# TABLE OF CONTENTS

**PREFACE** ........................................................................................................ iv  
**INTRODUCTION** .............................................................................................. 1  
1. LIKELIHOOD RATIO TRACKER THEORY .......................................................... 4  
   1.1 Bayesian Inference Model .......................................................................... 4  
      1.1.1 Simplifying Structures ...................................................................... 5  
      1.1.2 Recursive Computation ..................................................................... 6  
      1.1.3 Motion Updating .............................................................................. 7  
      1.1.4 Information Updating ...................................................................... 7  
   1.2 Extension to a Likelihood Ratio Tracker .................................................... 8  
      1.2.1 Basic Recursion ............................................................................... 8  
      1.2.2 Simplified Recursion ...................................................................... 10  
   1.3 A Decision Theoretic Approach to Calling Target Present ......................... 12  
      1.3.1 Decision Theory Paradigm .............................................................. 12  
      1.3.2 Example .......................................................................................... 14  
2. APPLICATION TO PERISCOPE DETECTION RADAR ........................................ 16  
   2.1 Problem Statement .................................................................................... 16  
   2.2 System Architecture .................................................................................. 17  
   2.3 Clutter Tracker .......................................................................................... 19  
      2.3.1 Data Stationarity ............................................................................. 20  
      2.3.2 Optimal Filter for Linear Prediction ............................................... 20  
   2.4 Target Tracker .......................................................................................... 24  
      2.4.1 State Space ..................................................................................... 24  
      2.4.2 Initialization .................................................................................... 24  
      2.4.3 Target Motion .................................................................................. 25  
      2.4.4 New Target Introduction ............................................................... 25  
   2.5 The Measurement Likelihood Ratio ............................................................ 26  
      2.5.1 Single Cell Likelihood Ratio ............................................................ 26  
      2.5.2 Multiple Cell Likelihood Ratio ......................................................... 27  
      2.5.3 Accounting for Uncertain Clutter Mean Level ................................... 28  
   2.6 Numerical Implementation Issues ............................................................... 28  
      2.6.1 Velocity Sheets ............................................................................... 29  
      2.6.1.1 Static Sheets .............................................................................. 29  
      2.6.1.2 Moving Sheets .......................................................................... 30  
      2.6.2 Sampled-Field Version of LRT ......................................................... 30
3. THEORETICAL PERFORMANCE PREDICTION ........................................ 32
  3.1 Log-likelihood Accumulation .................................................. 33
    3.1.1 Constant Target .................................................................. 34
    3.1.2 Fluctuating Target .............................................................. 36
    3.1.3 Mismatched Processing Method ............................................ 38
    3.1.4 Effect of Uncertain Background ......................................... 40
    3.1.5 Unknown Signal Level .......................................................... 42
    3.1.6 Target Capture Loss ............................................................ 45
    3.1.7 Conclusions ..................................................................... 46
  3.2 Log-likelihood Accumulation on a Track ................................... 47
    3.2.1 Three Models of Clutter Encountered .................................. 48
    3.2.2 Numerical Results ............................................................... 51
  3.3 Clutter Peak Analysis .............................................................. 58
    3.3.1 Simulated Data Scenes Considered ...................................... 58
    3.3.2 Processing Methodology ..................................................... 58
    3.3.3 Local Peak Distributions ..................................................... 59
    3.3.4 Extrapolation to Four False Alarm Rates .............................. 60
    3.3.5 Estimated Receiver Operating Characteristics ...................... 61
  3.4 Log-likelihood Accumulation for Alternate Data Types and Processors 61
    3.4.1 Case 1: No Clutter Estimation ............................................ 64
    3.4.2 Case 2: No Mean Clutter Estimate With Thresholded Data ....... 71
    3.4.3 Case 3: Perfectly Known Mean Clutter ................................ 75
    3.4.4 Case 4: Data Normalized by Mean Clutter ............................ 77
    3.4.5 Case 5: Data Normalized by Mean Clutter and Thresholded .... 82
    3.4.6 Conclusions .................................................................... 87
4. UCTP SOFTWARE ........................................................................ 94
  4.1 Clutter Tracker ..................................................................... 94
    4.1.1 Inputs ........................................................................... 95
    4.1.2 Class Structure ................................................................. 95
    4.1.3 Functions ..................................................................... 96
      4.1.3.1 Covariance Calculation .............................................. 96
      4.1.3.2 Filter Coefficient Calculation ....................................... 97
      4.1.3.3 Residual Variance ..................................................... 97
      4.1.3.4 Evaluation of Residual Variance for Arbitrary Filter Weights 97
    4.1.4 Outputs .................................................................. 97
  4.2 Likelihood Ratio Tracker ......................................................... 98
    4.2.1 Main Event Loop ............................................................... 99
    4.2.2 Measurement Likelihood Ratio .......................................... 99
      4.2.2.1 Single / Wide / Nyquist ........................................... 99
      4.2.2.2 Residual Variance Usage ........................................ 100
      4.2.2.3 Static / Moving Velocity Sheets ................................. 100
      4.2.2.4 Radar to Nodestar Cell Conversion ............................ 100
    4.2.3 Motion Update ................................................................. 100
  4.3 Graphical User Interface ........................................................ 101
    4.3.1 Selecting the Clutter File and Injecting the Target(s) ............ 104
    4.3.2 Generating the H Coefficients and Running the Clutter Tracker 105
    4.3.3 Running the LRT .............................................................. 106
    4.3.4 Image Processing ............................................................. 106
  4.4 Utility Programs ................................................................. 108
    4.4.1 Target Injection ............................................................... 108
    4.4.2 Local Peaks .................................................................. 109
PREFACE

This work was performed under contract N00014-95-C-0198 to the Office of Naval Research during the period August 1995 through April 1998.
INTRODUCTION

The performance of highly sensitive modern sensors searching for weak targets is severely diminished by the presence of clutter. Examples of such sensor systems include radar, both active and passive sonar, electro-optical, ESM, and infrared sensors. In each case, strong environmentally-induced clutter responses appear at the receiver, and, lacking an automated method for removing them from the tactical display, an operator is left to manually separate targets from clutter. Raising acceptance thresholds in existing algorithms can eliminate much of the clutter, but at the expense of dropping targets as well.

Objectives. To address this problem, the Office of Naval Research initiated the Uncluttered Tactical Picture (UCTP) project in 1995. The goal of the program was to devise and demonstrate a seamless detection methodology that automatically eliminates clutter from tactical displays and thereby enhances sensor utility. Metron was the fusion task leader. The objectives of the data fusion task were to (1) use dynamic models of the clutter along with Bayesian inference techniques to estimate environmental as well as target state parameters, and (2) feed the environmental estimates back into the data fusion system to remove clutter and improve the tactical picture.

Examples of environmental parameters are slowly varying parameters that affect sensor performance such as signal propagation conditions, and environmentally induced clutter. By estimating the environmental parameters, the modeling of the target response will be improved, and the clutter responses that are predicted to have come from environmental sources can be removed from the tactical picture.

Fusion Methodology. In response to the first goal Metron extended the traditional nonlinear Bayesian fusion theory of target tracking exemplified by Stone and Corwin (1994) to one that estimates environmental parameters simultaneously with target state. A general likelihood ratio tracker (LRT) was developed which tracked the joint state without a target
being present, while simultaneously accumulating likelihood ratios over possible target tracks, performing “track-before-detect” processing on the target. A statistical decision theory method was developed to take the LRT output and call detection, balancing in a quantitative fashion the tradeoff between false alarms and target detections.

**Periscope Detection Radar Test Problem.** In order to demonstrate our method, the general theory was applied to the problem of detecting an intermittently exposed, small RCS submarine periscope by a scanning high resolution shipboard radar amidst strong ocean backscatter clutter. With the proliferation of quiet diesel electric submarines, passive acoustic detection of submarines has becoming more difficult, especially in littoral regions where those submarines are likely to operate. An effective periscope detection radar that can automatically detect exposed periscopes and masts while sustaining a very low false alarm rate would greatly improve surface ship self-defense.

Metron developed a system architecture tailored to the periscope detection radar that involved two trackers operating simultaneously—one to estimate the time and space varying mean clutter backscatter, and another to detect the periscope using likelihood ratios conditioned on the clutter estimates. This architecture accomplishes one of the goals of the UCTP project by using real-time sensor returns to estimate the clutter caused by breaking ocean waves and feeding those estimates to the target tracker. The target tracker is a nonlinear, Bayesian, likelihood ratio tracker that integrates likelihood ratios over time to identify and call target tracks. Target likelihood functions appropriate to a two time scale, compound physical description of the spikey radar data at very low grazing angle were developed and coded into the target tracker.

**Simulation and Stimulation.** A high fidelity simulation of ocean backscatter from Dynamics Technology, Inc. (DTI) was used to develop and debug the computer code, and then to demonstrate the algorithms. Computer code for simulating periscope signatures was also developed by DTI. The starting point for Metron’s code was the Nodestar system, a discrete, nonlinear, recursive, Bayesian data fusion engine for performing target state estimation developed for the Spotlight Advanced Technology Demonstration. At the same time, we undertook a theoretical analysis to predict the rate at which targets would accumulate likelihood under a number of conditions such as target fluctuation statistics and degree of mismatch between assumed and true target strength. Another analysis compared the maximum performance available given five different types of radar data (raw data, thresholded data, data with clutter estimated, data normalized by clutter estimate, and normalized data thresholded) all processed optimally.
Performance Benchmarking. The final phase of the project was an extensive benchmarking effort to determine the performance of the UCTP system operating on available experimental clutter data with injected simulated target signatures, and to compare it to other periscope detection methods. Six suitable clutter scenes were obtained from DTI. The performance analysis consisted of runs with just the clutter data to observe and analytically fit fluctuations of the detection statistic, and then runs with injected target signatures over a full range of target strengths to determine performance parametrically. The detection thresholds used in the target runs were computed by extrapolating the analytic fits from the clutter runs to maintain a predetermined false alarm rate.

Summary. To summarize, the approach to periscope detection developed in the UCTP project relies on following four tools working together:

1) High resolution radar with sufficient resolution to resolve the clutter structure

2) Clutter tracker designed to estimate the mean clutter backscatter without confusing the target as aggravated clutter

3) Likelihood converter that performs optimal clutter-conditioned nonlinear transformation on the data

4) Track-before-detect likelihood processor

Organization of the Report. Chapter 1 presents the mathematical theory used in the joint target and environment estimation. It is a Bayesian approach in likelihood ratio form. In Chapter 2 we discuss the application of that theory to the periscope detection radar problem. The system architecture with target and clutter trackers is described. In addition, formulas for the measurement likelihood functions tailored to the problem are presented. The subject of Chapter 3 is theoretical performance analysis. Target likelihood accumulation in a number of cases is discussed along with clutter peak results using four large simulated data scenes, and their combination into ROC curves. The second half of the chapter compares optimal processors for five radar data types. Chapter 4 describes the software developed for the UCTP project. This includes the clutter and target trackers, a graphical user interface, and a number of utility routines. Results from the performance benchmarking are presented in Chapter 5.
CHAPTER 1

1. LIKELIHOOD RATIO TRACKER THEORY

In this chapter we present the theoretical basis for jointly estimating environmental and target state parameters. The framework adopted is an extension of the nonlinear, recursive, Bayesian approach used for target tracking in Stone and Corwin (1994). The first section discusses the Bayesian inference model and simplifying structures that allow a recursive computation. The second section extends this tracker to a likelihood ratio form that performs "track-before-detect" processing by accumulating measurement likelihood ratios over time. This form permits one to simultaneously answer the questions: Is there a target present? and If so, where is it? The final section provides an algorithm rooted in statistical decision theory for declaring the existence of a target based on the likelihood ratio tracker output.

