the beginning of the engagement. This is reducible to:

$$r(t) = v(TTI - t) \quad \text{Equation 4.9}$$

where TTI is the time-to-intercept at the start of the engagement. Substituting equation 4.9 into equation 4.7, we can write:

$$U(t) = \frac{I_0 e^{-kv(TTI - t)}}{v^2 (TTI - t)^2} \quad \text{Equation 4.10}$$

We now have an equation with an explicit dependence on TTI. Furthermore, this equation is still parametrized by only three values:  $I_0/v^2$ ,  $kv$ , and TTI. The first two of these three parameters are not intuitively pleasing; however, since the only parameter of interest is TTI, the loss in intuitive simplicity in the first two parameters is more than made up for by the appearance of TTI as a parameter.

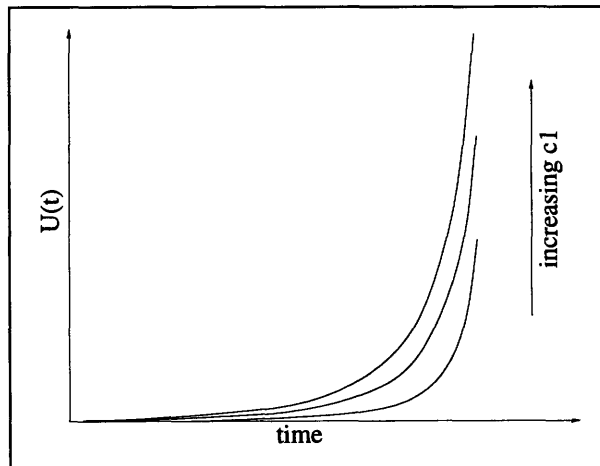
For convenience, let us define the vector  $c$ :

$$c = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} \frac{I_0}{v^2} \\ -kv \\ TTI \end{bmatrix} \quad \text{Equation 4.11}$$

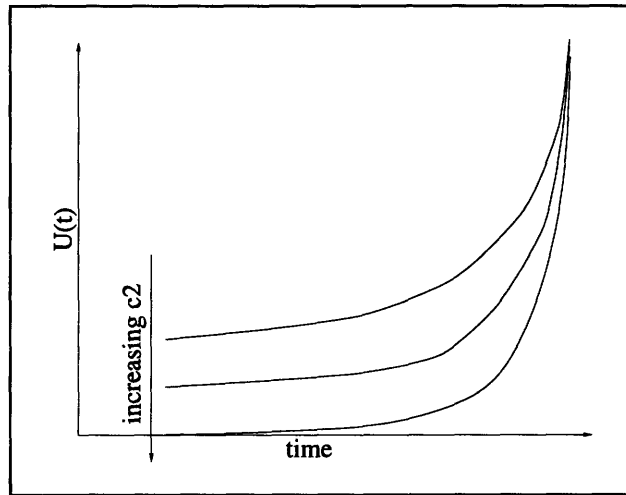
This definition allows us to rewrite equation 4.10 as:

$$U(t) = \frac{c_1 e^{c_2(c_3-t)}}{(c_3-t)^2} \quad \text{Equation 4.12}$$

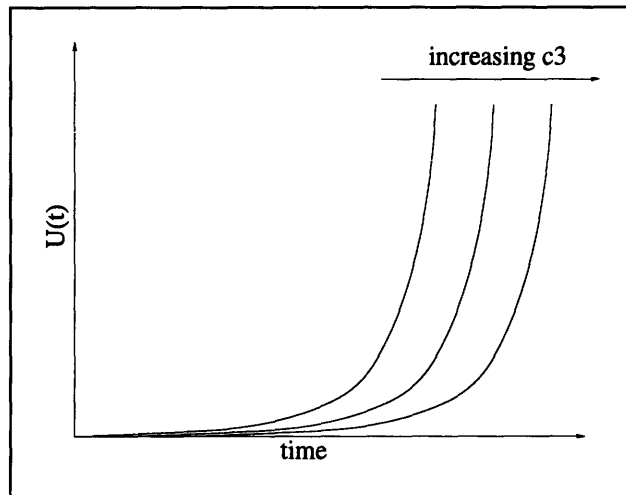
It is easy to see how  $U(t)$  varies with each of these parameters.  $c_1$  is an amplitude coefficient, and figure 4.1 shows the family of curves obtained by varying  $c_1$ . The  $c_2$  parameter represents an attenuation coefficient, and the family of curves defined by different  $c_2$  values is shown in figure 4.2. Finally, varying  $c_3$  (TTI) results in the family of curves depicted in figure 4.3.



**FIGURE 4.1: Family of  $c_1$  curves**



**FIGURE 4.2: Family of  $c_2$  curves**



**FIGURE 4.3: Family of  $c_3$  curves**

Sensor output growth is nonlinear, especially at high elapsed time. This is immediately apparent from equation 4.12 or figure 2.2. In order to facilitate our further analysis, we will modify our function  $U(t)$  so that it is more linear at distant range. Let us define:

$$j(t) = \ln(U(t)) \quad \text{Equation 4.13}$$

so that:

$$j(t) = \ln c_1 + c_2(c_3 - t) - 2 \ln(c_3 - t) \quad \text{Equation 4.14}$$

Figure 4.4 depicts how the familiar sensor output growth function (figure 2.2) looks when expressed in terms of log output, as in equation 4.14. Note that at distant range (at elapsed time near zero) this curve is roughly linear.

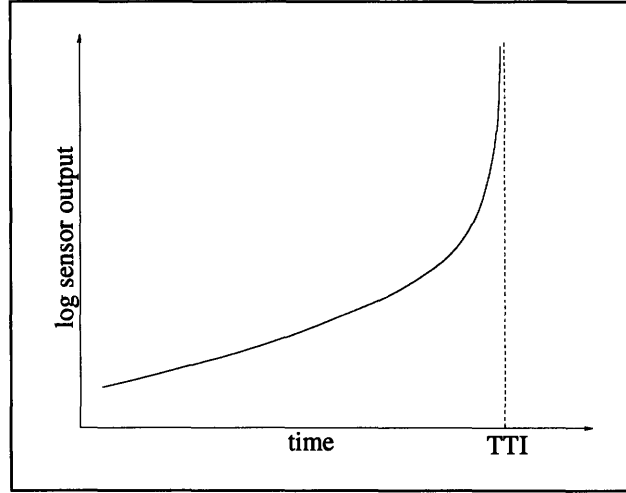


FIGURE 4.4: Log sensor output growth

### 4.3 The Newton-Raphson method

Having defined the form of the curve to which observed sensor data will be fit, we will now describe the numerical heart of the algorithm: the Newton-Raphson method for solving nonlinear equations.

#### 4.3.1 Error minimization

We are interested in obtaining an estimate of TTI by fitting the observed log integrated pixel sensor output vector to a curve of the form in equation 4.14 by choosing  $c$  so that the squared error between the fitted data and the observed data is minimized. Let us define  $w$  as a series of observed log integrated pixel sensor measurements obtained from sequential frames. Let us also define  $j$  as a sampled version of  $j(t)$  in equation 4.14. We are interested in minimizing the quantity:

$$\epsilon = (j - w) \cdot (j - w) = \sum_{i=1}^n (j[i] - w[i])^2 \quad \text{Equation 4.15}$$

where  $j$  and  $w$  both contain  $n$  elements. Expanding, we obtain:

$$\epsilon = \sum_{i=1}^n ((\ln c_1 + c_2 (c_3 - t_i) - 2 \ln (c_3 - t_i)) - j[i])^2 \quad \text{Equation 4.16}$$

where  $t_i$  is the elapsed time at sensor frame  $i$ .

At any error minimum, we know that:

$$\frac{\partial \epsilon}{\partial c_1} = 0 \quad \text{Equation 4.17}$$

$$\frac{\partial \epsilon}{\partial c_2} = 0 \quad \text{Equation 4.18}$$

$$\frac{\partial \epsilon}{\partial c_3} = 0 \quad \text{Equation 4.19}$$

That is, the directional derivative of the error function is zero in all directions. The Newton-Raphson method exploits this property to find an estimate of  $c_1$ ,  $c_2$ , and  $c_3$ .

### 4.3.2 Newton-Raphson for explicit equation solving

The Newton-Raphson method is a procedure for iteratively solving a nonlinear equation. The method requires knowledge of the function, an observed value and an initial guess in order to operate. The reader is directed to [26] for a detailed description of this method; we will present a brief overview.

Suppose that we observe the quantity  $y$  generated by some process that we model by  $f$ , i.e.:

$$y = f(x) \quad \text{Equation 4.20}$$

Suppose that we wish to determine  $x$  in a case in which  $f$  is difficult to invert. The Newton-Raphson method provides us with a way to do this. If we can numerically calculate the gradient matrix  $Df$ , then the Newton-Raphson method is able to produce a series of estimates of  $x$  recursively using the following formulae:

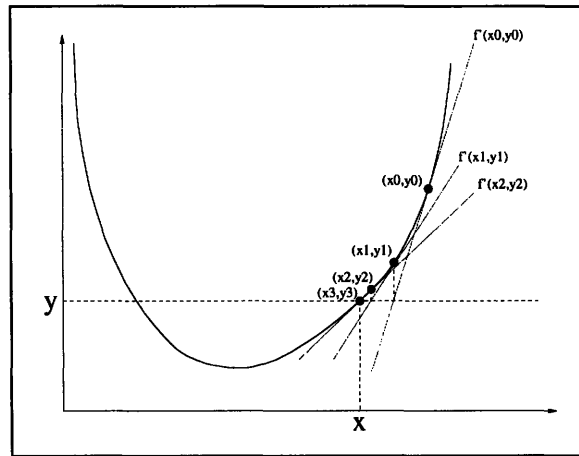
$$x_{i+1} = x_i + (Df)^{-1} (y - y_i) \quad \text{Equation 4.21}$$

$$y_{i+1} = f(x_{i+1}) \quad \text{Equation 4.22}$$

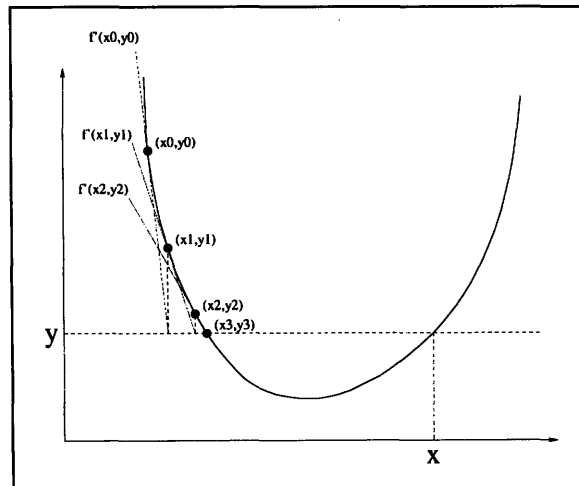
where  $x_i$  denote the estimates of  $x$ , and  $x_0$  is an initial guess provided by the user. Newton-Raphson can be visualized as a procedure for riding the gradient of  $f$  to find  $x$ . This approach is based on the assumption that over small ranges of  $x$ , the model  $f$  is nearly linear.

Figure 4.5 presents a pictorial representation of the Newton-Raphson iteration when  $y$ ,  $f$ , and  $x$  are scalar. This figure depicts convergence to the desired solution. Note that it is possible for Newton-Raphson to converge to an incorrect estimate; this possibility is presented in figure 4.6. (When Newton-Raphson is used for explicit equation solving, con-

vergence to an incorrect estimate can only occur in the case when  $f$  is non-invertible, i.e., when the relationship between  $x$  and  $y$  is not one-to-one.) It is also possible, given a poor initial guess or a poorly behaved model  $f$ , for the Newton-Raphson iteration not to converge at all. As an example, consider a model  $f$  that exhibits rapid fluctuations over a small range of  $x$ . In a case like this, the linearization on which the Newton-Raphson method is based might break down, and successive iterations might not converge.



**FIGURE 4.5: Newton-Raphson iteration in two dimensions**



**FIGURE 4.6: Newton-Raphson iteration: incorrect convergence**



### 4.3.3 Newton-Raphson for error minimization

Generally speaking, any model  $f$ , whether scalar- or vector-valued, will contain inaccuracies. Real-life systems are also subject to noise. In the presence of noise or model inaccuracies, we might observe a  $y$  not equal to any  $f(x)$ . Consider, for instance, the model  $y = x^2$ . This model cannot produce any negative values of  $y$ . If the system modeled by this equation actually includes unmodeled noise, a negative value of  $y$  might be observed. The Newton-Raphson method as it has been described so far would be unable to produce an estimate of  $x$  that results in the observed  $y$ . Fortunately, a simple modification allows us to use the Newton-Raphson method in the presence of noise.

We will not use the Newton-Raphson method to solve equation 4.12 explicitly for  $c$ , but instead we will use this approach to minimize the error  $\epsilon$  in equation 4.16. Recall from equations 4.17-4.19 that at the minimum of the error function, all directional derivatives are equal to zero. We will use Newton-Raphson to solve for the vector  $c$  that satisfies these equations. Let us define:

$$g = \begin{bmatrix} \frac{\partial \epsilon}{\partial c_1} \\ \frac{\partial \epsilon}{\partial c_2} \\ \frac{\partial \epsilon}{\partial c_3} \end{bmatrix} \quad \text{Equation 4.23}$$

so that our objective is to find  $c$  that satisfies:

$$g(c) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad \text{Equation 4.24}$$

This construction allows us to rewrite the Newton-Raphson equations 4.21 and 4.22 as:

$$c_{i+1} = c_i - (Dg)^{-1} d_i \quad \text{Equation 4.25}$$

$$d_{i+1} = g(c_{i+1}) \quad \text{Equation 4.26}$$

This implementation of Newton-Raphson is still an iterative process that solves for a function input given a specific observed output. The input is  $c$  from equation 4.11. The output is the error-derivative function  $g(c)$ . The function relating the input and output is the gradient of the curve-fit error. This implementation of the Newton-Raphson method finds the parameter vector  $c$  that produces the gradient value  $[0 \ 0 \ 0]^T$ . We know that any  $c$  that pro-

duces this zero gradient value corresponds to a local minimum or local maximum in the error function. Newton-Raphson seeks a local minimum by construction.

We can easily calculate the  $Dg$  matrix in equation 4.25. Note that:

$$Dg = \begin{bmatrix} \frac{\partial g_1}{\partial c_1} & \frac{\partial g_1}{\partial c_2} & \frac{\partial g_1}{\partial c_3} \\ \frac{\partial g_2}{\partial c_1} & \frac{\partial g_2}{\partial c_2} & \frac{\partial g_2}{\partial c_3} \\ \frac{\partial g_3}{\partial c_1} & \frac{\partial g_3}{\partial c_2} & \frac{\partial g_3}{\partial c_3} \end{bmatrix} \quad \text{Equation 4.27}$$

Calculations yield:

$$Dg = \begin{bmatrix} n & -\sum_{i=1}^n (c_3 - t_i) & -nc_2 - 2 \sum_{i=1}^n \frac{1}{c_3 - t_i} \\ -\sum_{i=1}^n (c_3 - t_i) & \sum_{i=1}^n (c_3 - t_i)^2 & \left( \begin{array}{l} 2c_2 \sum_{i=1}^n (c_3 - t_i) \\ + 2 \sum_{i=1}^n \log(c_3 - t_i) \\ -nc_1 + n + \sum_{i=1}^n j[i] \end{array} \right) \\ \left( \begin{array}{l} -nc_2 \\ -2 \sum_{i=1}^n \frac{1}{c_3 - t_i} \end{array} \right) & \left( \begin{array}{l} 2c_2 \sum_{i=1}^n (c_3 - t_i) \\ + 2 \sum_{i=1}^n \log(c_3 - t_i) \\ -nc_1 + n + \sum_{i=1}^n j[i] \end{array} \right) & \left( \begin{array}{l} nc_2^2 + 2c_2 \sum_{i=1}^n \frac{1}{c_3 - t_i} \\ + (4 + 2c_1) \sum_{i=1}^n \frac{1}{(c_3 - t_i)^2} \\ -4 \sum_{i=1}^n \frac{\log(c_3 - t_i)}{(c_3 - t_i)^2} \\ -2 \sum_{i=1}^n \frac{j[i]}{(c_3 - t_i)^2} \end{array} \right) \end{bmatrix} \quad \text{Equation 4.28}$$

Thus we have the means to implement the Newton-Raphson iteration equations 4.25 and 4.26. In order to produce a TTI estimate using these equations, we also need an initial guess of  $c$ . Our procedure for obtaining an initial guess is described in section 4.3.4.6.

#### 4.3.4 Other Newton-Raphson design issues

The Newton-Raphson method is an iterative procedure. It can in theory run forever, producing estimates that are more and more precise. It is necessary to specify parameters to control the final estimate precision. It is also necessary to decide what form the input data should take, and how much input data should be provided. This section outlines these and other related issues.

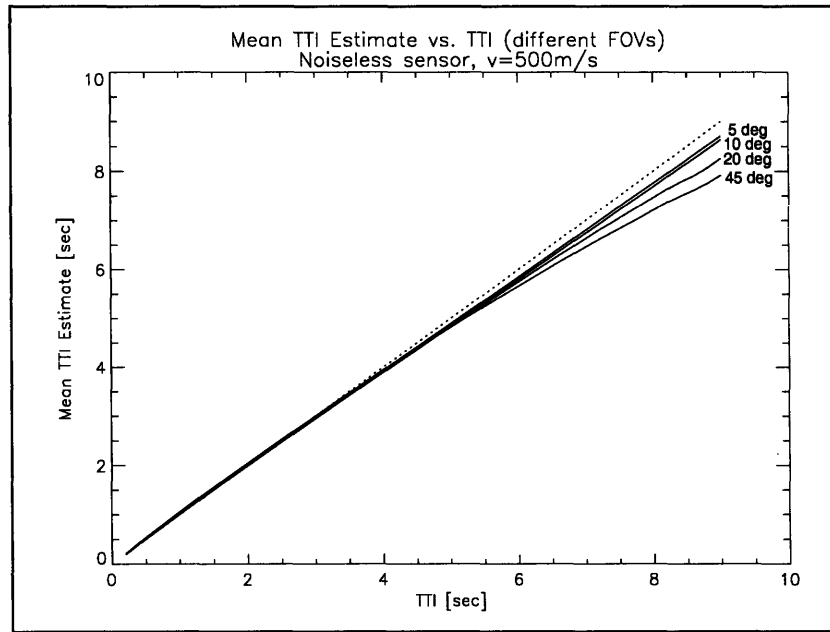
##### 4.3.4.1 Choice of FOV

In section 4.1 we described how narrowing the FOV about AOA results in a sensor output vector that contains a higher proportion of direct radiation. In a noiseless system it is desirable to restrict FOV as much as possible, subject to the pixel blurring constraints also presented in section 4.1. In a noisy system, however, there are problems that arise when the FOV is reduced too much.

Initial testing indicated that at low signal levels, curve-fitting was generally unsuccessful. The Newton-Raphson iteration often diverged, and the estimate of  $c$  was generally unreliable in the cases where the iteration did converge. There is a point at which SNR is simply too low and the sequence of integrated pixel data too poorly behaved to ensure reliable algorithm performance. We can counteract this problem and increase the SNR by widening FOV in order to admit more radiation. However, this is contrary to our original intention in narrowing the FOV—that is, to mask out scattered radiation. On the one hand, we would like to reduce FOV to ensure that our pixel integration is primarily influenced by direct radiation, as our model requires. On the other hand, we would like to increase FOV so that even at distant ranges, SNR is high enough for the algorithm to produce an estimate despite possible deviations from ideality.

Let us consider a constant-intensity source approaching a noiseless sensor. In this case, any inaccuracies in the TTI estimate produced by the algorithm are due to the presence of scattered radiation since in effect the SNR is infinite. Figure 4.7 shows how, at distant ranges, the TTI estimate does in fact deviate from the ideal estimate for any choice of FOV. The dotted line in this figure corresponds to a perfect (i.e., zero-bias) TTI estimate. The solid lines, annotated with FOV, were obtained by averaging over the TTI estimates from all 27 atmospheres in table 3.1. Note that because this plot was obtained from algorithm runs in a noiseless environment, source radiant intensity is irrelevant and multiple runs are unnecessary. This figure clearly shows that widening FOV results in poorer algorithm performance. This is due to the admission of more scattered radiation into the pixel integration  $U(t)$ . Because the traces in figure 4.7 are averages over 27 atmospheres, a specific set of atmospheric conditions may result in relatively more or less deviation from ideality than shown in this figure. For instance, a low visibility atmosphere (indicative of high scattering) will produce a trace with more deviation than that exhibited by a trace

obtained in a high-visibility atmosphere. Figure 4.7 is useful in providing an indication of behavior in an “average” atmosphere, and in depicting the general trend of more scattered radiation worsening algorithm performance.



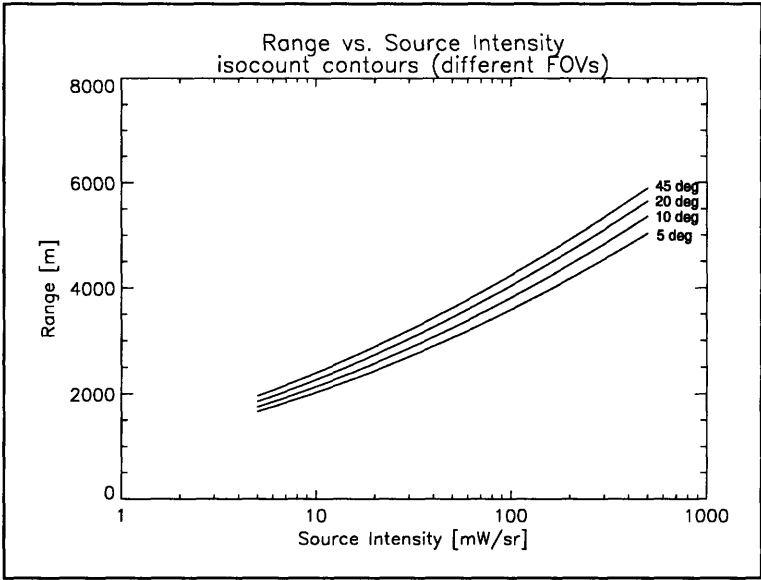
**FIGURE 4.7: Effect of FOV on mean TTI estimate**

Using the data in figure 4.7, we can identify a range value (for a given choice of FOV) beyond which the presence of scattered radiation makes the TTI estimate “too unreliable.” (There are many ways to define the term “too unreliable;” one such way is to identify the point at which the estimate deviates from the actual value by more than 10%.) Let us call this point the scattering threshold. Note that the scattering threshold is dependent on sensor FOV, but not on source radiant intensity.