1.1 Bayesian Inference Model

We consider the data fusion inference problem involving the joint estimation of target and environmental state. This is an extension of the existing theory for estimating target state (see Ho and Lee (1964), Jazwinski (1970), and Stone and Corwin (1994)). Let \( X \) be the state space of a single target. The target state space can include position (3D), velocity, and any attribute associated with the target that is deemed important. Let \( E \) be the environmental state space. This will consist of selected environmental parameters. Let \( S = X \times E \) be the joint environmental and target state space. Pick an arbitrary starting time and designate it as \( t = 0 \). For each \( t \) we wish to estimate \( S(t) = (X(t), E(t)) \), the state of the target and environment at time \( t \). Since both \( E(t) \) and \( X(t) \) are uncertain, we view them as random variables. In addition, we view \( S = \{S(t); t \geq 0\} = \{(X(t), E(t)); t \geq 0\} \) as a stochastic process.
The prior information about the target, its presumed motion through state space, and the evolution of environmental parameters, is incorporated into the probability measure $P_0$ defining this process. Thus $P_0$ becomes the prior distribution for the Bayesian inference process. Suppose that by time $t$ we have a set of observations $\Theta^{K(t)}(t) = \{\Theta_1, \Theta_2, \ldots, \Theta_{K(t)}\}$. Each observation $\Theta_i$ can be a discrete event that occurred at some time $u$ where $0 \leq u \leq t$, or a continuous event such as the time series output from a sensor that has occurred over an interval of time. The data fusion problem is now equivalent to computing the posterior process

$$\left\{S(u); u \geq 0 \mid \Theta^{K(t)}(t)\right\} = \left\{(X(u), E(u)); u \geq 0 \mid \Theta^{K(t)}(t)\right\} = \left\{\tilde{S}(u); u \geq 0\right\} = \left\{\tilde{X}(u), \tilde{E}(u); u \geq 0\right\}.$$ 

The distribution of $\left\{\tilde{S}(u); u \geq 0\right\}$ is the posterior distribution on the joint state of the target and the environment for $u \geq 0$. This is the best estimate of $\tilde{S}(u)$ at time $u$ given the data $\Theta^{K(t)}$. The posterior distribution contains all of the information about $\tilde{S}(u)$ that we can infer from $\Theta^{K(t)}$. From this distribution we can compute means, 86% containment regions, approximate ellipses, or whatever summary information we wish about the estimated state of the target and the environmental parameters.

Considered in this light, the problem of jointly estimating target and environmental state is a complicated application of Bayes' rule. We now propose three simplifying assumptions that will allow us to recursively compute the posterior distribution of $S$.

1.1.1 Simplifying Structures

Let us assume that the observations in $[0, t]$, $\Theta^{K(t)} = (\Theta(t_1), \ldots, \Theta(t_k))$, are obtained at a set of discrete times $0 \leq t_1 < t_2 < \ldots < t_k \leq t$ where $k = K(t)$. Let $s_i = (x_i, e_i)$ denote the joint value of the environment and target state at time $t_i$ for $i = 1, \ldots, k$, and let

$$p(s_1, \ldots, s_k, s) = \Pr\{S(t_1) = s_1, \ldots, S(t_k) = s_k, S(t) = s\}$$

be the prior probability (density) that the process $S$ passes through the states $s_1, \ldots, s_k$ at times $t_1, \ldots, t_k$. Suppose that the joint likelihood function, $L$, for this set of observations depends only on the system states, $S(t_1), \ldots, S(t_k)$, at these times. Then we have

$$\Pr\{\Theta^k = (\theta_1, \ldots, \theta_k) \mid S(u), 0 \leq u \leq t\} = \Pr\{\Theta^k = (\theta_1, \ldots, \theta_k) \mid S(t_1) = s_1, \ldots, S(t_k) = s_k\}.$$
We can now define

\[ L(\theta_1, \ldots, \theta_k | s_1, \ldots, s_k) \equiv \Pr\{ \Theta^k = (\theta_1, \ldots, \theta_k) | S(t) = s_1, \ldots, S(t_k) = s_k \}. \]

Let \( q(t, s) = \Pr\{ \tilde{S}(t) = s \} \). The function \( q(t, \cdot) \) gives the posterior distribution on \( S(t) \) given \( \Theta^k \). By Bayes theorem,

\[
q(t, s) = \frac{\Pr\{ \Theta^k = (\theta_1, \ldots, \theta_k) \text{ and } S(t) = s \}}{\Pr\{ \Theta^k = (\theta_1, \ldots, \theta_k) \}}
= \frac{\int L(\theta_1, \ldots, \theta_k | s_1, \ldots, s_k) p(s_1, s_2, \ldots, s_k, s) \, ds_1 ds_2 \cdots ds_k}{\int L(\theta_1, \ldots, \theta_k | s_1, \ldots, s_k) p(s_1, s_2, \ldots, s_k, s) \, ds_1 ds_2 \cdots ds_k ds}. \tag{1.1}
\]

Equation (1.1) gives a formula for calculating the joint probability (density) of \( \tilde{S}(t) = (\tilde{E}(t), \tilde{X}(t)) \). In general there will be no nice method of computing (1.1). Instead one will have to use numerical methods such as Monte Carlo integration to perform the calculations in (1.1).

1.1.2 Recursive Computation

We now make some additional simplifying assumptions that will allow us to perform the calculation in (1.1) in a recursive manner. First, the stochastic process \( S \) must be Markovian in the state space \( S = X \times E \). Second the likelihood of an observation \( \Theta(t_i) \) given a path of the process must depend only on the present state, i.e.,

\[
\Pr\{ \Theta(t_i) = \theta_i | S(t_i) = s_1, \ldots, S(t_k) = s_k \} = \Pr\{ \Theta(t_i) = \theta_i | S(t_i) = s_i \} = L_i(\theta_i | s_i),
\]

and the distribution of \( \Theta(t_i) \) must be independent of \( \Theta(t_j) \) given \( S(t_i) = s_1, \ldots, S(t_k) = s_k \) for \( i \neq j \). These are statements about the way in which the observations and the errors in observation depend upon the true tactical and environmental state variables.

Let \( p_0 \) be the probability (density) function for \( S(0) \) and \( p_i(s, s_{i-1}) = \Pr\{ S(t_i) = s_i | S(t_{i-1}) = s_{i-1} \} \). Then

\[
p(s_1, \ldots, s_k) = \int \left[ \prod_{i=1}^{k} p_i(s_i, s_{i-1}) \right] p_0(s_0) \, ds_0, \tag{1.2}
\]

\[
L(\theta_1, \ldots, \theta_k | s_1, \ldots, s_k) = \prod_{i=1}^{k} L_i(\theta_i | s_i).
\]
Substituting (1.2) into (1.1), we obtain

\[
q(t_k,s) = \frac{1}{C'} \int L_k(\theta_k | s) \left[ \prod_{i=1}^{k-1} L_i(\theta_i | s_i) \right] p_k(s, s_{k-1}) \left[ \prod_{i=1}^{k-1} p_i(s_i, s_{i-1}) \right] p_0(s_0) ds_0 ds_1 \cdots ds_{k-2} ds_{k-1} \\
= \frac{1}{C'} L_k(\theta_k | s) \int p_k(s, s_{k-1}) \left[ \int \prod_{i=1}^{k-1} L_i(\theta_i | s_i) p_i(s_i, s_{i-1}) \right] p_0(s_0) ds_0 ds_1 \cdots ds_{k-2} ds_{k-1} ds_{k-1} \\
= \frac{1}{C} L_k(\theta_k | s) \int p_k(s, s_{k-1}) q(t_{k-1}, s_{k-1}) ds_{k-1} \tag{1.3},
\]

where \( C = \int L_k(\theta_k | s) \int p_k(s, s_{k-1}) q(t_{k-1}, s_{k-1}) ds_{k-1} ds. \)

Equation (1.3) provides the basic recursive method of computing \( q(t_k, s) \). This recursion can be thought of as taking place in two steps, motion updating followed by information updating.

1.1.3 Motion Updating

The integral in the last line of equation (1.3) updates the environment and target state distribution at time \( t_{k-1} \) for the "motion" of the environment and target through the state space up to time \( t_k \). Let \( q^-(t_k, \cdot) \) denote the motion-updated distribution. Then

\[
q^-(t_k, s) = \int p_k(s, s_{k-1}) q(t_{k-1}, s_{k-1}) ds_{k-1} \text{ for } s \in S.
\]

1.1.4 Information Updating

The pointwise multiplication of \( q^-(t_k, \cdot) \) by the likelihood function \( L_k(\theta_k, \cdot) \) and the normalization by \( C \) constitutes the information update. Thus

\[
q(t_k, s) = \frac{1}{C} L_k(\theta_k | s) q^-(t_k, s) \text{ for } s \in S.
\]

If there has been no observation at time \( t_k \), then \( L_k(\theta_k | s) = 1 \), and there is no information update, only a motion update.

Additional simplifying assumptions may be warranted by the specific environmental effects to be modeled. For example, if it is assumed that the environmental dynamics is not affected by current target state, but conversely that target dynamics may be affected by environmental state, then the probability density \( p_i(s_i, s_{i-1}) \) will factor in the following way
\[ p_i(s_i, s_{i-1}) = h_1(x_i \mid x_{i-1}, e_{i-1}) h_2(e_i \mid e_{i-1}) \]  

(1.4)

where \( s_i = (x_i, e_i) \). Such a factorization represents a considerable practical simplification.

### 1.2 Extension to a Likelihood Ratio Tracker

In this section we describe how to extend the concept above to a likelihood ratio tracker. This tracker differs from the more common trackers in that it does not assume that a target is present, hence this type of tracking is often called track-before-detect. The likelihood ratio version can be used to jointly answer the questions: (1) is a target present, and (2) if so where is it? This will be appropriate for cases where one is unsure whether a target is present or not.

Let \( H_0 \) indicate the event that no target is present in the interval \([0, t]\). Let \( H_x \) be the event, the target is present at \( x \). When we write the event \( S(u)=s=(x,e) \), we mean that the event \( H_x \) has occurred at time \( u \) and the environment is \( e \). When we write \( X(u)=x \), we mean that the event \( H_x \) has occurred at time \( u \).

We wish to compute the likelihood ratio surface

\[
R(t,x) = \frac{\Pr \{ X(t) = x \mid \Theta^k = (\theta_1, \ldots, \theta_k) \} \Pr \{ H_0 \mid \Theta^k = (\theta_1, \ldots, \theta_k) \}}{\Pr \{ H_0 \mid \Theta^k = (\theta_1, \ldots, \theta_k) \} \Pr \{ H_0 \} }
\]

(1.5)

for \( s = (x,e) \in S \) and \( t \geq 0 \).

This is the ratio of the probability of the target being present and in state \( x \) given the observations \( \Theta^k = (\theta_1, \ldots, \theta_k) \) to the probability that no target is present given the observations.

#### 1.2.1 Basic Recursion

Under the three simplifying assumptions in the previous section, we can calculate \( R \) in a recursive fashion by using two intermediate functions \( Q_0 \) and \( Q_i \).
Let
\[ p_0(s) = \Pr\{S(0) = s \mid \text{target present}\} \text{ for } s \in S, \]
and
\[ h_0(e) = \Pr\{E(0) = e \mid \text{no target present}\} \text{ for } e \in E. \]
Define
\[ Q_i(t,s) = \Pr\{S(t) = s \& \Theta^{K(t)} = (\theta_1, \ldots, \theta_{K(t)})\} \text{ for } t \geq 0 \text{ and } s \in S \]
\[ Q_0(t,e) = \Pr\{H_0 \& E(t) = e \& \Theta^{K(t)} = (\theta_1, \ldots, \theta_{K(t)})\} \text{ for } t \geq 0 \text{ and } e \in E \]
and
\[ L_0(\theta_k \mid H_0 \& e) = \Pr\{\Theta(t_k) = \theta_k \mid H_0 \& e\} \text{ for } k = 1, 2, \ldots \text{ and } e \in E. \]
Then \( R \) satisfies the following recursion involving \( Q_0 \) and \( Q_i \)
\[ Q_0(0,s) = p_0(s)(1 - \Pr\{H_0\}) \text{ for } s = (x,e) \in S, \]
\[ Q_0(0,e) = h_0(e)\Pr\{H_0\} \text{ for } e \in E, \]
\[ R(0,x) = \frac{\int_E Q_0(0,(x,e))de}{\int_E Q_0(0,e)de} \]
\[ = \frac{\int_E p_0(x,e)de(1 - \Pr\{H_0\})}{\Pr\{H_0\}} \text{ for } x \in X. \]
and for \( k \geq 1, \)
\[ Q_i(t_k,s) = L_k(\theta_k \mid s)\int_s p_k(s,s_{k-1})Q_i(t_{k-1},s_{k-1})ds_{k-1} \text{ for } s = (x,e) \in S \quad (1.6) \]
\[ Q_0(t_k,e) = L_0(\theta_k \mid H_0 \& e)\int_E h_2(e,e_{k-1})Q_0(t_{k-1},e_{k-1})de_{k-1} \text{ for } e \in E \quad (1.7) \]
\[ R(t_k,x) = \frac{\int_E Q_i(t_k,(x,e))de}{\int_E Q_0(t_k,e)de} \text{ for } x \in X \quad (1.8) \]
Note that
\[ \frac{Q_i(t_k,s)}{Q_0(t_k,e)} = L_k(\theta_k \mid s)\int_s p_k(s,s_{k-1})Q_i(t_{k-1},s_{k-1})ds_{k-1} \]
\[ = \frac{\int_E h_2(e,e_{k-1})Q_0(t_{k-1},e_{k-1})de_{k-1}}{\int_E Q_0(t_{k-1},e_{k-1})de_{k-1}}. \]
where

\[
\Lambda_k(\theta_k, s) = \frac{L_k(\theta_k | s)}{L_0(\theta_k | H_0 \& e)} \quad \text{for } s = (x, e) \in S \text{ and } k \geq 1.
\]