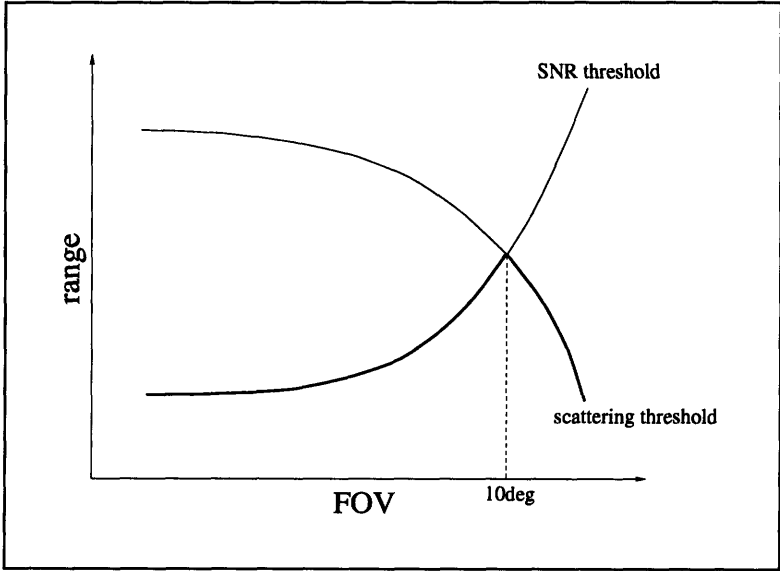
As described above, there is also a signal level below which the SNR is too low for the Newton-Raphson method to converge reliably. Let us call this point the SNR threshold. For any choice of FOV, and for a source of a given radiant intensity, this SNR threshold can be translated into a range threshold; that is, it is possible to identify a range beyond which the signal received by the sensor will be too low to ensure reliable algorithm performance. We can plot a series of “isocount contours,” or curves indicating at what range a source of given radiant intensity will produce a given  $U(t)$  for various FOVs. A group of isocount contours for a representative value of  $U(t)$  is depicted in figure 4.8. The multiple curves in this figure represent different choices of FOV. For a given radiant intensity and FOV, we can use a figure like this to determine the range of the SNR threshold.

The scattering threshold and the SNR threshold each define an approximate maximum source-sensor platform range at which a reliable TTI estimate may be made for a source of

given radiant intensity. We can plot the FOV dependence of both the SNR threshold and the scattering threshold on a single plot, as in figure 4.9. Note that for any choice of FOV, the maximum range at which the algorithm will be able to produce a reliable TTI estimate is given by the lower of these two thresholds, represented by the bold line in the figure. We would like to maximize this range. We can do so by choosing FOV to correspond to the point where the two curves in figure 4.9 cross. For a source of moderate brightness, this occurs at an FOV of approximately 10 degrees. This is the FOV used in the curve-fitting algorithm; it is the  $\rho$  of equation 4.5.



**FIGURE 4.8: Effect of FOV on isocount contours**



**FIGURE 4.9: SNR and scattering thresholds**

#### 4.3.4.2 Newton-Raphson abort conditions

During algorithm execution, it is possible to identify cases in which the algorithm is not exhibiting behavior commensurate with a convergent iteration. Execution should be aborted in these cases so that the algorithm can be run again with a different initial guess or with new data. One sign that the algorithm is not converging is an increase in the distance between successive iteration estimates. For a convergent iteration,  $|c_{i+1} - c_i|$  will generally (though not necessarily monotonically) decrease as  $i$  increases. If this behavior is not observed, then the iterative process is divergent and should be aborted.

Another case in which the algorithm execution should be aborted is when the error corresponding to the iteration estimate is not decreasing. This is essentially same condition as above, but applied to  $\epsilon$  instead of  $c$ . This condition states that  $|\epsilon_{i+1} - \epsilon_i|$  should generally decrease as  $i$  increases.

When one of these abort conditions is met, it is typically because of a low SNR and poorly behaved data. Trying to fit a sensor output growth curve to a sequence of data that is essentially low-mean noise is futile. A low SNR transforms the shape of the error function from a relatively smooth bowl with a single local minimum to a rough mess with many local minima but a relatively small set of initial guesses that will provide a convergence to one of these minima.

One final abort criterion, forced by equation 4.25 and not by any iterative concerns, occurs when  $Dg$  is singular. This singularity could be due to computational issues like roundoff or truncation, but it could also be caused by iterating to a genuine flat spot on the error derivative surface. In either case, algorithm progress is forced to halt even if the iteration appears to be converging. While the other abort conditions are relatively common, this condition was encountered on only a handful of occasions in the hundreds of thousands of times the curve-fitting algorithm was run.

#### 4.3.4.3 Error threshold

Theoretically, the algorithm could iterate forever, producing more and more precise estimates of TTI that (while possibly incorrect) would satisfy the solution condition of equation 4.24 arbitrarily well. (Note that in reality, estimate precision is eventually limited by the precision of the hardware.) From a practical standpoint, the algorithm should stop running as soon as it has a reasonably good estimate. This frees up the processor for other things, such as subsequent algorithm runs.

Essentially, we must answer the question: “How good is good enough?” We do this by setting a tolerance for termination, or a threshold for the quantity  $|\epsilon_{i+1} - \epsilon_i|$ . Once the subsequent algorithm errors dip below this threshold, the estimate is deemed sufficiently precise and the iteration is halted. The value chosen for the curve-fitting algorithm, somewhat arbitrarily, is  $10^{-5}$ .

#### 4.3.4.4 Duration of data collection

Another design issue is the choice of how much data to provide the algorithm. This is exactly the same issue that was discussed in section 3.1.3 in the context of the parameter-matching algorithm. This choice represents a trade-off: a longer observation interval generally results in a better estimate, but also requires more latency before the algorithm begins computation and produces an estimate. As in the parameter-matching algorithm, 2 seconds was chosen as the data window length. Sensor data for the curve fitting algorithm was collected at a frame rate of 100 Hz; thus the algorithm operates on 200 frames of data.

Recall that the parameter-matching algorithm operates on 2 seconds of sensor data collected at 10 Hz, or twenty frames of data. The parameter-matching algorithm frame rate was chosen to be slower in the interest of minimizing the parameter-matching data set. A 100 Hz frame rate provides more data points. Because we are fitting data to a curve, we prefer a long sequence of data with incremental changes between successive points to a short sequence of data with larger jumps between points. The penalty of using a higher frame rate is a lower SNR; integrating at 100 Hz produces a sensor output with Poisson mean 10 times lower than that obtained by integrating at 10 Hz.

#### 4.3.4.5 Wavelength dependence correction factor

Initial algorithm runs showed that the TTI estimate, even for a bright source at close range, was consistently inaccurate by a nearly constant, near-unity multiplicative factor. In other words:

$$\hat{TTI} \approx \zeta \times TTI \quad \text{Equation 4.29}$$

for most estimates. This proved to be the case even in the absence of noise and even with the suppression of scattering by the TWES. On the other hand, the algorithm correctly predicted TTI when fed data that was explicitly generated to be of the form given in equation 4.7.

Upon close inspection, the source of this “error” was found. Recall the assumptions in chapter 2 of narrow spectral response for our sensor and a nearly constant  $k$  over this narrow bandwidth. While these are generally good assumptions, they are not strictly true. Our blackbody source radiates energy over the entire spectrum, our sensor’s conversion efficiency does vary with wavelength, and the extinction coefficient is also dependent on wavelength.

The deviation of the TTI estimate from TTI by a nearly constant factor indicates that our assumptions are slightly flawed. The wavelength dependency influences the sensor output in such a way as to result in a TTI estimate that is incorrect by a nearly constant multiplicative factor. We could attempt to correct this problem by rederiving the sensor output equation, and designing an algorithm that can estimate TTI for a sensor with wavelength-dependent  $\kappa$ , an atmosphere with wavelength-dependent  $k$ , and a source with wavelength-

dependent  $N$ . However, it is much easier simply to identify  $\zeta$  in equation 4.29 and to divide the current algorithm's output by its value.

Analysis of early algorithm runs indicated that both bias and rms error of the noiseless, direct-radiation (i.e., ideal circumstances) algorithm estimates could be minimized by choosing  $\zeta$  in equation 4.29 approximately equal to 1.07. Thus this "wavelength dependence correction factor" was applied to the estimate for each subsequent algorithm run, as a back-end modification of the curve-fitting algorithm estimate. All quoted TTI estimates, including all those in the analysis to follow, have been corrected by this factor.

#### 4.3.4.6 Initial guess

The Newton-Raphson iteration requires an initial guess of the parameter vector  $c$  as a seed. A poor initial guess can result in an inaccurate or non-convergent iteration. It is not obvious how to provide an initial guess of  $c$  that is good enough to result in a reliable TTI estimate. This initial guess problem has been tackled before. In [27], Scheffé outlines a method for producing a reasonably good initial guess of physical parameters for a numerically based TTI estimation algorithm. Her approach assumes an arbitrary value for the initial TTI guess and then uses this value to get ballpark figures for other parameters. Given TTI, or  $c_3$  in equation 4.11, the equation governing direct radiation sensor output growth is a function of only two unknown parameters,  $c_1$  and  $c_2$ . Scheffé shows that in this case, there is a simple closed-form solution for the two parameters that minimize the two-parameter fit error (given the  $c_3$  guess and observed sensor data). These values are used as the initial guess for our curve-fitting algorithm.

It turns out that this method actually produces an initial guess that, while often quite different from the final algorithm output, is usually sufficient as an algorithm seed. The fact that the final estimate is often very different from the initial guess is encouraging; if the output were always very similar to the initial guess, this would seem to indicate that the error function was filled with local minima, and attempts to descend any large-scale error gradient were being confounded by small-scale effects. Scheffé's method is implemented as the front end of the curve-fitting algorithm; it provides an initial guess to the Newton-Raphson iteration.

#### 4.3.4.7 Local averaging

Section 4.2 outlined the decision to work with log data. This decision was motivated by the fact that taking a logarithm usually results in more well-behaved and linear pixel integration data. Indeed, log data does in fact behave very well under most cases. However, in some cases even log data is poorly behaved. When SNR is low, log data becomes jagged (just as non-log data does). Jagged curves generally do not elicit good performance from curve-fitting algorithms [28]. The Newton-Raphson iteration, for instance, typically diverges when faced with noisy sensor data with a Poisson mean on the order of 30 or 40 counts/frame. If at all possible, we would like to find a way to smooth out the observed data in cases like these.



One way to do this is local averaging. Local averaging is, in effect, low-pass filtering that tends to smooth out sharp transitions in the data. Local averaging makes physical sense in this case. We would like to reduce noise effects so that the sensor output values look more like the true Poisson mean values at each frame. At distant ranges (usually corresponding to low signal levels), the Poisson mean is usually nearly constant from frame to frame. (This is because at distant range the sensor output growth is slow.) The weak law of large numbers tells us that the mean of successive observations of the same random variable converges to that variable's mean [9]. Consecutive sensor output values are essentially observations of the same random variable, due to the nearly constant Poisson mean among these frames. Thus by performing local averaging on sensor data before feeding it to the algorithm, we are approximating the true Poisson mean of the data and enhancing algorithm performance.

Practically speaking, local averaging should only be done when it is beneficial; that is, when the SNR is below some certain threshold. Local averaging at high sensor output levels, which typically correspond to close ranges where the sensor output is changing quickly, is not only unnecessary but also detrimental. Local averaging is, in effect, low-pass filtering. It will tend to flatten out steep trends in the data. This is undesirable. Thus we should identify a threshold above which local averaging will not be performed. This threshold was chosen to be 50 counts/frame, which is commensurate with the observation in section 4.3.4.1 that the Newton-Raphson method exhibits convergence problems at low  $U(t)$  values.

## 4.4 Summary

The development of the curve-fitting algorithm is based on the assumption that it is possible to approximate direct radiation by constricting FOV about AOA on a high-resolution sensor. The form of direct radiation growth is known, and the sensor data obtained by constricting FOV about AOA should mimic this growth. By a simple manipulation, the direct radiation growth function can be rewritten to have an explicit dependence on TTI; this explicit dependence makes a curve-fit possible. The Newton-Raphson method, an iterative procedure for solving nonlinear equations, can produce an estimate of the physical parameters that generated the observed sensor data; TTI is one of these parameters. Various modifications to the data and the Newton-Raphson method improve the curve-fitting algorithm performance.

## 4.5 Analysis

The curve-fitting algorithm was tested on the same test set as described in section 3.6.1. 100 runs were performed for each case in the test set. The plots used to analyze performance of the curve-fitting algorithm are structured the same way as those used in the previous section. It is important, while viewing the plots in this section, to resist the temptation to make comparisons with the corresponding plots from the parameter-matching algorithm. Any comparison without examination of the contexts of the two algorithms may be misleading. Comparisons will be made in chapter 5.

There is no guarantee that any given run of the curve-fitting algorithm will result in a convergent estimate. We can examine performance of the curve-fitting algorithm by producing mean TTI estimate and rms error plots for the estimates that do converge, and noting the fraction of estimates that did not converge. Convergence is an important consideration when judging performance of the curve-fitting algorithm; an algorithm that produces an excellent estimate 10 percent of the time but diverges 90 percent of the time is of little use.

Table 4.1 lists the fraction of estimates that converged for each velocity-radiant intensity pair. Note the trends in this table: convergence is strongly dependent on both velocity and source radiant intensity. A slow or dim source is much less likely to produce a convergent estimate than a fast or bright source. Recall from section 4.3.4.1 that Newton-Raphson convergence is dependent on a high SNR. This explains the radiant intensity dependence: SNR is heavily influenced by source radiant intensity. The velocity dependence can be explained in the same way as the parameter-matching algorithm's rms error velocity dependence: a high-velocity source produces a sensor output growth function with a much more prominent trend. A low-velocity source produces a flatter curve that tends to get bogged down in the noise. Another way to think of this is in terms of the relationship between elapsed time and sensor output over a fixed observation interval. For a flat curve, there is little dependence of output upon time; the output is nearly constant from the beginning of the data to the end of the data. This minor dependence becomes very difficult to perceive in the presence of noise. A steep curve, on the other hand, implies a marked dependence of sensor output on time. The Newton-Raphson iteration is much more likely to notice this dependence in the same noisy environment.

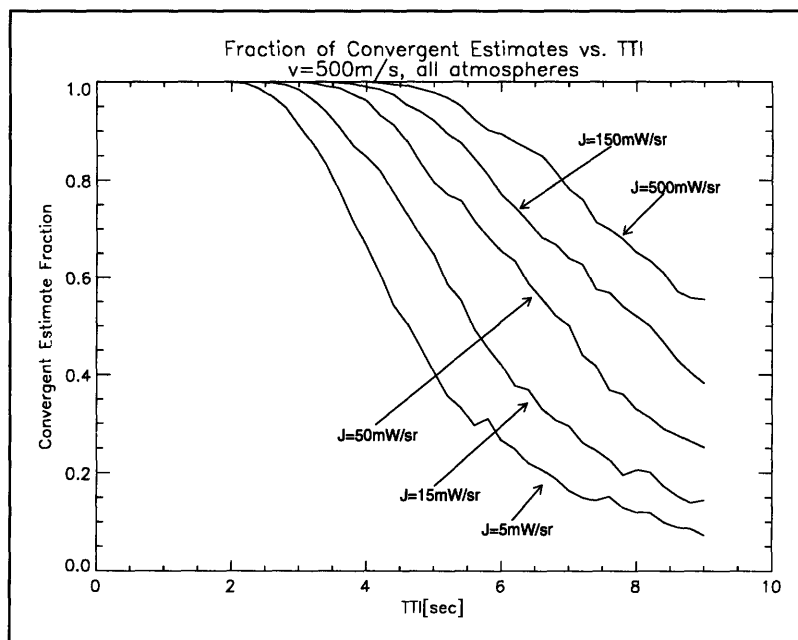
**TABLE 4.1: Curve-fitting algorithm: convergence percentage**

	250 m/s	500 m/s	750 m/s	1000 m/s
5 mW/sr	40.5	55.4	63.3	67.1
15 mW/sr	48.1	65.1	73.3	76.8
50 mW/sr	57.0	75.5	82.6	85.5
150 mW/sr	65.6	83.1	89.0	91.0
500 mW/sr	74.7	89.3	93.3	95.1

The convergence problem is definitely significant. The convergence percentage of 40.5% for velocity 250 m/s and source radiant intensity 5 mW/sr is dismal. The only good thing that can be said about this is that at such a low closing velocity, TTI is high and thus there might be time to perform an algorithm run on new data if the initial run did not converge. Indeed, if we look at convergence percentage vs. TTI, we see a slightly more promising picture. Figure 4.10 depicts convergence percentage vs. TTI (averaged over all atmospheres) for a 500 m/s source of different radiant intensities. Note that with a TTI of 2 seconds ( $r = 1000$  m), the algorithm converges almost all of the time for any source radi-

ant intensity. With a TTI of 4 seconds ( $r = 2000$  m), the algorithm converges approximately two-thirds of the time for a very dim source (radiant intensity of 5 mW/sr), and over 80% of the time for all other sources. Thus a high enough SNR will generally ensure convergence. This is supported by examination of the convergence percentage for different ozone concentrations; as ozone concentration decreases (and SNR increases), the fraction of convergent estimates increases.

We shall return to the issue of convergence. In the meantime, let us examine the behavior of the convergent estimates. Figure 4.11 contains plots depicting mean TTI estimate vs. TTI for each of the four test set velocities. This figure, and those to follow, is organized identically to the corresponding figures in chapter 3. The lightly dotted line represents a zero bias. The traces are average values over all 27 atmospheres.



**FIGURE 4.10: Fraction of convergent TTI estimates vs. TTI, different source radiant intensities**

Every trace in the plots shown in figure 4.11 shows a near-zero bias for low TTI. Above a certain threshold, however, bias significantly worsens. The mean TTI estimate no longer tracks TTI above this threshold; instead, it remains nearly constant. Let us call the point in each curve where this transition occurs the “knee” of the curve. Bias is small below the knee; bias increases sharply above each curve’s knee.