To see that (1.6) holds, we observe that

\[
Q_k(t_k, s) = \Pr\{S(t_k) = s \& \Theta^k = (\theta_1, \ldots, \theta_k)\}
\]

\[
= \int L_k(\theta_k | s) \left[ \prod_{i=1}^{k-1} L_i(\theta_i | s_i) \prod_{i=1}^{k-1} p_i(s_i, s_{i-1}) \right] p_0(s_0)(1 - \Pr\{H_0\}) ds_0 ds_1 \cdots ds_{k-2} ds_{k-1}
\]

\[
= L_k(\theta_k | s) \int p_k(s, s_{k-1}) \left[ \prod_{i=1}^{k-1} L_i(\theta_i | s_i) p_i(s_i, s_{i-1}) \right] p_0(s_0)(1 - \Pr\{H_0\}) ds_0 ds_1 \cdots ds_{k-2} ds_{k-1}
\]

\[
= L_k(\theta_k | s) \int p_k(s, s_{k-1}) Q_k(t_{k-1}, s_{k-1}) ds_{k-1}.
\]

Similarly, we obtain (1.7) as follows:

\[
Q_0(t_k, e) = \Pr\{H_0 \& E(t_k) = e \& \Theta^k = (\theta_1, \ldots, \theta_k)\}
\]

\[
= \int L(\Theta^k = (\theta_1, \ldots, \theta_k) | H_0 \& (e_1, \ldots, e_k)) h_2(e, e_{k-1}) \left[ \prod_{i=1}^{k-1} h_2(e_i, e_{i-1}) \right] h_0(e_0) \Pr\{H_0\} de_0 de_1 \cdots de_{k-1}
\]

\[
= L_0(\theta_k | H_0 \& e) \int \left[ \prod_{i=1}^{k-1} L_0(\theta_i | H_0 \& e_i) \right] h_2(e, e_{k-1}) \left[ \prod_{i=1}^{k-1} h_2(e_i, e_{i-1}) \right] h_0(e_0) \Pr\{H_0\} de_0 de_1 \cdots de_{k-1}
\]

\[
= L_0(\theta_k | H_0 \& e) \int h_2(e, e_{k-1}) Q_0(t_{k-1}, e_{k-1}) de_{k-1}. \quad (1.9)
\]

Now (1.8) follows from the definition of \( R \) in (1.5).

### 1.2.2 Simplified Recursion

In some special cases this recursion can be replaced by a simplified one. Suppose that the environmental state \( e_i \) is assumed to be known for \( i = 1, \ldots, k \). Alternatively, we might measure the environment at each time \( t_i \) and replace the distribution on the environment by the mean of the measurement. The effect of this in the above formulation is that for the known or measured values \((\hat{e}_0, \ldots, \hat{e}_k)\) of the environment \( \Pr\{(\hat{e}_0, \ldots, \hat{e}_k)\} = 1 \).
In this case (1.9) becomes

\[
Q_0(t_k, e) = \begin{cases} 
L_0(\theta_k | H_0 \& e)Q_0(t_k, \hat{e}_{k-1}) & \text{for } e = \hat{e}_k \\
0 & \text{for } e \neq \hat{e}_k,
\end{cases}
\]

and

\[
\int Q_0(t_k, e) \, de = Q_0(t_k, \hat{e}_k) = \Pr\{H_0\} \prod_{i=1}^k L_0(\theta_i | H_0 \& \hat{e}_i). \tag{1.10}
\]

Using (1.4), we have \( p_i(s_i, s_{i-1}) = h_i(x_i | x_{i-1}, \hat{e}_{i-1}) \) where \( s_i = (x_i, \hat{e}_i) \). In a manner similar to (1.10), we obtain

\[
\int Q_1(t_k, (x,e)) \, de = Q_1(t_k, (x, \hat{e}_k))
\]

\[
= L_k(\theta_k | x, \hat{e}_k) \int_X h_i(x | x_{k-1}, \hat{e}_{k-1}) Q_1(t_{k-1}, (x_{k-1}, \hat{e}_{k-1})) \, dx_{k-1}.
\]

We can now write

\[
R(t_k, x) = \frac{Q_1(t_k, (x, \hat{e}_k))}{Q_0(t_k, \hat{e}_k)}
\]

\[
= \frac{L_k(\theta_k | x, \hat{e}_k) \int_X h_i(x | x_{k-1}, \hat{e}_{k-1}) Q_1(t_{k-1}, (x_{k-1}, \hat{e}_{k-1})) \, dx_{k-1}}{\Pr\{H_0\} \prod_{i=1}^k L_0(\theta_i | H_0 \& \hat{e}_i)}
\]

\[
= \Lambda_k(\theta_k | x, \hat{e}_k) \int_X h_i(x | x_{k-1}, \hat{e}_{k-1}) R(t_{k-1}, x_{k-1}) \, dx_{k-1} \quad \text{for } k \geq 1
\]

where \( \Lambda_k \) is the measurement likelihood ratio, i.e.,

\[
\Lambda_k(\theta_k | x, \hat{e}_k) = \frac{L_k(\theta_k | x, \hat{e}_k)}{L_0(\theta_k | H_0 \& \hat{e}_k)}.
\]

The simplified recursion becomes

\[
R(0, x) = \frac{p_0(x, \hat{e}_0)(1 - \Pr\{H_0\})}{\Pr\{H_0\}} \quad \text{for } x \in X,
\]
and, for \( k \geq 1 \),

\[
R(t_k, x) = \Lambda_k(\theta_k | x, \hat{e}_k) \int_{\mathbb{X}} h_i(x | x_{k-1}, \hat{e}_{k-1}) R(t_{k-1}, x_{k-1}) \, dx_{k-1} \quad \text{for } x \in \mathbb{X}
\]

1.3 A Decision Theoretic Approach to Calling Target Present

We now show how one can use the posterior likelihood ratio \( R(t, x) \) to decide when to call a target present. One of the virtues of a likelihood ratio tracker is that it can track before it detects. By this we mean that the tracker can associate peaks in the measurement likelihood ratio function over time without assuming they are caused by a target. A series of observations may produce small peaks in the measurement likelihood ratio \( \Lambda \). If these peaks correspond to a consistent velocity hypothesis for the target’s movement over time, then they will reinforce one another, becoming larger and larger in the posterior likelihood function \( R(t, x) \). At some point we want to call this large peak a detection and say there is a target present. This section presents a logical method for doing this based on Bayesian Decision Theory (see for example Ferguson (1968)).

1.3.1 Decision Theory Paradigm

At time \( t \) we have a set of hypotheses:

\[
H(t) = \{H_0(t) \cup \{H_x(t) ; x \in \mathbb{X}\}\}
\]

where

\[
H_0(t) = \text{target not present in } [0, t]
\]

\[
H_x(t) = \text{target present at } x \text{ at time } t.
\]

Similarly, there is a set of actions

\[
A(t) = \{A_0(t) \cup \{A_x(t) ; x \in \mathbb{X}\}\}
\]

where

\[
A_0(t) = \text{declare target not present in } [0, t]
\]

\[
A_x(t) = \text{declare target present at } x \text{ at time } t.
\]
Let $X^+ = \{0\} \cup X$. Then $X^+$ is the state space $X$ augmented by the state 0, target not present.

For each action $A_x$, there is an associated payoff

$$g(x, x') = \begin{cases} \text{Payoff for declaring target present at } x \text{ when } H_{x'} \text{ is true} \\ \end{cases} \text{ for } x' \in X^+.$$ 

This gain function is defined by the modeler to weigh the benefit of the different actions. An optimal decision rule is one which maximizes the expected gain, given the current estimate of target state.

Let

$$q(t_k, x) = \Pr\{X(t) = x \mid \Theta^t\} \text{ for } x \in X,$$

$$q(t_k, 0) = \Pr\{H_0 \mid \Theta^t\}.$$ 

From definitions of $Q_0$ and $Q_t$, we have

$$q(t_k, x) = \frac{\int_{E} Q_t(t_k, (x, e)) de}{C} \text{ for } x \in X,$$

$$q(t_k, 0) = \frac{\int_{E} Q_0(t_k, e) de}{C}$$

where

$$C = \int_{E} Q_0(t_k, e) de + \int_{S} Q_t(t_k, s) ds.$$ 

From (1.8) we obtain

$$R(t_k, x) = \frac{q(t_k, x)}{q(t_k, 0)} \text{ for } x \in X.$$ 

The expected payoff for taking action $A_x$ at time $t$ is

$$G(t_k, x) = g(x, 0)q(t_k, 0) + \int_{X} g(x, x')q(t_k, x') dx'$$  \hspace{1cm} (1.11)$$

We seek an action $A_{x^*}$ that will maximize $G(t_k, \cdot)$. That is we seek $x^*$ such that

$$G(t_k, x^*) = \max_{x \in X^*} G(t_k, x).$$
1.3.2 Example

As an example consider a situation in which we wish to decide whether or not to call a target present and specify its location. In this case we might use the following payoff function.

\[
g(x, x') = K(1, 1) \text{ if } \|x - x'\| \leq D \text{ for } x, x' \in X
\]
\[
g(x, 0) = -K(1, 0) \text{ if } x \neq 0
\]
\[
g(0, x') = -K(0, 1) \text{ for } x' \in X
\]
\[
g(0, 0) = K(0, 0)
\]

where all the \(K\)'s are positive. The constant \(D\) could represent the effectiveness radius of a weapon to be launched. The expected payoff then becomes

\[
G(t_k, x) = K(1, 1) \int_{\|x' - x\| \leq D} q(t_k, x') dx' - K(1, 0) q(t_k, 0) \text{ for } x \neq 0.
\]
\[
G(t_k, 0) = -K(0, 1) \int_x q(t_k, x') dx' + K(0, 0) q(t_k, 0) \text{ for } x = 0.
\]

From these two equations, we see that we do should the following to maximize \(G(t_k, \cdot)\).

Choose \(x = 0\) if

\[
K(1, 1) \int_{\|x' - x\| \leq D} q(t_k, x') dx' + K(0, 1) \int_x q(t_k, x') dx' < [K(0, 0) + K(1, 0)] q(t_k, 0) \text{ for } x \in X
\]

which holds if and only if

\[
K(1, 1) \int_{\|x' - x\| \leq D} \frac{q(t_k, x')}{q(t_k, 0)} dx' + K(0, 1) \int_x \frac{q(t_k, x')}{q(t_k, 0)} dx' < K(0, 0) + K(1, 0) \text{ for } x \in X
\]

which in turn holds if and only if

\[
K(1, 1) \int_{\|x' - x\| \leq D} R(t_k, x') dx' + K(0, 1) \int_x R(t_k, x') dx' < K(0, 0) + K(1, 0) \text{ for } x \in X.
\]

Otherwise choose \(x^*\) equal to the \(x\) that maximizes

\[
K(1, 1) \int_{\|x' - x\| \leq D} q(t_k, x') dx' - K(1, 0) q(t_k, 0)
\]

which is equivalent to choosing the \(x\) that maximizes

\[
\int_{\|x' - x\| \leq D} R(t_k, x') dx'.
\]
The above rule can be summarized as follows:

**DECISION RULE**

Choose $x = 0$ if

$$K(1,1) \int_{|x' - x| \leq D} R(t_k, x') dx' + K(0,1) \int_{X} R(t_k, x') dx' < K(0,0) + K(1,0) \text{ for } x \in X.$$

Otherwise choose $x^*$ equal to the $x$ that maximizes

$$\int_{|x' - x| \leq D} R(t_k, x') dx'.$$

(1.12)

Choosing $x$ to maximize (1.12) is similar to choosing the peak in the posterior likelihood ratio surface that has the highest integrated likelihood ratio where the integration is taken over the neighborhood $\{x' : \|x' - x\| \leq D\}$. There is a possible difference. Suppose two peaks are close together. Then by choosing $x$ between them, the integration in (1.12) might be able to pull in the mass from both peaks and produce a larger value than by choosing $x$ to be centered at either one of the peaks.

The target detection system will monitor the test statistic in brackets on the left and call target present when it exceeds the threshold on the right. The weighting among the $K$’s allows the designer to trade off between false alarm rate and probability of detection.
CHAPTER 2

2. APPLICATION TO PERISCOPE DETECTION RADAR

Chapter 1 presented the mathematics to be used for incorporating environmental effects directly into a tracking solution. In order to apply that theory to a specific problem, detailed modeling is needed. In particular, it is necessary to develop models for the target, the environment, and the sensor responses. This chapter presents models for those components, indicates how the various pieces fit together in a system architecture, and describes some numerical issues in the implementation.