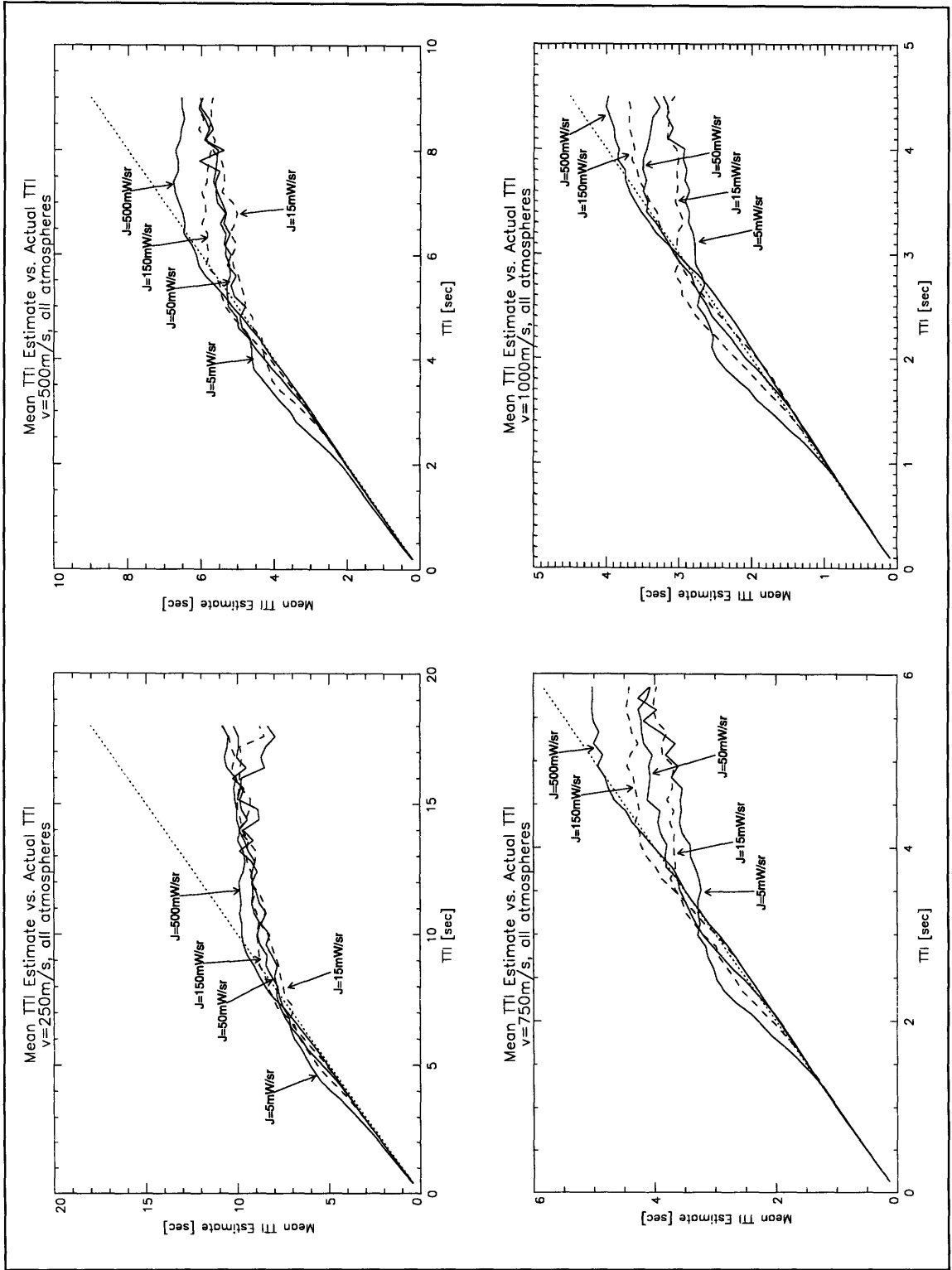


FIGURE 4.11: Curve-fitting algorithm: mean TTI estimate vs. TTI

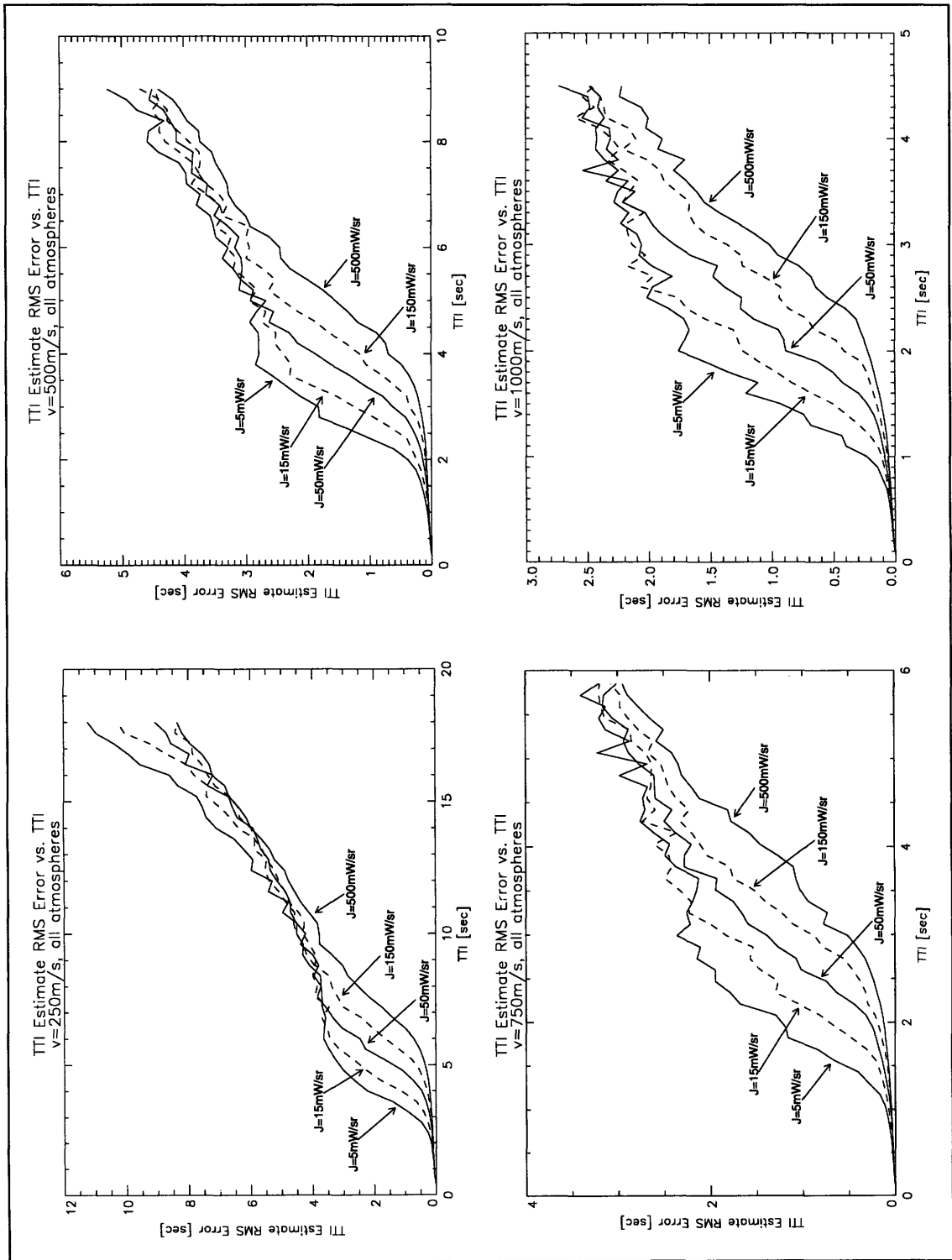
Source radiant intensity moves the knee in the curve out to a higher TTI. This effect is quite pronounced. As an example, let us examine the plot for closing velocity 750 m/s. The knee in the 5 mW/sr trace is approximately at  $TTI = 2.5$  s. The knee for the 15, 50, and 150 mW/sr traces fall around 3, 3.5 and 4 s, respectively; finally, the knee for the 500 mW/sr trace occurs at TTI close to 5 s. (Comparison to figure 4.8 shows that these values are roughly coincident with the range at which the signal drops below 50 counts/frame.) The behavior in the other plots is similar: higher source radiant intensity guarantees better mean TTI estimate performance at a more distant range. Radiant intensity not only improves algorithm convergence, but also reduces bias in the convergent runs.

There is a simple explanation for the presence of the knees in the bias curves. Recall that the Newton-Raphson method is more likely to produce a convergent estimate for sensor data in which there is a perceived trend. Sensor data that is roughly flat will not fit well to a sensor output growth curve unless it is nearly noiseless. Note that at high TTI, the sensor output growth curve is typically very flat and of relatively low magnitude. These are the worst possible conditions for TTI estimation by the curve-fitting algorithm. For flat, low-magnitude data, the Newton-Raphson iteration generally does not converge.

Let us consider a case in which noise affects a flat, low-magnitude curve so that it appears steeper, as if the source were approaching at closer range. This could result in convergence to a TTI estimate that is lower than actual TTI. Thus at distant range, the curve-fitting algorithm generally either cannot converge because the input data is too jagged, or is “fooled” into convergence to a TTI estimate that is too low. Thus the mean convergent estimate will be lower than actual TTI. (This phenomenon is quite similar to the distribution clipping described in conjunction with the parameter-matching algorithm.)

Another characteristic shared by all the traces is a slight but marked deviation above zero bias just before the knee of the curve, most noticeable in the traces corresponding to dim sources. This observation also has a simple explanation. In a noisy environment, the spread of the algorithm’s TTI estimates is relatively wide. The maximum TTI estimate is unbounded. The minimum TTI estimate, however, is 0. This leads to asymmetry in the spread of the estimates and tends to drive up the mean estimate in a low-SNR environment. This is what is happening near the knees of the traces, when SNR is low. (Again, note the similarity to distribution clipping.) Comparing the four plots it is also apparent that, once again, mean TTI estimate tracks actual TTI much better at a high velocity.

Recall that ozone concentration had a profound effect on the bias curves for the parameter-matching algorithm. Ozone concentration also affects curve-fitting algorithm results, though the effect is much less noticeable. Increasing ozone concentration has two general effects on a bias curve. First of all, it moves the knee in the curve to a closer range. Second, it worsens the bias above the knee of the curve. However, the bias above the knee of the curve is high even for a low ozone concentration atmosphere, and thus excluding high ozone concentration atmospheres from the averaging does not make the traces in figure 4.11 look qualitatively different. Atmospheric effects on curve-fitting algorithm performance will be examined later in this section; for now, we note that high ozone concentration atmospheres result in generally higher bias and rms error, but do not exhibit the kind of outlying behavior we saw in conjunction with the parameter-matching algorithm.



**FIGURE 4.12: Curve-fitting algorithm: TTI estimate rms error vs. TTI**

Let us now examine the rms error characteristics of the curve-fitting algorithm, plotted in figure 4.12. These plots clearly show that rms error is highly dependent upon source radiant intensity. A brighter source results in a lower rms error. This is true at every velocity: all four plots show a clear dependence on radiant intensity. Again, this is due to the higher signal level caused by a brighter source.

Each rms error plot can be broken into three general regions. At very low TTI, rms error is nearly zero for any radiant intensity. For larger TTI values, the traces are widely separated. Finally, at a high TTI, the traces are again roughly coincident.

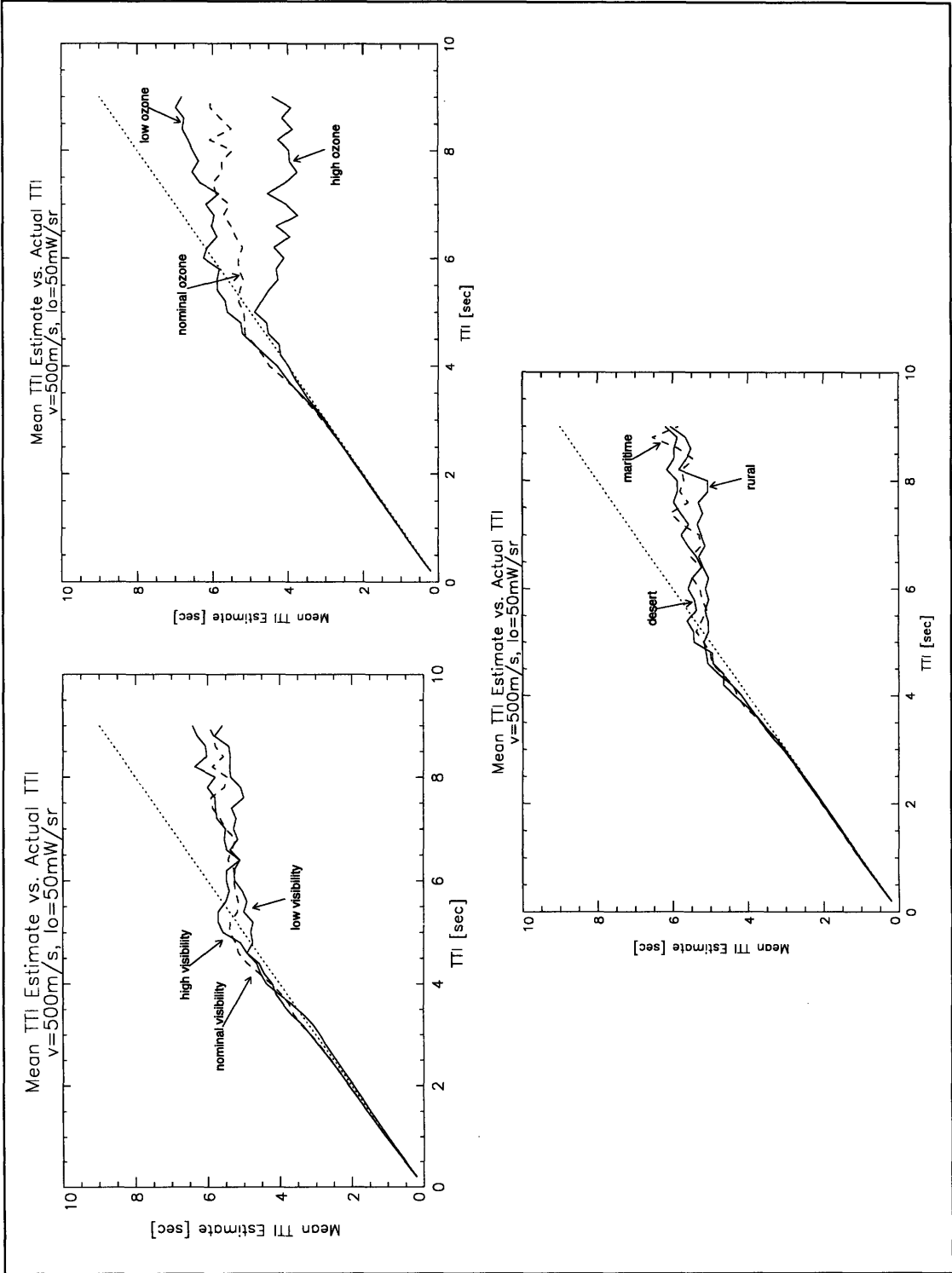
At very close range, SNR will generally be high and the sensor output growth curve will generally be steep. This is true for a source of any radiant intensity and velocity. Thus at close range we would expect the algorithm to perform well for any velocity-radiant intensity combination in the test set. This accounts for the near-zero rms error shown by all traces at low TTI. As range increases, we begin to notice significant differences in algorithm performance for sources of different brightness. At a given range, a dim source will result in a much lower SNR than a bright source. Since we have already observed that algorithm performance depends strongly on SNR, it makes sense that as range increases we should notice a bright source resulting in a lower rms error than a dim source.

At distant range, SNR will generally be low for a source of any radiant intensity. We observed that the knees in the curves in figure 4.11 are the result of operation in a low SNR environment. Above the knees in the mean TTI estimate curves in figure 4.11, rms error is primarily driven by the large estimate bias. Comparison of figures 4.11 and 4.12 shows that the convergence of the traces in each rms error plot is in fact roughly coincident with the knee in the traces in each bias plot. Essentially, above the knee in the bias curve, any TTI estimate produced by the algorithm is of relatively little value.

Just as the bias is lower at higher closing velocities, rms error is lower under the same conditions. This only serves to reaffirm the point that steeper data greatly enhances algorithm performance. High ozone concentration atmospheres again result in higher rms errors (as will be discussed below) but do not have nearly as significant of an impact as was seen in the parameter-matching algorithm analysis.

Figures 4.13 and 4.14 display mean TTI estimate and rms error performance, respectively, for the curve-fitting algorithm in different atmospheric conditions. Once again, these plots are for velocity 500 m/s and source radiant intensity 50 mW/sr only; the results in these figures generalize to other velocity-intensity pairs.

Figure 4.13 shows that mean TTI estimate performance is nearly independent of atmospheric arena and visibility but somewhat more dependent on ozone concentration. Atmospheric arena and visibility barely affect the estimate bias. The traces for all choices of atmospheric arena or all choices of visibility are all roughly identical. These curves are, in fact, nearly identical to the 500 m/s-50 mW/sr trace in figure 4.11. Thus it appears that changes in scattering do not have a significant effect on curve-fitting algorithm performance.



**FIGURE 4.13: Curve-fitting algorithm: mean TTI estimate vs. TTI, atmosphere subsets (v = 500m/s, J = 50mW/sr)**

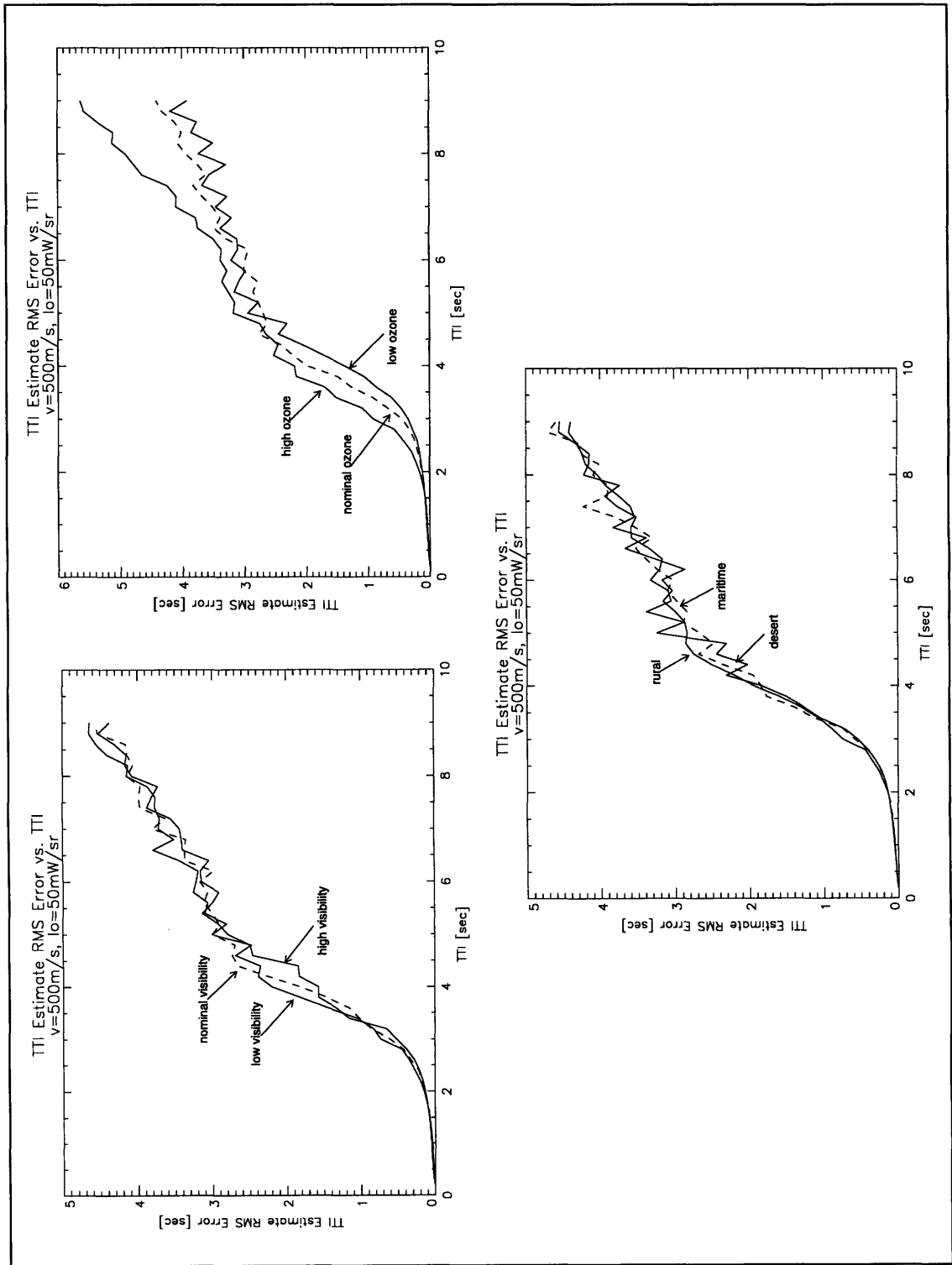


Ozone concentration does have an effect on mean estimate, with higher ozone concentration resulting in a knee at a lower TTI. This is attributable to the decrease in SNR in a high-absorption atmosphere. Ozone concentration is the primary contributor to absorption in the UV regime. Thus we see performance that is better in atmospheres with a lower concentration of ozone. Increasing ozone concentration is similar to decreasing source radiant intensity, in that either of these changes decreases SNR and moves the knee of the curve to a closer range. Note that below the knee, each curve exhibits relatively low bias. In fact, the high ozone concentration atmosphere subset results in the lowest below-knee bias. This is due to the fact that absorption is exponentially dependent on range. A higher ozone concentration results in a higher absorption coefficient  $k_a$ , and thus a steeper sensor output growth curve at close ranges (despite a lower signal). So at close range, when SNR is relatively high, a high ozone concentration actually helps the curve-fitting algorithm bias (by creating a steeper sensor output growth curve) more than it hinders it (by creating a lower signal).

The rms error plots in figure 4.14 paint a familiar picture. We see that rms error is nearly independent of variations in atmospheric arena, with all three choices resulting in nearly coincident traces. Visibility has a slight impact on rms error. Close inspection shows that below a TTI of approximately 4.5 s (where bias begins to dominate rms error) the high-visibility atmosphere subset generally has the lowest rms error and the low-visibility atmosphere subset has the highest. Predictably, ozone concentration has the most profound effect on rms error. Indeed, the ozone concentration plot in figure 4.14 is strikingly similar to any of the rms plots in figure 4.12: these plots exhibit very nearly the same shape and trends. The trend in the ozone concentration plot in figure 4.14 clearly links low ozone concentration with low rms error. Thus the influence of ozone concentration in determining algorithm performance is once again underscored. It is not clear why the high ozone concentration atmospheres have much less of a relative impact on rms error than they did in the parameter-matching algorithm. The difference in rms error between the high ozone concentration atmospheric subset and the nominal ozone concentration atmospheric subset is slightly greater than that between the nominal ozone concentration subset and the low ozone concentration subset, but not to the extent that was observed in the previous chapter.

Figures 4.13 and 4.14 imply that the direct radiation approximation achieved by narrowing FOV is roughly valid in any atmosphere. Changes in atmospheric conditions have a relatively mild effect on curve-fitting algorithm performance, except in the cases where absorption is affected. Different scattering environments do not tend to affect performance differently. Ozone concentration has a marked effect on performance due to its impact on SNR; the specific scattering environment has almost no noticeable effects.

The results presented in this section indicate that the bias and rms error performance of our curve-fitting algorithm is superior to that of previous efforts cited in section 2.7. However, the curve-fitting algorithm's convergence problem casts significant doubt on its applicability. We postpone discussion of whether the curve-fitting algorithm could be implemented in a real system until the next chapter.



**FIGURE 4.14: Curve-fitting algorithm: TTI estimate rms error vs. TTI, atmosphere subsets ( $v = 500\text{m/s}$ ,  $J = 50\text{mW/sr}$ )**

## 4.6 Curve-Fitting Algorithm Enhancements and Modifications

One significant problem with the curve-fitting algorithm is its possibility of non-convergence. This problem has been discussed in previous sections. There are a several ways this problem might be sidestepped if not corrected.

Most non-convergence problems occur at low SNR. One possible way to work around the non-convergence problem would be to wait to begin the Newton-Raphson iteration until sensor data is of sufficiently high magnitude. This would reduce the chance of a divergent iteration. Obviously, the drawback to this approach is that with every successive sensor data frame, the clock is ticking. By waiting to begin algorithm computation, there is a greater chance that the aircraft will not survive to see the end of computation.

A better solution to this problem would be to provide better indications as to whether a given iteration is likely to converge. By identifying a non-convergent iteration as quickly as possible, wasted time is minimized. If a non-convergent iteration is identified, it can be aborted immediately and a new iteration, using the most recent two seconds of sensor data, can commence. In this way, the chances of obtaining a convergent estimate in time to initiate countermeasures are maximized.

Another modification to the curve-fitting algorithm could provide a metric to judge the reliability of a given TTI estimate. In a simulation, we can calculate mean error and rms error over repeated runs because we know the true value of TTI; in a real-world system, the only information available aside from the sensor data is the TTI estimate and the associated curve-fitting error. We can imagine building a library consisting of TTI estimates and their associated curve-fitting errors and their mapping to rms TTI estimate error. In this way, we could look up any TTI estimate/curve-fitting error pair we might encounter in the real world in order to see how the algorithm fared under similar circumstances in the past. In this way we could provide the user with some kind of a “confidence” measure—an indication of how much faith to put in the current estimate based on previous similar estimates.

## **5.0 CONCLUSIONS**

The past four chapters have described the development and analysis of two TTI estimation algorithms. This chapter restates the main points presented in the past chapters, compares the relative performance of each algorithm to past work and then to each other, and attempts to give a sense of each algorithm's absolute utility. This chapter concludes with a discussion of some possibilities for future work in the area of TTI estimation.

### **5.1 Algorithm Summary**

The high resolution of our sensor enabled us to develop two algorithms based on previously unavailable information. Specifically, the high sensor resolution provides small-angle scattering data and also the means for approximating the direct radiation contribution to the sensor output. The parameter-matching algorithm utilizes the small-angle scattering data; the curve-fitting algorithm relies on the direct radiation approximation.

#### **5.1.1 The parameter-matching algorithm**

The parameter-matching algorithm is based on the assumption that the small-angle scattered data provides information about TTI. This algorithm operates by comparing a parametrized observed (noisy) sensor output vector to a canonical set of parametrized noiseless sensor output vectors. The parametrization process is essentially a coarse spatial subsampling of the sensor output, and reduces the sensor output vector to a reasonable size. The TTI estimate is obtained by minimizing angular deviation, which is equivalent to minimizing squared error. The canonical set traversal is performed using a coarse-to-fine search with local abort criteria to reduce the set traversal time.

#### **5.1.2 The curve-fitting algorithm**

The curve-fitting algorithm is based on the assumption that by integrating pixel outputs within a narrow window centered on AOA it is possible to obtain a good approximation to the direct radiation component of the sensor output. The curve-fitting algorithm fits this integrated pixel data to a function dependent on three parameters, one of which is TTI. The curve-fitting algorithm uses an iterative procedure known as the Newton-Raphson method to generate an estimate of the three curve parameters, thus producing a TTI estimate.

### **5.2 Comparison of Algorithm Performance and Applicability**

Comparison of the mean TTI estimate plots in figures 3.8 and 4.11, and the rms error plots in figures 3.11 and 4.12, implies that the parameter-matching algorithm is much more successful at estimating TTI than the curve-fitting algorithm. Not only does the parameter-matching algorithm produce a generally lower-bias estimate, it also produces an estimate with lower rms error for any set of physical parameters. In addition, the parameter-matching algorithm always produces an estimate, whereas under some conditions the curve-fit-

ting algorithm will not converge. Thus it appears the parameter-matching algorithm outperforms the curve-fitting algorithm in all tested areas. We will undertake a more complete comparison of the algorithms (including a consideration of the context in which each algorithm was tested) in section 5.2.3. First, however, we will compare the performance of our algorithms to the performance of past efforts discussed in section 2.7 in order to gauge our algorithms' success and applicability.