Section 2.1 describes the periscope detection problem, the form of the data to be processed, and a basic statistical model for the data. The system architecture is outlined in Section 2.2. The architecture splits into two parts: a clutter tracker and a target tracker. These two components are described in Sections 2.3 and 2.4. The measurement likelihood ratio appropriate for the periscope detection problem is presented in Section 2.5. Finally, Section 2.6 discusses two numerical issues in the implementation related to velocity sheets and an alternate viewpoint on representing the likelihood function.

2.1 Problem Statement

The UCTP project has focused its attention on the problem of detecting and tracking a submarine’s exposed periscope or mast from a shipboard radar. The nominal characteristics of the radar are similar to the AN/APS-137: X-band with 30 cm range resolution, a 2.4° beamwidth, and a 5 Hz scan rate. It is assumed that the periscope will be exposed on the order of 5 seconds, although this is not directly used by the tracking system which is operating
recursively and will build up likelihood over the full exposure period. The search area extends 10 nm from the ship.

With its 30 cm resolution, such a radar is able to resolve ocean surface features along the dominant long waves. The source of environmentally induced clutter is enhanced backscatter near crests of the long waves. The returns there are substantially higher than the mean ambient level. These high intensity clutter spikes are impossible to eliminate using a higher acceptance threshold without seriously degrading detection performance.

**Statistical Description of Radar Backscatter.** A statistical characterization of the spiky clutter and the physical mechanisms have been the subject of both theoretical and experimental work for the past two decades (Jakeman and Pusey (1977), Ward, Baker, Watts (1990), and Lee *et al.* (1995)). One of the most successful approaches to describing the statistical behavior is a model with two time scales. In this model the received signal is represented as a compound process where a fast speckle process is modulated by a slower process describing the scattering features in a radar cell. Over short time intervals (up to approximately 250 ms) the intensity observed in a fixed range cell is Rayleigh distributed. Over longer times, the mean of the Rayleigh component follows a Gamma distribution whose shape parameter is a function of the radar and ocean parameters such as grazing angle, resolution cell size, frequency, look direction, and sea state. The speckle depends on minute variations in the surface and is treated as uncorrelated noise. The underlying mean however depends on larger scale structures which can be modeled and tracked. Radar system noise is treated as an independent additive complex Gaussian.

### 2.2 System Architecture

In the periscope detection radar problem, the most important environmental effect is the clutter caused by breaking waves. By estimating the clutter mean power in each cell, it will be possible to condition the likelihood function and correctly scale the raw data. Parameterizing the clutter level to the required resolution presents a formidable implementation problem. To counter that problem we will introduce some simplifications into the general Bayesian framework.

These simplifications are specific to the periscope detection problem. The first simplification is that the clutter tracker will produce an independent estimate for the mean clutter
level in each cell rather than a fully joint distribution due to the large number of data cells. The clutter estimate will vary spatially, but correlations in the estimate from cell to cell will not be used. The second simplification is that the evolution of the target and clutter are independent. In reality, the submarine’s motion will not affect the clutter, but the clutter’s motion may affect the target since submarines at periscope depth prefer to move across the direction of the waves to maintain stability.

Figure 2.1 shows the overall system architecture and the relationship between the two trackers. The clutter tracker is on the left and the target tracker in on the right. Time advances down the page. The radar data is received at time \( t \) and sent to each tracker. The clutter tracker preprocesses the data and filters it to produce an estimate of the clutter level in each radar range cell at the next time step, \( t+1 \). At time \( t \), the target tracker uses the clutter prediction at time \( t \) based on previous data, along with the radar data at time \( t \), to compute the measurement likelihood function and perform an information update. Then it motion updates the state estimate to time \( t+1 \), and the process repeats. Both of the trackers are discussed in more detail below.

In this design, the clutter estimate for time \( t \) is based on data received up to and including the previous time step. This was chosen so that the target likelihood calculation at time \( t \) can proceed as soon as the raw data is available. We have also implemented an alternate design that introduces a lag into the target tracker, allowing the clutter tracker to use data forward in time relative to the target tracker, and possibly improving its estimate. We used this variation in the performance studies reported in Chapter 5.
2.3 Clutter Tracker

The objective of the clutter tracker is to produce estimates of the clutter background and send the estimates to the target tracker. By estimating the clutter correctly, the effects of bright clutter in the target tracker can be reduced.

The clutter tracker design is a result of examining the clutter phenomenology, particularly the compound statistical description of the time scales involved. It is a statistical approach that uses temporal and spatial stationarity along with past statistical averages to predict future clutter levels. The clutter tracker is designed to estimate the statistical characteristics of the most recent data, compute an optimal filter, and then apply that filter to the
new data, until a time when a new filter is designed. The time period used for a filter is related to the variability of the environmental conditions and the location of the ship. Disjoint calculations are performed in "blocks" of range and time. The block size will be determined by the rate of change of hydrodynamic and meteorological quantities.

The radar may be moving while collecting data which will cause the range/azimuth cells to shift. The shift in the range direction will be greatest along the line of motion of the radar and decrease by the cosine of the look angle relative to that line for other beams. The clutter estimate will be performed relative to the most recent data point. Consequently, older data will be shifted an integral number of bins to align them relative to the most recent data. Once the location of the radar is known for the next scan, the data can be shifted again before use in the target tracker. The clutter patches will persist for only a few seconds and extend over a number of range cells so this rough motion compensation is sufficient.

2.3.1 Data Stationarity

The intensity data is first transformed from the IQ level to a sequence, call it \( x \), by taking the logarithm and removing any range dependent mean level. The estimation procedure will operate on \( x \) to produce an estimate \( \hat{x} \) which can then be transformed back to intensity by adding the mean and exponentiating. To predict future values we assume that \( x \) is stationary in time, which means that the past statistical behavior will continue into the future. We also assume that \( x \) is stationary in space as well. This will lead to simplifications noted below. However, both assumptions are not true: over time the sea surface characteristics and propagation will change; over space the radar cells change size, shape, and look angle relative to the sea. To compensate for the non-stationarity we create blocks in the data spatially and temporally. The blocks are chosen so that within each block \( x \) is stationary, but with different properties between blocks. The size of the blocks will be on the order of 15 minutes in time, a mile in range, and tens of degrees in azimuth.

2.3.2 Optimal Filter for Linear Prediction

Fix a block and a beam within the block. Let \( x(t,n) \) be conditioned data at time \( t \) in range bin \( n \). The sequence \( x \) is the sum of an underlying mean clutter level, denoted \( y \), and a zero-mean speckle term denoted \( s \). We will form an estimate of \( y(t,n) \) by using a weighed average of nearby cells:

\[
\hat{y}(t,n) = \sum_{i=-M}^{N} \sum_{j=-A}^{B} h(i,j) x(t-i,n-j).
\]
In the time dimension the sum extends over the past $N$ scans, the current scan, and $M$ scans into the future, while in the range direction it extends over bins $B$ less than the estimated range to $A$ greater. The values of $A$, $B$, $M$, and $N$ will be determined by the correlation structure in the data. A sketch of the data mask is shown in Figure 2.2.

![Figure 2.2: UCTP Clutter Filter Mask](image)

The optimum estimator is designed to minimize the variance of the error, i.e.

$$
E\left[ (\hat{y}(t,n) - y(t,n))^2 \right].
$$

This least squares problem arises often in regression analysis, time series modeling, and signal processing. The solution is one which projects the variable to be estimated onto the subspace spanned by the regressors. The residual is then orthogonal to that subspace. In other words,

$$
E\left[ \left( y(t,n) - \sum_{i=-M}^{N} \sum_{j=-A}^{B} h^*(i,j)x(t-i,n-j) \right) x(t-\alpha,n-\beta) \right] = 0 \quad (2.1)
$$

for $-M \leq \alpha \leq N$ and $-A \leq \beta \leq B$.

If we write the autocovariance function of $x$ as

$$
R_{xx}(k,l) = E[x(t+k,n+l)x(t,n)].
$$

equation (2.1) can be written
\[
\sum_{i=1}^{N} \sum_{j=-A}^{B} h^*(i, j) R_{xx}(\alpha - i, \beta - j) = R_{xx}(\alpha, \beta) - \delta_{(\alpha, \beta)} \sigma_s^2
\]  

(2.2)

for \(-M \leq \alpha \leq N\) and \(-A \leq \beta \leq B\). This system is a two-dimensional version of the Wiener-Hopf equation (Marple (1987)). The Toeplitz-like structure of these equations allow a computationally efficient order-recursive solution, but the matrices were small enough that we did not need to use this structure in order to solve the system. For real-time operation it may be important to utilize this structure.

The sequence \(x\) is the sum of \(y\) and \(s\). Since \(y\) and \(s\) are independent, the autocovariance of \(x\) is the sum of the autocovariances of \(y\) and \(s\). But \(s\) is independent from sample to sample, so its covariance is zero for any non-zero lags and equal to its variance at the origin. The terms on the right hand side of equation (2.2) are the covariances between \(x\) and \(y\), but this is equal to the autocovariance of \(y\).

The term \(\sigma_s^2\) is the variance of the log of the speckle. The logarithm of an exponential distribution is a Gumbel or extreme-value distribution. The variance is \(\pi^2/6\). In the performance runs we used a value slightly smaller than this because otherwise the covariance function for \(y\) (derived from the sample covariance of \(x\) and subtracting the speckle variance at the origin) would not be positive definite.

**Not Including the Cell to be Estimated.** A variation of the clutter filter mask used in the performance analysis did not allow the data from the cell to be estimated in the linear filter. This is analogous to "guard cells" in cell-averaging CFAR detectors. In order to enforce this condition and find the optimal filter coefficients, we simply zeroed out the correct elements in the matrix in order to force that coefficient to be zero. Note that this operation destroys the Toeplitz structure discussed above.

**Residual Variance.** We can directly calculate the variance \(V(h)\) of the estimator. For general filter coefficients we have

\[
V(h) = E \left[ (y(t,n) - \sum_{i=-M}^{N} \sum_{j=-A}^{B} h(i, j) x(t-i, n-j)) (y(t,n) - \sum_{i'=-M}^{N} \sum_{j'=-A}^{B} h(i', j') x(t-i', n-j')) \right]
\]

\[
= R_{yy}(0,0) - 2 \sum_{i=-M}^{N} \sum_{j=-A}^{B} h(i, j) R_{xy}(i, j)
\]

\[
+ \sum_{i=-M}^{N} \sum_{j=-A}^{B} \sum_{i'=-M}^{N} \sum_{j'=-A}^{B} h(i', j') h(i, j) R_{xx}(i-i', j-j')
\]
\[ V(h^*) = \mathbb{E} \left[ \left( y(t,n) - \sum_{i = -M}^{N} \sum_{j = -A}^{B} h^*(i, j)x(t-i,n-j) \right)y(t,n) \right] \]
\[ = R_x(0,0) - \sigma_z^2 - \sum_{i = -M}^{N} \sum_{j = -A}^{B} h^*(i, j)R_x(i,j) \]

If the target cell is not being used in the estimator, then the \( \sigma_z^2 \) term is absent.

**Covariance Estimates.** The optimal filter weights depend on the covariance of the sequence \( x \). We estimate the needed covariances by averaging lagged products of previous data. The averages are computed for a length of time (the temporal black size) to be determined. At the end of that period, equation (2.2) is solved for the filter weights \( h \). This procedure is done for each spatial block.

The covariance could also be derived a priori by a clutter model with environmental inputs. In that case, the lagged products would not to be computed, resulting in a simpler algorithm structure. An intermediate possibility is to use the models to predict the shape of the covariance function up to a small number of parameters. Estimating the small set of parameters would then be more efficient, reducing the needed computations.

**Processing Flow.** For each radar scan the data is preprocessed as discussed above. Then it is filtered with the set of coefficients \( h(i, j) \) to produce a clutter estimate. This estimate is transformed to intensity units and sent to the target tracker. A real-time tracker would have to do this main line of processing between radar scans, nominally 200 milliseconds. The most computationally intensive step is the filtering operation. Preliminary estimates show that using one SUN Sparc-1000 CPU, it takes approximately 200 milliseconds to filter the data on one azimuth, one nautical mile (6000 points), with 360 filter coefficients. Specialized digital signal processing (DSP) chips will be faster than this. For more azimuths, larger ranges, or more filter coefficients the time scales linearly.
The lagged products are also accumulated on each scan, and at a predefined interval the normal equations are solved to derive the optimal filter weights. The main computations are to solve a system of linear equations. Preliminary estimates show that with one SUN CPU, it takes approximately 1 second to solve a system of 250 equations. The time scales to the third power, so 500 equations take approximately 8 seconds, and 1000 equations 64 seconds. These are acceptable since the matrix solution is not in the main line of processing.

2.4 Target Tracker

The movement of the target over time and the response of the target seen by the sensor both need to be modeled. This section discusses the motion modeling while the statistical target signature and noise models will be discussed in Section 2.5. Also discussed in this section is the target state space and the modeling required to account for the periscope popping up. The target tracker operates as a likelihood ratio tracker discussed above.