### 5.2.1 The parameter-matching algorithm

The parameter-matching algorithm's performance is discussed in detail in section 3.6. Specifically, the plots in figures 3.8 and 3.11 characterize the bias and rms error of the parameter-matching algorithm. Viewed in direct comparison with the results of any of the past efforts cited in section 2.7, these results are far superior.

Previous curve-fitting work [14], as described in section 2.7.1, performs significantly worse than the parameter-matching algorithm developed herein. The testing of the original curve-fitting algorithm was done with sources closing at 500 m/s. As described previously, the original curve-fitting attempt resulted in rms error approximately equal to TTI except for the brightest sources. In comparison, the parameter-matching algorithm produces an rms error of approximately 0.3 s for a 50 mW/sr source closing at 500 m/s from 2000 m (TTI = 4 s). The same source closing from 1000 m (TTI = 2 s) gives a 0.1 s rms error. Our values are much better than these prior curve-fitting results.

Our parameter-matching algorithm results are also better than the EKF results posted by Luetgen [15]. For a moderately bright source closing at 250 m/s from a range of 2000 m (TTI = 8 s), Luetgen reports an rms error of approximately 6.0 s; for a range of 1000 m (TTI = 4 s), he reports an rms error of approximately 2.0 s. In comparison, the parameter-matching algorithm produces an rms error of approximately 0.7 s at range 2000 m and 0.4 s at range 1000 m under similar conditions. Again, this is a marked improvement.

The comparison of the parameter-matching algorithm's results to Andera's results [16] is especially illuminating, since Andera's algorithm is similar in concept to the parameter-matching algorithm. (Andera's algorithm was also tested over a subset of its training set.) Andera reports an rms error of 4.3 s for a 50 mW/sr source closing from 3000 m at velocity 250 m/s (TTI = 12 s); for the same source closing from 1500 m (TTI = 6 s), his rms error is 3.7 s. In comparison, the parameter-matching algorithm produces rms errors of 1.5 s and 0.6 s under the same conditions. For a 50 mW/sr source closing from 3000 m at velocity 750 m/s (TTI = 4 s), Andera reports an rms error of 2.2 s; for the same source closing from 1500 m (TTI = 2 s), 1.0 s. In comparison, our algorithm's rms error is approximately 0.7 s and 0.1 s under similar conditions.

The above comparisons show that our parameter-matching algorithm is more successful than any cited previous TTI estimation effort. Admittedly, the parameter-matching results were obtained by testing under artificial circumstances—the algorithm's test set was a subset of its training set. Thus the performance of the algorithm over this test set may not exactly reflect the performance of the algorithm in all real-world circumstances. How-

ever, this training set was chosen to be representative of the broad range of possible conditions in which the algorithm might operate. These results thus provide at least an indication that our algorithm is capable of significantly better performance than that seen before.

### 5.2.2 The curve-fitting algorithm

The performance of our curve-fitting algorithm is discussed in detail in section 4.5. Figure 4.11 depicts the bias performance of the curve-fitting algorithm; figure 4.12 depicts the rms error performance. In assessing the relative performance of our curve-fitting algorithm, we can compare our rms error results to those of previous efforts, as we did for the parameter-matching algorithm in the previous section. However, a significant issue in evaluating our curve-fitting algorithm's performance is its possibility of non-convergence. This complicates direct comparisons to previous work. Andera's algorithm [16] always produces an estimate. Although Luetngen's EKF algorithm [15] may not produce a TTI estimate that converges as elapsed time approaches infinity, a TTI estimate is returned for every algorithm run. The possibility of non-convergence is presumably an issue in the original curve-fitting effort [14], but no mention at all is made of convergence in [14]. Thus we are reduced to stating that our curve-fitting algorithm produces no estimate roughly one quarter of the time (see table 4.1). With this in consideration, we will compare its rms errors to past results.

Our curve-fitting algorithm performs significantly better than the previous curve-fitting algorithm [14]. Recall that the original algorithm's TTI estimate rms error was nearly as large as TTI itself. In contrast, our algorithm produces an rms error of 1.9 s for a 50 mW/sr source closing at 500 m/s from 2000 m (TTI = 4 s) and an rms error of 0.2 s for the same source approaching from 1000 m (TTI = 2 s). This is markedly better.

Our curve-fitting algorithm also outperforms Luetngen's EKF algorithm [15]. As mentioned in the previous section, his algorithm produced estimates with rms errors of 6.0 s and 2.0 s for sources closing at 250 m/s from 2000 m (TTI = 8 s) and 1000 m (TTI = 4 s). Our algorithm produces rms errors of 3.8 s and 0.6 s under these respective conditions. Note that the scale of the performance gap is most noticeable at close range.

The curve-fitting algorithm performs slightly worse than Andera's algorithm [16] in a low-SNR environment (distant range) but significantly better in a high-SNR environment (close range). Andera reports rms errors of 4.3 s and 3.7 s for a 50 mW/sr source approaching at 250 m/s from 3000 m (TTI = 12 s) and 1500 m (TTI = 6 s), whereas our algorithm produces rms errors of 5.0 s and 2.2 s under identical conditions. Andera also reports rms errors of 2.2 s and 1.0 s for 50 mW/sr sources approaching at 750 m/s from 3000 m (TTI = 4 s) and 1500 m (TTI = 2 s), respectively; our algorithm produces rms errors of 2.2 s and 0.3 s under the same circumstances.

The range of circumstances under which our curve-fitting algorithm was tested is more broad than that used for the previous algorithms. We see that our curve-fitting algorithm's rms error (for convergent estimates) is superior to that of previous efforts in a high-SNR

environment. In a low-SNR environment, the superiority of the curve-fitting algorithm is less marked (it is outperformed by Andera's algorithm), and any algorithm estimate produced by our curve-fitting algorithm at a very low SNR is essentially useless due to high bias and rms error.

When it results in a convergent estimate, our curve-fitting algorithm generally produces a lower rms error than that seen in previous work. It is unclear how we should rate this accomplishment. In the real-world, a TTI estimation algorithm is a single-failure system. Even a zero bias and rms error over convergent estimates cannot save the pilot who finds himself without a TTI estimate due to non-convergence.

### 5.2.3 Comparison of the two algorithms

A direct comparison of the results from the previous two sections implies that the performance of the parameter-matching algorithm is superior to that of the curve-fitting algorithm. Before we can make a real judgment, however, we must examine the circumstances under which each algorithm's results were obtained.

The parameter-matching algorithm was tested on an a semi-complete subset of its canonical set. In other words, its testing set was contained within its training set. This was due both to an admitted lack of foresight and to a lack of time and computational resources. We would thus expect the results cited in chapter 3 to be, in some sense, optimal. For instance, we would expect the rms error results shown in figure 3.11 to provide a lower bound on the rms error of the parameter-matching algorithm in a wider range of circumstances.

It is possible that the parameter-matching algorithm would perform significantly worse in a non-canonical set atmosphere. Adjacent range or velocity points in the canonical set are generally separated (in an angular deviation sense, as defined by equation 3.6) by a fraction of a degree. (Figure 3.7 demonstrates this.) Adjacent atmosphere points, however, are typically separated by an angle of several degrees. Thus there is a relatively wide "gap" between atmospheres in the canonical set. An engagement taking place in an atmosphere "between" two canonical set entries would probably result in some degradation of performance—though to what extent is uncertain.

We can make some qualitative comparisons between the parameter-matching algorithm performance and the curve-fitting algorithm performance. At distant ranges, the cited parameter-matching rms error is significantly better than that of the curve-fitting algorithm. At close ranges, however, a good deal of this advantage evaporates. (A direct comparison of figures 3.11 and 4.12 bears this out.) As SNR increases, both algorithms perform better, but the curve-fitting algorithm performance improves more markedly. Thus the apparent superiority of the parameter-matching algorithm might be suspect in high-SNR conditions.

Recall that the parameter-matching algorithm's data reduction scheme provides it with higher SNR data than that used by the curve-fitting algorithm. The parameter-matching algorithm uses a frame rate of 10 Hz; the curve-fitting algorithm uses a frame rate of

100 Hz. (The reasons for these choices are outlined in sections 3.1.3 and 4.3.4.4.) It is possible that the parameter-matching algorithm's superior performance may be due in part to this disparity in SNR. If the curve-fitting algorithm were modified to operate on 10 Hz data, its performance might improve.

We can directly compare the execution time of the algorithms. On a DEC Alpha workstation, the parameter-matching algorithm required, on average, 1.0 s of central processing unit (CPU) time to produce an estimate. In comparison, the curve-fitting algorithm took, on average, 0.3 s of CPU time to produce an estimate or abort due to non-convergence. Thus the curve-fitting algorithm will generally result in a lower-latency TTI estimate (or non-convergence declaration). This is a paramount concern in a real-world system.

It seems clear that even considering concerns about its testing conditions, and despite its longer execution time, the parameter-matching algorithm is the superior algorithm. It produces a TTI estimate with a lower rms error than that of the curve-fitting algorithm in all circumstances. It is guaranteed to produce an estimate. These observations outweigh the concerns.

The performance figures reported in section 3.6 are much better than those produced by any previous algorithm. Any TTI estimation system that produced such low rms errors would definitely be an asset to the pilot. A rule-of-thumb for gauging the success of a TTI estimation algorithm is that it should produce an estimate with an rms error under 20% of TTI. The parameter-matching algorithm meets this criterion except for dim sources at distant range (see figure 3.8) or dim sources in high ozone concentration atmospheres (see figure 3.14). The parameter-matching algorithm results are definitely good enough to provide a useful and effective TTI estimation system. We will discuss real-world implementation issues in the next section.

The utility of our curve-fitting algorithm in a real system is doubtful. Despite the fact that rms errors for convergent estimates are lower than those of past efforts, the possibility of non-convergence is a great liability. A system that cannot produce an estimate under some conditions is of questionable value. In the real world, the cost of a single divergent estimate could be the survival of the aircraft and pilot. Thus the performance reported for the curve-fitting algorithm indicates that it might not be useful in a real-world setting.

### **5.3 Implementability**

As indicated in section 1.3, an issue of prime importance when considering the implementability of a TTI estimation algorithm is computational burden. Any TTI estimation system on board a military aircraft must be as computationally simple as possible in order to provide a TTI estimate with minimum computation and with minimum latency.

In their final forms, the parameter-matching algorithm's computational latency is approximately three times as great as that of the curve-fitting algorithm. In a real-world system, this discrepancy would probably be even greater. In a real system, the parameter-matching algorithm's canonical set would need to be expanded to include not only azimuth and



elevation dimensions, but possibly a broader and finer atmospheric sampling. Thus the canonical set of a real system could easily be hundreds or thousands of times greater than the current canonical set. This would result not only in an increase in the storage requirements of the system, but also an increase in the set traversal time. It is unlikely that an algorithm based on a canonical set even a hundred times larger than ours could be implemented in real-time.

It is true that either algorithm could be modified in order to reduce computational expense and latency. For instance, the coarse-to-fine search could be made coarser, or a new and better search method (possibly gradient descent) could be adopted. Alternately, the construction of the canonical set could be re-examined. It is possible that the canonical set could be made sparser with the implementation of some kind of an interpolation scheme to retain high resolution. In any case, it seems unlikely that a parameter-matching algorithm based on a canonical set as finely resolved as the present one could be implemented in a real system. The excellent performance of our parameter-matching algorithm, however, indicates that the parameter-matching concept is promising, and that canonical set design alternatives should be examined.

The curve-fitting algorithm could be implemented in a real system almost in its current incarnation. Section 4.6 contains several suggestions for working around the non-convergence problem. The operation of the algorithm could be made faster (at the possible expense of accuracy) by adjusting the termination tolerance or data window length. A real system would need to include provisions for the possibility of a non-zero elevation and the resulting variable extinction coefficient  $k$  over the radiation path. This innovation could possibly be achieved without significant conceptual changes to the algorithm.

One drawback to the parameter-matching algorithm is that it depends explicitly on a set of assumed atmospheric conditions. In an anomalous atmosphere, the performance of the parameter-matching algorithm will always be uncertain. The curve-fitting algorithm, on the other hand, depends on the more general direct radiation growth assumption. We know that equation 4.7 will govern the contribution of direct radiation to sensor output in any homogenous atmosphere, and the results of figure 4.14 indicate that the approximation to direct radiation by narrowing FOV is roughly equally valid in a wide range of atmospheric conditions. Thus the curve-fitting algorithm has the conceptual advantage of validity in the general case, not simply a set of discrete cases.

## 5.4 The Real World

We made a number of simplifying assumptions in section 2.6 in order to simplify algorithm development and testing. In the real world, these assumptions might not hold. Before a real system is developed these assumptions must be re-examined.

Several of the assumptions in section 2.6 are likely to be valid in almost all real-world engagements. Due to proportional navigation, a missile will usually approach its target along a linear path at a constant velocity. In the cases where the approach changes throughout an engagement (i.e., when AOA is non-constant) it would be relatively simple

to track the missile in order to acquire the integrated pixel data at each frame. (It should be noted that a change in missile velocity might be difficult to detect and to compensate for.)

The assumption of homogenous atmospheric conditions is also reasonably safe, given an aircraft in level flight. It is unlikely that local atmospheric conditions would change significantly over the course of a typical engagement. Of course, if the aircraft were near a cloud boundary, for instance, this assumption might be violated. Also, if the aircraft were not in fact in level flight, but diving or climbing, then local atmospheric conditions would change during an engagement.

Two assumptions in particular are unlikely to hold in the real world. The first of these is the assumption that the missile's radiation characteristics are similar to those of a constant-temperature isotropic blackbody. Over the narrow spectral response bandwidth of our sensor, a missile's radiation characteristics might be similar to those of our modeled blackbody. In general, however, the assumptions of constant temperature and isotropic emission are invalid. This presents a significant problem to both algorithms, since both algorithms depend on the assumption of constant source spectral radiance throughout an engagement. Changes in source temperature, or in the attitude of a non-isotropic source, will result in changes in the amount of radiation received by the sensor. In general, temperature fluctuations are the bigger concern; proportional navigation and missile construction ensure that a missile's attitude will remain nearly constant throughout an engagement. Non-isotropy thus will not manifest itself extensively in the direct radiation. It would be more likely to cause a change in the relationship between direct and scattered radiation collected at the sensor. Sophisticated source models might be able to account for this possibility, and the canonical set of the parameter-matching algorithm could be reconstructed accordingly. The curve-fitting algorithm might require no modifications at all to handle a non-isotropic source; it is based on an approximation to the direct radiation received by the sensor, and this approximation will be roughly equally valid regardless of whether the source is isotropic.

Changes in source temperature might be difficult to accommodate. A change in source temperature (which could be caused by the burning of propellant to adjust attitude or trajectory) results in a change in source spectral radiance. If spectral radiance changes could be detected and compensated for, then the algorithms could operate without significant modification or degradation of performance. If such changes went undetected, however, the effect might be catastrophic.

The other typically invalid assumption is that of zero source azimuth and elevation. This assumption is completely artificial. In the real world, a missile may obviously approach from any direction. A non-zero AOA results in variable atmospheric conditions over the direct radiation path and a non-symmetrical sensor output. Any real system must be robust to non-zero AOA. As indicated in the previous section, each algorithm could be modified in order to accommodate this possibility. The canonical set of the parameter-matching algorithm could be expanded to include azimuth and elevation dimensions, and the curve-fitting algorithm could possibly be modified in order to allow non-constant

extinction coefficients. These or equivalent modifications are necessary in a real-world system.

## 5.5 Future Work

There are many avenues of TTI estimation that could be pursued. Some of these are suggested directly by the work done in this thesis; others are independent paths. This section briefly outlines a few of them.

One direction suggested by the evaluation of the parameter-matching algorithm is re-evaluation of the canonical set. The canonical set is both the basis for and the bane of the parameter-matching algorithm. Its existence allows the algorithm to produce a TTI estimate, but any fairly complete canonical set is prohibitively large. In order to implement a real parameter-matching algorithm, the canonical set would need to be redesigned and reduced in size. One way to do this is to develop a scheme for interpolating sensor outputs. If this were possible, then the canonical set resolution (and thus size) could be significantly reduced. The interpolation scheme could be used to obtain sensor outputs corresponding to physical parameter sets not explicitly included in the canonical set. Such an algorithm might be more complex, but this would more than likely be counteracted by the significant reduction in canonical set size.

The sensor output parametrization scheme used in the parameter-matching algorithm could be redeveloped. A frame-to-frame parametrization scheme might prove to be more compact and informative. An improvement in the parameterization scheme would likely result in an improvement in algorithm performance.

Another research direction immediately suggested by the parameter-matching algorithm is a re-evaluation of the canonical set search method. The coarse-to-fine search, while a great improvement over a simple sequential search, is probably far from optimal. Other search methods could be used to find an error minimum much more quickly. One of these, the gradient descent method, was mentioned in section 3.4.2. Such a modification could greatly reduce the computational latency of the parameter-matching algorithm.

The curve-fitting algorithm also suggests areas for future research. The most obvious of these is attempting to improve the likelihood of convergence. This area provides perhaps the greatest opportunity for improvement to the curve-fitting algorithm. Some suggestions in this direction were discussed in section 4.6.

There are numerous areas of potential research not touched upon in this thesis. One of these is a return to the EKF method used by Luetzgen to estimate TTI. This approach has the advantage of providing an ongoing estimate of TTI throughout an engagement. The increased resolution of an imaging sensor might pave the way for the success of this approach.

A general area of future research is making the algorithms presented herein, and others, more applicable in the real world. The ideal algorithm would be dependent on none of the

assumptions discussed in section 2.6, and able to operate in a wider range of circumstances.

One interesting and immediate direction of research suggested by this thesis is an attempt at developing a hybrid TTI estimation algorithm that utilizes aspects of both the parameter-matching and curve-fitting algorithms. A hybrid algorithm might use a coarse canonical set to provide a rough TTI estimate, which could then be used as an initial guess for a curve-fitting algorithm. An initial guess obtained in this manner might be sufficient to greatly increase the probability of convergence for the curve-fitting algorithm. This approach also has the advantage of utilizing a parameter-matching approach in the low-SNR portion of the data window (when parameter-matching performance is far superior to that of the curve-fitting algorithm), and then moving to a curve-fitting approach in the high-SNR portion (when performance of the two algorithms is more equivalent).

## References

1. J. S. Accetta and D. L. Shumaker, assoc. eds., The Infrared and Electro-Optical Systems Handbook, 8 vols. Vol. 7: Countermeasure Systems, ed. by D. H. Pollock. SPIE Optical Engineering Press, 1993.
2. J. S. Accetta and D. L. Shumaker, assoc. eds., The Infrared and Electro-Optical Systems Handbook, 8 vols. Vol. 4: Active Electro-Optical Systems, ed. by C. S. Fox. SPIE Optical Engineering Press, 1993.
3. J. S. Accetta and D. L. Shumaker, assoc. eds., The Infrared and Electro-Optical Systems Handbook, 8 vols. Vol. 5: Passive Electro-Optical Systems, ed. by S. B. Campana. SPIE Optical Engineering Press, 1993.
4. W. S. Ross, "Point-Source Optical Propagation in a Multiple Scattering Medium," Ph.D. Thesis, Dept. of EECS, M.I.T., Aug. 1980.
5. R. M. Gagliardi and S. Karp, Optical Communications. Wiley, 1976.
6. G. R. Elion, Electro-Optics Handbook, Electro-Optics Series, v. 3. Marcel Dekker, 1979.
7. L. J. Pinson, Electro-Optics. Wiley, 1985.
8. S. Karp, R. M. Gagliardi, S. E. Moran and L. B. Stotts, Optical Channels: Fibers, Clouds, Water, and the Atmosphere, Applications of Communications Theory Series. Plenum Press, 1988.
9. A. W. Drake, Fundamentals of Applied Probability Theory. McGraw-Hill, 1967.
10. H. Weichel, Laser Beam Propagation in the Atmosphere, Tutorial Texts in Optical Engineering Series, v. TT3. SPIE Optical Engineering Press, 1990.
11. W. S. Ross, W. Jaeger, J. Nakai, T. Nguyen and J. Shapiro, "Atmospheric Optical Propagation: An Integrated Approach," Optical Engineering Vol. 21, pp. 775-785 (1982).
12. C. F. Bohren, ed., Selected Papers on Scattering in the Atmosphere, SPIE Milestone Series, v. MS 7. SPIE Optical Engineering Press, 1989.
13. F. X. Kneizys, E. P. Shuttle, L. W. Abreau, J. H. Chetwynd, Jr., G. P. Anderson, W. O. Gallery, J. E. A. Selby and S. A. Clough, "Users Guide to LOWTRAN 7," Air Force Geophysical Laboratory Report AFGL-TR-88-0177, 1988.
14. S. L. Angle, "An Evaluation of the All-Stars Time-To-Intercept (TTI) Algorithm," LIRIS IR&D Report, Aug. 1990.
15. M. R. Luetgen, "Trajectory Estimation of an Optically Radiating Source," M.S. Thesis, Dept. of EECS, M.I.T., May 1990.
16. C. Andera, "Trajectory Estimation of an Optically Radiating Source," M.Eng. Thesis, Dept. of EECS, M.I.T., May 1995.
17. A. H. Jazwinsky, Stochastic Processes and Filtering Theory. Academic Press, 1970.

18. D. B. Backer, Guaranteed Properties of the Extended Kalman Filter. M.I.T., Laboratory for Information and Decision Systems, 1987.
19. E. B. Saff and A. D. Snider, Fundamentals of Complex Analysis for Mathematics, Science, and Engineering. Prentice Hall, 1976.
20. M. R. Luetgen, Optical Flow Computation via Multiscale Regularization. M.I.T., Laboratory for Information and Decision Systems, 1992.
21. K. S. Shanmugan and A. M. Breipohl, Random Signals: Detection, Estimation, and Data Analysis. Wiley, 1988.
22. W. Molenaar, Approximations to the Poisson, Binomial and Hypergeometric Distribution Functions. Mathematisch Centrum, 1970.
23. J. S. Lim, Two-Dimensional Signal and Image Processing. Prentice Hall, 1990.
24. L. W. Johnson and R. D. Ross, Numerical Analysis. Addison-Wesley, 1982.
25. G. C. Holst, Electro-Optical Imaging System Performance. JCD Publishing and SPIE Optical Engineering Press, 1995.
26. S. Yakowitz and F. Szidarovsky, An Introduction to Numerical Computations. Collier Macmillan, 1986.
27. M. Scheffé, "Time-To-Intercept (TTI) Algorithm Experimentation," LIRIS IR&D report, June 1987.
28. W. H. Press, B. P. Flannery, S. A. Teukolsky and W. T. Vetterling, Numerical Recipes: The Art of Scientific Computing. Cambridge University Press, 1986.

729-93