2.4.1 State Space

Each beam of the radar is treated separately. The nominal periscope is up for only a short period of time and the chance of transiting from one beam to another is small, especially with overlapping beams. Within each beam, the target’s state space is two-dimensional: range and range rate. All quantities are measured relative to the radar. We do not try to estimate motion orthogonal to the look direction because with the scales involved it is difficult to use a single radar to “triangulate” the target. The limits on range rate will be set based on how fast a submarine can move relative to the local currents with its periscope exposed and the velocity of the ship.

The state space is discretized in range. In order to extract the most benefit from the high resolution data the cells should be smaller than the radar resolution. The discretization in range rate component is variable, set to approximate the true rate without unnecessarily adding to the computation and memory cost with unneeded hypotheses.

2.4.2 Initialization

A prior estimate of the probability that a target is present at the start of algorithm processing is necessary to start the recursion. This parameter will be the value for $R(0,x)$. This
can depend on location and velocity if prior information is available to favor certain location and velocity hypotheses.

2.4.3 Target Motion

Target motion from one time step to another is specified in the function \( h_t \) of Section 1.1.4. The standard method for using the periscope is to put it up a little as possible. The periscope is expected to be up for as little as five seconds and then disappear for much longer time periods. Consequently, the model for target motion is that it will be constant velocity. No attempt is made to model the movement of the submarine between periscope exposures. Each exposure will be treated as an independent event in the LRT. A submarine with its periscope extended cannot be traveling at high speed, but will need to have some headway to stabilize the platform. Over the short period of exposure we assume that the range rate is constant. Likelihood from cell \( (r, v) \) is displaced to \( (r + v \Delta t, v) \) over a time interval \( \Delta t \). In this model each velocity hypothesis can be treated independently.

2.4.4 New Target Introduction

One added feature in the periscope problem is that the periscope may first appear at any point in the state space. In most tracking problems the target must enter from the boundaries of the search area, so cells that transition from outside the search area into the search area are set to the initial value. Between scans there is a certain probability that the periscope appears at point \( x \). Define

\[
n(t, x) = \Pr\{\text{target appears at } x \text{ at time } t \mid H_0(t^-)\}.
\]

In addition to the transport of likelihood through the usual motion, there is a second step in the motion updating that sets \( R(t^-, x) \) to \( R(t^+, x) \) by the formula

\[
R(t^+, x) = \frac{R(t^-, x) + n(t, x)}{1 - \int n(t, x) \, dx}.
\]
2.5 The Measurement Likelihood Ratio

The measurement likelihood ratio is calculated to fuse new data into the state estimate. The clutter tracker, by raising the background level in areas of high clutter, insures that bright clutter returns will not produce incorrectly high likelihood. Thus, the clutter estimate conditions the target likelihood function. Similarly, the target likelihood conditions the clutter estimate, but the effect is small so we have ignored that interaction in our design.

2.5.1 Single Cell Likelihood Ratio

The observation at time $t$ is the received intensity $I(x,t)$ over all radar range bins. The likelihood ratio is a function of target position only since the radar is not measuring velocity. Let $x$ be a hypothetical target position. Also, let

$$\gamma(x,t) = \text{mean clutter from state (cell) } x \text{ at time } t.$$ 

In this section we shall assume that clutter is estimated with sufficient accuracy that we can take the estimate to be equal to the actual mean value. The next section discusses the modification for uncertain clutter level.

Let $I(x,t)$ be the intensity received from cell $x$ at time $t$ and define

$$f_0(\zeta | \gamma(x,t)) = \Pr \{ I(x,t) = \zeta \mid \text{mean clutter} = \gamma(x,t) \& \text{no target present at } x \}$$

$$f_1(\zeta | \gamma(x,t)) = \Pr \{ I(x,t) = \zeta \mid \text{mean clutter} = \gamma(x,t) \& \text{target present at } x \}.$$ 

Following the two-scale physical model for ocean clutter, when no target is present the distribution of the intensity $I(x,t)$ is Rayleigh with mean $\gamma(x,t)$, i.e.,

$$f_0(\zeta | \gamma(x,t)) = \frac{1}{\gamma(x,t)} \exp \left( -\frac{\zeta}{\gamma(x,t)} \right) \text{ for } \zeta \geq 0.$$ 

**Constant Target.** We first assume that a target located at $x$ will produce a complex response $W(x)$ that adds to the clutter with uniform phase relative to the (complex) clutter vector at $x$. Let $w(x)$ be the (constant) modulus of $W(x)$. The likelihood (density) function of the received intensity $\zeta$ from state $x$ given the target is at $x$ at time $t$ is thus a Rice distribution

$$f_1(\zeta | \gamma(x,t)) = \frac{1}{\gamma(x,t)} \exp \left( -\frac{\zeta + w(x)^2}{\gamma(x,t)} \right) I_0 \left( \frac{2\zeta w(x)}{\gamma(x,t)} \right) \text{ for } \zeta \geq 0,$$
where $I_0$ is the modified Bessel function of order zero.

With this model, the only data affected by the presence of a target is the return from the cell containing the target. Terms involving data from other cells cancel in the measurement likelihood ratio, resulting in the following form

$$
\Lambda_i(\theta_i | x, \gamma) = \frac{L_i(\theta_i | X(t_i) = x \& \gamma)}{L_0(\theta_i | H_0 \& \gamma)} = \frac{f_i(\theta_i(x) | \gamma(x, t_i))}{f_0(\theta_i(x) | \gamma(x, t_i))}
$$

$$
= \exp \left( -\frac{w^2(x)}{\gamma(x, t_i)} \right) I_0 \left( \frac{2 \theta_0^2(x) w(x)}{\gamma(x, t_i)} \right) \text{ for } x \in X.
$$

The fundamental Bayesian recursion proceeds by using this expression which is the likelihood ratio of observing the response $\theta_i$ given the target is present at $x$ to the likelihood of observing the response $\theta_i$ given no target is present. Both the numerator and denominator of $\Lambda_i(\theta_i | x, \gamma)$ depend on the mean clutter $\gamma(x, t_i)$ at $x$.

**Fluctuating Target.** For an exponentially fluctuating target, the response in intensity with target present is also exponential. Letting $w(x, t)$ be the mean intensity (not amplitude) we have

$$
f_i(\xi | \gamma(x, t)) = \frac{1}{\gamma(x, t) + w(x, t)} \exp \left( \frac{-\xi}{\gamma(x, t) + w(x, t)} \right) \text{ for } \xi \geq 0.
$$

The measurement likelihood ratio is thus

$$
\Lambda_i(\theta_i | x, \gamma) = \frac{1}{1 + w(x, t) / \gamma(x, t)} \exp \left( \frac{\theta_i - w(x, t)}{\gamma(x, t) + w(x, t)} \right).
$$

### 2.5.2 Multiple Cell Likelihood Ratio

An alternate model for the target signature has been investigated which involves widening the model to more than one pixel. In that case, the measurement likelihood function becomes a function of the target position, not simply the cell that it is in. As a function of position, the target will produce some spatial response in the nearby radar cells. That response is used for the target strength above in either the constant or fluctuating models. The speckle noise is independent from cell to cell, so the measurement likelihood ratio is a product over all cells affected by the target.
2.5.3 Accounting for Uncertain Clutter Mean Level

The computation in the previous section assumes that we know the mean clutter level very accurately. In practice we have an estimate of $\gamma(j,t_i)$ with uncertainty. That is, we have a probability distribution

$$P_i(j,g) = \Pr\{\gamma(j,t_i) = g\} \text{ for } g \geq 0$$

To account for this uncertainty we modify the calculation of $\Lambda_i$ by averaging the numerator $f_i$ and the denominator $f_0$ of the likelihood ratio over the distribution of $\gamma(x,t_i)$ to obtain the total likelihood under the two hypotheses. The result is

$$\Lambda_i(\theta_i | x, \gamma) = \frac{\int f_i(\theta_i(x) | \gamma(x,t_i) = g) P_i(x,g) dg}{\int f_0(\theta_i(x) | \gamma(x,t_i) = g) P_i(x,g) dg}$$

In our implementation we use a discrete approximation to $P_i(j,g)$ with six points. Five points are located symmetrically in a multiplicative sense around the estimated mean level. The spread of the five points is controlled by the variance of the clutter mean estimator that is calculated at the same time as the optimal filter coefficients. The sixth point is a large multiple (100) of the estimated clutter mean with small weight (1%). This distribution was designed after investigating the error distribution on simulated data where the true clutter mean was known.

2.6 Numerical Implementation Issues

In this section we briefly describe two issues related to the numerical implementation of the target tracker. The first is the method used to represent the motion of the target. The second is an alternate method for describing the likelihood ratio surface that does not use cells; we call this the sampled-field tracker.

The discrete form of tracker that we have implemented consists of a number of cells in the target state space of range and range rate. For each cell we maintain a likelihood ratio. The likelihood ratio associated to a cell is interpreted as the probability that a target is anywhere in the cell divided by the probability that the target is not in the search area. As far as where the target is located in the cell, a cellular tracker can't say; it only knows that the target is somewhere in the cell. For tracking problems where the likelihood ratios are very broad, there
might not be enough information to support a decision as to where in a cell the target might be. The broadness of the measurement likelihood ratio indicates that the single piece of information is insufficient to localize a target at the subcell level. However, some care must be taken in the motion model to prevent impossible hypotheses from acquiring likelihood.

2.6.1 Velocity Sheets

One of the methods designed to maintain a “clean” probability model is fractional shifts of velocity sheets. Consider the following problem: Suppose that one starts a tracking problem (using a discrete tracker in probability form, not in likelihood ratio form) knowing for certain that the target is in a particular spatial cell. We then perform a motion update over a short time interval. How does the probability mass distribute itself? One option is to keep the spatial grid constant and move a portion of the mass out of the original cell into surrounding cells, with the direction determined by the particular velocity and the fraction of mass moved by the amount that the translated version of the original cell overlaps the adjacent cell. If the distance moved is an integral number of cell widths, then the mass on a fixed velocity sheet does not spread, it is still contained in a single cell. However, if the distance is not an integral number of cells, then the area of support of the probability function at that one velocity hypothesis has increased to more than a single cell. If this is repeated a number of times, then it is possible for mass to move a number of cells equal to the number of time update steps. This is clearly undesirable since we often know an absolute maximum on target speed, and the probability support appears to be diffusing faster than this rate.

2.6.1.1 Static Sheets

One way around this problem, implemented in Nodestar, is to maintain for each sheet a “fractional offset” along with the probability information. This fractional offset represents motion of the velocity sheet a non-integral number of cells. Fractional offsets allows the tracker to avoid diffusing the probability mass. For a motion update over a short time interval, the sheet will update the fractional offset but maintain all the probability mass in the same cells. When the accumulated fractional offset is more than a full cell, then the probability mass is passed in toto to the next cell and the fractional offset decremented by a full unit.

In essence, with fractional offsets the individual sheets are slightly offset with respect to one another. However, when evaluating a measurement likelihood ratio, Nodestar would use the same method for each sheet (assuming the function was independent of velocity). This is what we call “static sheets” in the UCTP system.
2.6.1.2 Moving Sheets

However, when the measurement likelihood ratio is varying rapidly over the size of a spatial cell, it make a difference where the measurement likelihood function is evaluated. The "moving sheets" option allows the measurement likelihood ratio to be evaluated at slightly offset positions from the primary spatial grid. The advantages are that the benefits of fractional offsets are maintained and the fine-detail tracking performance improved, but at the expense of more computations. We have used the moving sheets method for the performance runs.

2.6.2 Sampled-Field Version of LRT

As noted in Section 2.5.2, the position of the target in a cell will change the model for the data expected in nearby cells, and thereby change the measurement likelihood ratio. For example, if a target is in the middle of a spatial cell, the signature will be peaked to a certain level in that cell, with symmetrically reduced levels in the adjacent cells. If the target is in the same cell but near an edge, then the expected returns will be approximately equal in the primary cell and the one adjacent cell. The cell on the other side of the primary cell will have a lowered expected return.

In the cellular version, in order to compute the measurement likelihood ratio for a spatial cell (in either the static or moving sheet options) the likelihood ratio is computed for a number of hypothetical target locations in the cell and then averaged. In the long run, if we had evaluated the measurement likelihood ratio consistently at the true location and accumulated those values, we would expect a higher sum than averaging over target locations in the cell and then accumulating those. The likelihood ratios at the alternate, but incorrect, locations in the cell dilute the pure likelihood ratio. The sampled-field version tries to alleviate this problem.

The Bayesian theory can be developed in continuous space, and, under the same independence and Markov assumptions, results in the same two alternating processes of motion updating and information updating. We have only discretized the full likelihood ratio field into cells in order to implement the theory in finite terms inside a computer. However, there is another way to render the representation finite. We adopt the viewpoint that we will attempt to calculate the exact value of the likelihood field at a number of discrete points. The points will be chosen in such a way that the full field can be exactly recovered from the sample. This immediately suggests the Nyquist Sampling Theorem which states that a continuous signal known to contain no frequency components above a certain frequency can be exactly recovered from a sample at twice the highest frequency.
If we knew that the likelihood ratio was frequency limited, then we could use this approach. However, a short consideration shows that the likelihood ratio is not band-limited because as the likelihood peak develops over the target the peak becomes sharper and sharper. However, the logarithm of the likelihood ratio does not become more peaked. Of course the peak value rises, but the peak does not become sharper. The information update step in log-likelihood units is an addition, and addition does not increase the frequency content. In the velocity component, the peak does become sharper, but only over the interval that the target is present. If we can be satisfied with time integration for a fixed duration, then the peakedness is limited.

We now briefly indicate how the information and motion update steps are executed for the sampled-point method. For the information update, we evaluate the measurement log-likelihood ratio at exactly the sample points (no averaging over cell width) and add that value to the log-likelihood field. The motion update step is slightly more complicated. If there is no diffusion, then to know the motion updated value at a sample point, we can find the point that gets mapped to that point by the motion update. This will most likely not be a sample point. However, we can exactly recover that value from the sample values at that time via interpolation. In practice it is easier to implement all of the interpolation for a full velocity sheet at once through a finite Fourier transform, when the fractional shift becomes a frequency-dependent rotation of Fourier components.

We have experimented with the sampled-field version for the periscope detection problem, but not found any great advantage over the cell method with small cells for that problem.
CHAPTER 3

3. THEORETICAL PERFORMANCE PREDICTION

This chapter is concerned with estimating the performance of the UCTP system outlined in Chapters 1 and 2. The target tracker accumulates likelihood over many hypothetical tracks, one of which is the target track, and then declares a target present when that value exceeds some threshold. The analysis of performance consists then of two parts: determining how high the target peaks are likely to accumulate and finding the threshold settings necessary to enforce the desired false alarm rate. The threshold settings are based on the level of likelihood that accumulates over clutter patches.

Section 3.1 examines the accumulation rates of target likelihood in a number of situations. Fluctuating targets are compared to constant targets. The loses due to a mismatch between the statistical assumptions in the processor and the true statistics is studied. The mismatches examined are in the target fluctuation statistics (constant and exponentially fluctuating) and also in the target strength. This last comparison is important because the true target strength will not be known in an operational system. The loses due to target capture are also studied in this section.

Section 3.2 presents an analysis of target likelihood accumulation for a number of scans, with a statistical model for the clutter encountered. Three models are presented for the clutter samples and numerical results computed for each case three target levels in four clutter scenes.

Section 3.3 discusses the peak values obtained by running the UCTP on four large simulated clutter scenes provided by DTI. A rigorous theoretical model of the clutter peak values is not available, but for each scene the clutter peak levels were found to follow a gamma distribution quite closely, allowing a numerical fit. From the numerical fit we were able to
extrapolate to much smaller false alarm rates. The log-likelihood ratio threshold settings necessary for four false alarm rates (one per beam per hour, one per beam per day, one per hour over the whole region, and one per day over the whole region) in the four simulated clutter scenes are derived. From this and the results in the previous section, ROC curves are developed for the three target levels discussed above.

Section 3.4 of this chapter examines the likelihood accumulation for five separate data types and processors. The five cases considered are: 1) no clutter estimate performed at all; 2) no clutter estimate, and the data is thresholded prior to processing; 3) the clutter is estimated perfectly; 4) the data is normalized by a perfect clutter estimate prior to processing; and 5) the data is the same as in case 4, but it is thresholded prior to processing. In each case we assume that the likelihood ratio statistic appropriate to the data at hand is used for the processor. The analysis proceeds by evaluating the log-likelihood accumulation for each case. The comparison quantitatively shows the benefits of obtaining an estimate of the clutter mean.

3.1 Log-likelihood Accumulation

The Likelihood Ratio Tracker operates by integrating measurement likelihood ratios over hypothetical tracks. A decision rule takes the output of the LRT and computes a test statistic. A simple form of the statistic is to take the peak value of the likelihood ratio across the state space and declare a target present if that value is above a threshold. The threshold setting determines the tradeoff between detections and false alarms. Typically the threshold is set to limit the number of false alarms, and then the detection performance varies depending on the signal strength: a stronger target will be more easily detected at the same threshold setting than a weaker one.

The present section describes a theory of target detection probability as a function of threshold setting. The first step in that analysis is to determine the rate at which likelihood is accumulated over the target track through the measurement likelihood ratio. The measurement likelihood ratios are multiplied along a track, which is equivalent to summing the logarithms of the measurement likelihood ratios, referred to as log-likelihoods. The log-likelihoods are statistical quantities that depend on the data received on a particular scan. We will be interested in the first two moments of the log-likelihoods.
In this section a number of tables are presented with the first two moments of the log-likelihoods as a function of signal strength under different test conditions and processing options. The test conditions include a constant target strength added into a complex Gaussian noise distribution with known variance and a fluctuating target strength added into the same noise. Two cases with imperfectly known noise level are also considered. These are of interest since the UCTP clutter tracker produces estimates of the clutter level, and the processing takes into account that the estimates have an error associated to them.

Two processing options are then considered where the target strength is unknown and the processor assumes some target strength. The signal models are constant or fluctuating. This analysis is directed to quantifying the effect of assuming a minimum detectable level in the processing and then facing a stronger target.

The last two cases studied look at the effect of a mismatch between the target model and the assumed model in the processor. In both cases we assume that the mean level is known, but in the first case the target is constant, but processed as if it were fluctuating, while in the second case the conditions are reversed, a fluctuating target is processed as if it were constant. These cases are of interest since the processor for a fluctuating target is computationally simpler than for a constant strength target.

The results in this section are the foundation for determining the target log-likelihood accumulation. The tables are specifically for a single scan. If the clutter-to-noise ratio is small the detector is limited by the target strength relative to the system noise. The analysis here is directly applicable to that case since the relative target strength is constant over all scans, and it is straightforward to compute probability of detection. In a realistic scene the clutter will vary quite a bit, and this must be modeled in the computation of PD. The subsequent section presents results for this case as a function of the clutter spikiness parameter ν.

3.1.1 Constant Target

In the constant strength target case we assume that the noise is a complex Gaussian and the target adds a complex vector into the noise with random phase relative to the noise. The distributions of the received intensity x under the two hypotheses are

\[ p_1(x, B, T) = \frac{1}{B} \exp\left(-\frac{x + T}{B}\right) I_0\left(2\sqrt{\frac{T}{B}} \sqrt{\frac{x}{B}}\right) \]
\[ p_0(x, B) = \frac{1}{B} \exp \left( -\frac{x}{B} \right) \]

\( B \) is the background level, the sum of the clutter received at the radar and the system noise. \( T \) is the constant target intensity. \( T, B, \) and \( x \) are all in units of power. This shows that the distributions are all scale invariant: everything can be measured relative to the background. The likelihood ratio and log-likelihood for this case are

\[ LR(x, B, T) = \exp \left( -\frac{T}{B} \right) I_0 \left( 2 \sqrt{\frac{T}{B}} \sqrt{\frac{x}{B}} \right) \]

\[ LLR(x, B, T) = -\frac{T}{B} + \log I_0 \left( 2 \sqrt{\frac{T}{B}} \sqrt{\frac{x}{B}} \right) \]

For the expected log-likelihood we have

\[ E(LLR)_1 = \int_0^\infty LLR(x, B, T) p_1(x, B, T) \, dx \]

\[ = \int_0^\infty \left\{ -\frac{T}{B} + \log I_0 \left( 2 \sqrt{\frac{T}{B}} \sqrt{\frac{x}{B}} \right) \left\{ \frac{1}{B} \exp \left( -\frac{x + T}{B} \right) I_0 \left( 2 \sqrt{\frac{T}{B}} \sqrt{\frac{x}{B}} \right) \right\} \right\} \, dx \]

\[ = \int_0^\infty \left\{ -S + \log I_0 \left( 2 \sqrt{S} y \right) \right\} \left\{ \exp \left( -(y + S) \right) I_0 \left( 2 \sqrt{S} y \right) \right\} \, dy \]

\( S = T/B \) is the normalized target strength and \( y = x/B \) is the normalized data. This last integral can be evaluated numerically. The variance of the log-likelihood statistic can also be computed numerically. The result is Table 3.1.

The results can be applied to the UCTP periscope detection problem as follows. A periscope may be exposed for 10 seconds, or 50 scans. Preliminary investigation of the clutter peak statistics shows that a log-likelihood threshold of approximately 50 units is necessary to limit false alarms to 1 per hour in a circle that extends 10 nautical miles from the ship. Thus, the log-likelihood must grow by one unit per scan. From the chart we see that a signal strength of a little more than 2 (= 3 dB) is absolutely necessary to achieve a 50 percent PD in this circumstance. This calculation is fine when the background is constant, as it is for low CNR, when the target strength is measured relative to the system noise. The more realistic case of a strong clutter component will be treated in the next section.
<table>
<thead>
<tr>
<th>$S$</th>
<th>$E(LLR)$</th>
<th>$\sigma(LLR)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>1.0</td>
<td>0.320</td>
<td>0.898</td>
</tr>
<tr>
<td>2.0</td>
<td>0.962</td>
<td>1.574</td>
</tr>
<tr>
<td>3.0</td>
<td>1.738</td>
<td>2.100</td>
</tr>
<tr>
<td>4.0</td>
<td>2.580</td>
<td>2.532</td>
</tr>
<tr>
<td>5.0</td>
<td>3.459</td>
<td>2.903</td>
</tr>
<tr>
<td>6.0</td>
<td>4.362</td>
<td>3.231</td>
</tr>
<tr>
<td>7.0</td>
<td>5.282</td>
<td>3.589</td>
</tr>
<tr>
<td>8.0</td>
<td>6.212</td>
<td>3.803</td>
</tr>
<tr>
<td>9.0</td>
<td>7.151</td>
<td>4.058</td>
</tr>
<tr>
<td>10.0</td>
<td>8.097</td>
<td>4.298</td>
</tr>
</tbody>
</table>

Table 3.1

3.1.2 Fluctuating Target

When the target has an exponentially fluctuating target strength, and it is in random complex phase to the noise, then this is equivalent to having a complex noise that is Gaussian and a complex target that is also Gaussian. The return seen in a cell occupied by the target is the sum of these two, also a complex Gaussian. Thus the target present and target absent distributions in this case are:

$$p_i(x, B, T) = \frac{1}{B + T} \exp\left(-\frac{x}{B + T}\right)$$

$$p_0(x, B) = \frac{1}{B} \exp\left(-\frac{x}{B}\right)$$

So the likelihood and log-likelihood are

$$LR(x, B, T) = \frac{B}{B + T} \exp\left(\frac{xT}{B(B + T)}\right)$$

$$LLR(x, B, T) = x \frac{T}{B(B + T)} - \log\left(1 + \frac{T}{B}\right)$$
This is a particularly simple log-likelihood function since is linear in the data. This allows us to compute the expected values and variance analytically. The result is

\[
E(\text{LLR})_t = \frac{T}{B} - \log\left(1 + \frac{T}{B}\right)
\]

\[
= S - \log(1 + S)
\]

\[
\text{Var}(\text{LLR})_t = \left(\frac{T}{B}\right)^2
\]

\[
= S^2
\]

A small table shows these numbers.

<table>
<thead>
<tr>
<th>( S )</th>
<th>( E(\text{LLR}) )</th>
<th>( \sigma(\text{LLR}) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0.000</td>
<td>0.0</td>
</tr>
<tr>
<td>1.0</td>
<td>0.307</td>
<td>1.0</td>
</tr>
<tr>
<td>2.0</td>
<td>0.901</td>
<td>2.0</td>
</tr>
<tr>
<td>3.0</td>
<td>1.614</td>
<td>3.0</td>
</tr>
<tr>
<td>4.0</td>
<td>2.391</td>
<td>4.0</td>
</tr>
<tr>
<td>5.0</td>
<td>3.208</td>
<td>5.0</td>
</tr>
<tr>
<td>6.0</td>
<td>4.054</td>
<td>6.0</td>
</tr>
<tr>
<td>7.0</td>
<td>4.921</td>
<td>7.0</td>
</tr>
<tr>
<td>8.0</td>
<td>5.803</td>
<td>8.0</td>
</tr>
<tr>
<td>9.0</td>
<td>6.697</td>
<td>9.0</td>
</tr>
<tr>
<td>10.0</td>
<td>7.602</td>
<td>10.0</td>
</tr>
</tbody>
</table>

Table 3.2

The analytic form shows that the mean is quadratic in the signal strength for weak targets and linear for stronger targets.

We see that the expected growth in log-likelihood is approximately equal in the two cases. The fluctuating target is slightly harder to detect based strictly on the mean. The variance is also larger in the fluctuating case, so the threshold must be set a bit higher for the probabilities of detection greater than 50%.
3.1.3 Mismatched Processing Method

**Constant Target Strength, Processed as Fluctuating.** If we process the data as a fluctuating target, when it is really a constant strength target then for the likelihood ratio statistic we have

\[ LR_{stat}(x, B, T) = \frac{B}{B + T} \exp \left( \frac{xT}{B(B + T)} \right) \]

\[ LLR_{stat}(x, B, T) = x \frac{T}{B(B + T)} - \log \left( 1 + \frac{T}{B} \right) \]

This is linear in the data, so we can compute the moments easily from the true distribution which is a Rice distribution

\[ p_1(x, B, T) = \frac{1}{B} \exp \left( -\frac{x + T}{B} \right) I_0 \left( \frac{T}{B} \right) \sqrt{\frac{x}{B}} \]

\[ \mathbb{E}(LLR_{stat}) = \frac{T}{B} - \log \left( 1 + \frac{T}{B} \right) = S - \log(1 + S) \]

\[ \text{Var}(LLR_{stat}) = \frac{(B + 2T)T^2}{B(B + T)^2} = \frac{(1 + 2S)S^2}{(1 + S)^2} \]

<table>
<thead>
<tr>
<th>S</th>
<th>(\mathbb{E}(LLR))</th>
<th>(\sigma(LLR))</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>1.0</td>
<td>0.307</td>
<td>0.866</td>
</tr>
<tr>
<td>2.0</td>
<td>0.901</td>
<td>1.491</td>
</tr>
<tr>
<td>3.0</td>
<td>1.614</td>
<td>1.984</td>
</tr>
<tr>
<td>4.0</td>
<td>2.391</td>
<td>2.400</td>
</tr>
<tr>
<td>5.0</td>
<td>3.208</td>
<td>2.764</td>
</tr>
<tr>
<td>6.0</td>
<td>4.054</td>
<td>3.090</td>
</tr>
<tr>
<td>7.0</td>
<td>4.921</td>
<td>3.389</td>
</tr>
<tr>
<td>8.0</td>
<td>5.803</td>
<td>3.665</td>
</tr>
<tr>
<td>9.0</td>
<td>6.697</td>
<td>3.923</td>
</tr>
<tr>
<td>10.0</td>
<td>7.602</td>
<td>4.166</td>
</tr>
</tbody>
</table>

Table 3.3
The loses for processing a constant target as if it were fluctuating are not very great, on the order of 5-10%. This means that the mismatched processor will take 5-10% more scans to detect a target on average than the correctly matched processor.

**Fluctuating Target Strength, Processed as Constant.** The formulas is this case are:

\[
p_1(x, B, T) = \frac{1}{B + T} \exp\left(-\frac{x}{B + T}\right)
\]

\[
LLR_{stat}(x, B, T) = -\frac{T}{B} + \log I_0 \left( \frac{T}{B} \sqrt{\frac{x}{B}} \right)
\]

\[
E(LLR_{stat})_1 = \int_0^\infty LLR_{stat}(x, B, T)p_1(x, B, T)dx
\]

\[
= \int_0^\infty \left\{-\frac{T}{B} + \log I_0 \left( \frac{T}{B} \sqrt{\frac{x}{B}} \right) \right\} \left\{ \frac{1}{B + T} \exp\left(-\frac{x}{B + T}\right) \right\} dx
\]

\[
= \int_0^\infty \left\{-S + \log I_0 \left(2\sqrt{Sy}\right)\right\} \left\{ \frac{1}{1 + S} \exp\left(-\frac{y}{1 + S}\right) \right\} dy
\]

<table>
<thead>
<tr>
<th>S</th>
<th>E(LLR)</th>
<th>σ(LLR)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>1.0</td>
<td>0.291</td>
<td>0.992</td>
</tr>
<tr>
<td>2.0</td>
<td>0.822</td>
<td>1.952</td>
</tr>
<tr>
<td>3.0</td>
<td>1.432</td>
<td>2.897</td>
</tr>
<tr>
<td>4.0</td>
<td>2.083</td>
<td>3.834</td>
</tr>
<tr>
<td>5.0</td>
<td>2.758</td>
<td>4.768</td>
</tr>
<tr>
<td>6.0</td>
<td>3.449</td>
<td>5.700</td>
</tr>
<tr>
<td>7.0</td>
<td>4.152</td>
<td>6.630</td>
</tr>
<tr>
<td>8.0</td>
<td>4.862</td>
<td>7.558</td>
</tr>
<tr>
<td>9.0</td>
<td>5.579</td>
<td>8.483</td>
</tr>
<tr>
<td>10.0</td>
<td>6.298</td>
<td>9.404</td>
</tr>
</tbody>
</table>

Table 3.4
By processing a fluctuating target as if it were constant, the average rate of log-likelihood accumulation decreases relative to the correct processor. The effect increases at higher signal levels; at $S=2$ the loss is 9%, while at $S=10$ the effect is a 17% loss.

### 3.1.4 Effect of Uncertain Background

The previous results were for cases when the background was known exactly. These would apply if the clutter tracker were able to perfectly estimate the clutter power. In reality, there will be some error in the clutter tracker estimate. Early UCTP simulation runs showed that it was necessary to include this error distribution when computing the likelihood ratio function. The effect of this was to "soften" the likelihood function, preventing isolated clutter spikes from contributing very large likelihood. The accumulation over the target is also affected, slowing down the rate.

**Constant Target Strength.** One error distribution used for UCTP runs is:

$$p\left(\frac{1}{4} g\right) = \frac{1}{12}, \quad p\left(\frac{1}{2} g\right) = \frac{1}{4}, \quad p(g) = \frac{1}{3}, \quad p(2g) = \frac{1}{4}, \quad p(4g) = \frac{1}{12},$$

where $g$ is the output of the clutter tracker. We can include this in the computation of the expected log-likelihood accumulation as follows:

$$p_i(x, B, T) = \sum_k p_k \frac{1}{m_k B} \exp\left(-\frac{x + T}{m_k B}\right) I_0\left(2 \sqrt{\frac{T}{m_k B}} \sqrt{\frac{x}{m_k B}}\right)$$

$$p_0(x, B) = \sum_k p_k \frac{1}{m_k B} \exp\left(-\frac{x}{m_k B}\right)$$

The likelihood ratio and log-likelihood ratio are lengthy to write out, but are

$$LR(x, B, T) = \frac{p_i(x, B, T)}{p_0(x, B)}$$

$$LLR(x, B, T) = \log LR(x, B, T)$$

and the expected target accumulation is

$$E(LLR)_i = \int LLR(x, B, T) p_i(x, B, T) dx.$$
This assumes that the error distribution used in the processor is the correct one. Once again, the integral can be done numerically, and the result is the following chart and graph:

<table>
<thead>
<tr>
<th>S</th>
<th>E(LLR)</th>
<th>σ(LLR)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>1.0</td>
<td>0.226</td>
<td>0.612</td>
</tr>
<tr>
<td>2.0</td>
<td>0.593</td>
<td>0.921</td>
</tr>
<tr>
<td>3.0</td>
<td>0.971</td>
<td>1.108</td>
</tr>
<tr>
<td>4.0</td>
<td>1.336</td>
<td>1.234</td>
</tr>
<tr>
<td>5.0</td>
<td>1.684</td>
<td>1.324</td>
</tr>
<tr>
<td>6.0</td>
<td>2.016</td>
<td>1.393</td>
</tr>
<tr>
<td>7.0</td>
<td>2.333</td>
<td>1.447</td>
</tr>
<tr>
<td>8.0</td>
<td>2.637</td>
<td>1.492</td>
</tr>
<tr>
<td>9.0</td>
<td>2.930</td>
<td>1.530</td>
</tr>
<tr>
<td>10.0</td>
<td>3.213</td>
<td>1.564</td>
</tr>
</tbody>
</table>

Table 3.5

We see the effect of having an uncertain background level: for the same target strength, the mean is lower, and the variance is also lower. If we take as our objective to accumulate one log-likelihood unit per scan, then a signal strength of slightly more than 3 (~ 5 dB) is required.

**Fluctuating Target Strength.** Using the notation from the previous sections we have

\[ p_1(x,B,T) = \sum_k p_k \frac{1}{m_k B + T} \exp\left(-\frac{x}{m_k B + T}\right) \]

\[ p_0(x,B) = \sum_k p_k \frac{1}{m_k B} \exp\left(-\frac{x}{m_k B}\right) \]

And, as always,

\[ LR(x,B,T) = \frac{p_1(x,B,T)}{p_0(x,B)} \]

\[ LLR(x,B,T) = \log LR(x,B,T) \]
and the expected target accumulation is

\[ E(\text{LLR})_1 = \int \text{LLR}(x, B, T)p_i(x, B, T)dx \]

We compute the integral numerically to find:

<table>
<thead>
<tr>
<th>S</th>
<th>E(\text{LLR})</th>
<th>\sigma(\text{LLR})</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>1.0</td>
<td>0.158</td>
<td>0.555</td>
</tr>
<tr>
<td>2.0</td>
<td>0.390</td>
<td>0.889</td>
</tr>
<tr>
<td>3.0</td>
<td>0.630</td>
<td>1.158</td>
</tr>
<tr>
<td>4.0</td>
<td>0.868</td>
<td>1.398</td>
</tr>
<tr>
<td>5.0</td>
<td>1.104</td>
<td>1.621</td>
</tr>
<tr>
<td>6.0</td>
<td>1.337</td>
<td>1.837</td>
</tr>
<tr>
<td>7.0</td>
<td>1.568</td>
<td>2.049</td>
</tr>
<tr>
<td>8.0</td>
<td>1.797</td>
<td>2.259</td>
</tr>
<tr>
<td>9.0</td>
<td>2.024</td>
<td>2.468</td>
</tr>
<tr>
<td>10.0</td>
<td>2.250</td>
<td>2.676</td>
</tr>
</tbody>
</table>

Table 3.6

Once again, the mean and variance have decreased due to the unknown background level. The fluctuating target is still slightly harder to detect than the constant strength target.

### 3.1.5 Unknown Signal Level

If the target strength is unknown, we can perform our processing assuming some target strength. If that assumption is incorrect then the processing will be suboptimal. This section looks at the degree to which the processor is suboptimal for a constant strength target which is processed as a constant strength.

**Constant Target Strength.** Define

\[ T_s = \text{true target strength}, \text{ and} \]

\[ T_p = \text{target intensity assumed in the processor}. \]
Then the statistic computed to be the log-likelihood ratio is

\[
LLR_{stat}(x, B, T_p) = -\frac{T_p}{B} + \log I_0 \left( 2 \sqrt{\frac{T_p}{B}} \sqrt{\frac{x}{B}} \right)
\]

The “stat” subscript is to emphasize the fact that the quantity computed is not the true log-likelihood unless the assumed strength matches the true strength. We are interested in the behavior of this statistic under the true target distribution. We find that

\[
E(LLR_{stat}) = \int_0^\infty \left\{ -\frac{T_p}{B} + \log I_0 \left( 2 \sqrt{\frac{T_p}{B}} \sqrt{\frac{x}{B}} \right) \right\} \left\{ \frac{1}{B} \exp \left( -\frac{x+T_p}{B} \right) I_0 \left( 2 \sqrt{\frac{T_p}{B}} \sqrt{\frac{x}{B}} \right) \right\} dx
\]

A similar expression holds for the second moment. These are computed numerically with the results in the following table. \(T_p\) is constant in a row and \(T_p\) is constant in a column. The two entries in each square are the expected value and the standard deviation. The shaded squares on the diagonal are optimal processors with the assumed level equal to the true level.

<table>
<thead>
<tr>
<th>Process Truth</th>
<th>1.0</th>
<th>2.0</th>
<th>3.0</th>
<th>4.0</th>
<th>5.0</th>
<th>6.0</th>
<th>7.0</th>
<th>8.0</th>
<th>9.0</th>
<th>10.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>0.320</td>
<td>0.188</td>
<td>-0.109</td>
<td>-0.500</td>
<td>-0.955</td>
<td>-1.457</td>
<td>-1.995</td>
<td>-2.562</td>
<td>-3.153</td>
<td>-3.765</td>
</tr>
<tr>
<td></td>
<td>0.898</td>
<td>1.394</td>
<td>1.776</td>
<td>2.098</td>
<td>2.381</td>
<td>2.638</td>
<td>2.873</td>
<td>3.092</td>
<td>3.298</td>
<td>3.492</td>
</tr>
<tr>
<td>2.0</td>
<td>0.823</td>
<td>0.962</td>
<td>0.874</td>
<td>0.658</td>
<td>0.358</td>
<td>-0.004</td>
<td>-0.414</td>
<td>-0.862</td>
<td>-1.341</td>
<td>-1.847</td>
</tr>
<tr>
<td></td>
<td>1.034</td>
<td>1.574</td>
<td>1.988</td>
<td>2.337</td>
<td>2.643</td>
<td>2.921</td>
<td>3.175</td>
<td>3.412</td>
<td>3.635</td>
<td>3.845</td>
</tr>
<tr>
<td>3.0</td>
<td>1.275</td>
<td>1.648</td>
<td>1.738</td>
<td>1.672</td>
<td>1.504</td>
<td>1.261</td>
<td>0.960</td>
<td>0.615</td>
<td>0.231</td>
<td>-0.185</td>
</tr>
<tr>
<td></td>
<td>1.113</td>
<td>1.672</td>
<td>2.100</td>
<td>2.460</td>
<td>2.777</td>
<td>3.064</td>
<td>3.327</td>
<td>3.572</td>
<td>3.802</td>
<td>4.019</td>
</tr>
<tr>
<td>4.0</td>
<td>1.688</td>
<td>2.266</td>
<td>2.513</td>
<td>2.580</td>
<td>2.528</td>
<td>2.390</td>
<td>2.186</td>
<td>1.930</td>
<td>1.631</td>
<td>1.295</td>
</tr>
<tr>
<td></td>
<td>1.197</td>
<td>1.771</td>
<td>2.209</td>
<td>2.578</td>
<td>2.903</td>
<td>3.197</td>
<td>3.466</td>
<td>3.718</td>
<td>3.953</td>
<td>4.176</td>
</tr>
<tr>
<td>7.0</td>
<td>2.761</td>
<td>3.850</td>
<td>4.488</td>
<td>4.883</td>
<td>5.121</td>
<td>5.245</td>
<td>5.282</td>
<td>5.249</td>
<td>5.160</td>
<td>5.024</td>
</tr>
<tr>
<td></td>
<td>1.266</td>
<td>1.846</td>
<td>2.290</td>
<td>2.664</td>
<td>2.993</td>
<td>3.290</td>
<td>3.564</td>
<td>3.819</td>
<td>4.058</td>
<td>4.284</td>
</tr>
<tr>
<td></td>
<td>1.276</td>
<td>1.856</td>
<td>2.300</td>
<td>2.675</td>
<td>3.005</td>
<td>3.303</td>
<td>3.577</td>
<td>3.832</td>
<td>4.071</td>
<td>4.298</td>
</tr>
</tbody>
</table>

Table 3.7
The means are a weak function of assumed target strength. The only times when the means become negative are when the assumed strength is much greater than the true strength. This is understandable intuitively since in that case the processor is looking for a strong target and the data, even with the target present, does not contain a strong signal.

Another interesting point that the chart quantifies is the loss due to using a single assumed target strength, set to the minimum detectable signal. For example, suppose that we have determined that $S=2$ is the minimum detectable signal, corresponding to a log-likelihood accumulation of approximately one unit per scan. If the true target strength were say 5, then an optimal processor tuned to that strength would accumulate at a rate of 3.5 log-likelihood units per scan. The processor with assumed strength 2 would accumulate at 2.8 units per scan. This is clearly not optimal, but it is sufficient since we needed only one unit per scan to detect.

**Fluctuating Target Strength.** This can be done analytically since log-likelihood is linear in the data. The formulas are:

$$E(LLR)_i = \left( \frac{B + T_x}{B + T_p} \right) \frac{T_p}{B} - \log \left( 1 + \frac{T_p}{B} \right)$$

$$Var(LLR)_i = \left( \frac{B + T_x}{B + T_p} \right)^2 \left( \frac{T_p}{B} \right)^2$$

Compared to the constant strength mismatched processor we see that the fluctuating model is less sensitive to overestimating the target strength. For example, if the true strength is 2 and the assumed strength is 10, then the constant processor shows a log-likelihood loss of 1.8 units per scan, while the fluctuating processor has a gain of 0.33 units per scan. For other mismatches, even in the other direction of underestimating the strength, the fluctuating model fares better. It is only when the assumptions match the truth that the constant target is easier to detect.
<table>
<thead>
<tr>
<th>Process Truth</th>
<th>1.0</th>
<th>2.0</th>
<th>3.0</th>
<th>4.0</th>
<th>5.0</th>
<th>6.0</th>
<th>7.0</th>
<th>8.0</th>
<th>9.0</th>
<th>10.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>0.307</td>
<td>0.235</td>
<td>0.114</td>
<td>-0.009</td>
<td>-0.125</td>
<td>-0.232</td>
<td>-0.329</td>
<td>-0.419</td>
<td>-0.503</td>
<td>-0.580</td>
</tr>
<tr>
<td>1.000</td>
<td>1.333</td>
<td>1.500</td>
<td>1.667</td>
<td>1.714</td>
<td>1.750</td>
<td>1.778</td>
<td>1.800</td>
<td>1.818</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.0</td>
<td>0.807</td>
<td>0.901</td>
<td>0.864</td>
<td>0.791</td>
<td>0.708</td>
<td>0.626</td>
<td>0.546</td>
<td>0.469</td>
<td>0.397</td>
<td>0.329</td>
</tr>
<tr>
<td>2.000</td>
<td>2.250</td>
<td>2.400</td>
<td>2.500</td>
<td>2.571</td>
<td>2.625</td>
<td>2.667</td>
<td>2.700</td>
<td>2.727</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.0</td>
<td>1.307</td>
<td>1.586</td>
<td>1.614</td>
<td>1.591</td>
<td>1.543</td>
<td>1.483</td>
<td>1.421</td>
<td>1.358</td>
<td>1.297</td>
<td>1.238</td>
</tr>
<tr>
<td>2.000</td>
<td>2.667</td>
<td>3.000</td>
<td>3.200</td>
<td>3.333</td>
<td>3.429</td>
<td>3.500</td>
<td>3.556</td>
<td>3.600</td>
<td>3.636</td>
<td></td>
</tr>
<tr>
<td>4.0</td>
<td>1.807</td>
<td>2.235</td>
<td>2.364</td>
<td>2.391</td>
<td>2.375</td>
<td>2.340</td>
<td>2.296</td>
<td>2.247</td>
<td>2.197</td>
<td>2.148</td>
</tr>
<tr>
<td>5.000</td>
<td>3.000</td>
<td>4.000</td>
<td>4.500</td>
<td>4.800</td>
<td>5.143</td>
<td>5.250</td>
<td>5.333</td>
<td>5.400</td>
<td>5.455</td>
<td></td>
</tr>
<tr>
<td>8.0</td>
<td>3.807</td>
<td>4.901</td>
<td>5.364</td>
<td>5.591</td>
<td>5.708</td>
<td>5.768</td>
<td>5.796</td>
<td>5.803</td>
<td>5.797</td>
<td>5.784</td>
</tr>
<tr>
<td>8.000</td>
<td>4.500</td>
<td>6.000</td>
<td>6.750</td>
<td>7.200</td>
<td>7.500</td>
<td>7.714</td>
<td>7.875</td>
<td>8.000</td>
<td>8.100</td>
<td>8.182</td>
</tr>
<tr>
<td>9.000</td>
<td>5.000</td>
<td>6.667</td>
<td>7.500</td>
<td>8.000</td>
<td>8.333</td>
<td>8.571</td>
<td>8.750</td>
<td>8.889</td>
<td>9.000</td>
<td>9.091</td>
</tr>
</tbody>
</table>

Table 3.8

3.1.6 Target Capture Loss

Target capture refers to the corruption of the clutter estimation procedure by the presence of the target. The detrimental effect of including the target in the clutter estimate is easy to understand intuitively. The target is generally brighter (has higher intensity) than the clutter. Large likelihood values obtain when the radar return is large relative to the estimated clutter level. By incorrectly elevating the clutter estimate, the target appears less bright relative to the clutter estimate, and the measurement likelihood value is decreased.

In order to quantify this loss, we consider the fluctuating target case from Section 3.1.2. Let \( B \) be the true background (clutter plus system noise) and let \( T \) be the target intensity. In the process of estimating the clutter background, assume that some fraction \( f \) of the target return is included. Let \( B' = B + fT \). We will then compute the log-likelihood ratio assuming a
background level of $B'$ while the true background level is $B$. The processor is then mismatched to the true statistics, and we have a case treated in Section 3.1.5. We find that the expected log-likelihood ratio accumulation is given by

$$E(\text{LLR}_\text{true}) = \left( \frac{S}{1 + fS} \right) \left( \frac{1 + S}{1 + (1 + f)S} \right) - \log \left( \frac{1 + (1 + f)S}{1 + fS} \right).$$

When $f$ is zero this expression reduces to the accumulation rate calculated in Section 3.1.2.

Figure 3.1 shows the expected accumulation rates for five capture fractions $f$ ranging from 5% to 25% over a range of target strengths between 0 and 10 compared to the accumulation rate with no target capture effect, shown as the top curve. The losses due to target capture can be appreciable. For example, in the extreme case plotted with a target strength of 10 and a capture fraction of 25%, the expected log-likelihood is 0.98 versus 7.60 with no target capture.

### 3.1.7 Conclusions

This section has presented a method for determining the expected log-likelihood accumulation rate and its variance over a target track as a function of target strength in a number of processing scenarios and processors. The two target models considered were a constant strength and an exponentially fluctuating mean. Optimal likelihood processors were considered for these cases, as well as mismatches in the target strength and the processor type. Additionally two cases were studied where the mean background was unknown and the processor included this information. The mean and variance were calculated numerically when necessary, and in a few cases analytic formulas were derived.

The fluctuating target is slightly harder to detect than the constant strength target when the two mean intensities agree. The log-likelihood accumulates a bit slower for the fluctuating target, but the variance is much greater. The effect of a larger variance is that in the regime of high probabilities of detection the threshold must be reduced for the fluctuating target. In a possible UCTP scenario with 50 scans and a log-likelihood threshold of 50 units, the target strength must be at least 2 dB above the system noise to detect with 50% probability. In a real situation, with strong clutter present, the number is higher, but this is an absolute minimum.

The effect of an uncertain background level is to decrease the log-likelihood accumulation quite a bit, but the variance decreases significantly as well. The fluctuating target is still harder to detect than the constant target. In the UCTP scenario considered above, a
fluctuating target must be at least 6.5 dB above the system noise for a 50% probability of detection.

Processing a constant strength target with an incorrect signal level is slightly suboptimal. If the assumed signal level is much greater than the truth, then the log-likelihood accumulation can actually be negative since the data, even with a target present, looks more like the noise than the assumed target signature. This effect is more pronounced for the constant strength processing; the fluctuating model does not degrade as quickly. If the assumed level is smaller than the true value, the log-likelihood will grow slower, but still faster than a weaker target exactly matched to the processor. This supports the use of a processor designed for the “minimum detectable signal.”

If the target fluctuation model assumption is incorrect, then the processor is also slightly suboptimal, on the order of 5-20%. Processing a fluctuating target as if it were constant shows greater losses than the opposite of processing a constant strength target as if it were fluctuating. The fluctuating target model is linear in the data, so there is a computational savings if this is used.

3.2 Log-likelihood Accumulation on a Track

Analysis in the previous section investigated the average rate of log-likelihood accumulation and its variance over a target track. That was for a single scan, or else for a track where the background level is constant. In this section we investigate the case where the mean clutter level is widely varying. The point statistics of the clutter mean are gamma distributed. For clutter scenes of interest the spikiness parameter of the gamma distribution is quite small, on the order of 0.01–0.05. This means that the tails are significant, and there is substantial mass near zero to balance the large excursions. A rough estimate of log-likelihood accumulation can be obtained be dividing the target strength by the sum of the system noise and the clutter mean. With the spiky distribution of clutter mean, this underestimates the accumulation because a single high clutter value will stall the log-likelihood accumulation, but there are many low clutter values which balance the high spike, and during this time the log-likelihood accumulates faster than expected. This section quantifies this effect. Three models are introduced for the clutter encountered along a track. For each of these, expressions for the rate and variance of track log-likelihood are derived. Numerical integrations are used to
evaluate those quantities as a function of clutter spikiness for three signal levels. Tables of probability of detection versus threshold are presented for the three signal levels and three models. The PD is computed based on the assumption that the sum of the measurement log-likelihood ratios over 50 scans is approximately Gaussian.

The two-scale model for radar returns has a slowly varying mean process which is then compounded with speckle. The point statistics of the underlying mean are gamma distributed. The standard gamma distribution is

\[ p(x) = \frac{1}{\Gamma(a)} x^{a-1} e^{-x} \]

which has mean and variance both equal to \( a \). The parameter \( a \) determines the shape of the distribution: when \( a < 1 \), the distribution is singular at zero and has long tails; when \( a = 1 \), the distribution is the exponential; when \( a > 1 \) the distribution becomes more bell shaped; and in the scaled limit, \( a \to \infty \) the distribution approaches a delta function. For the clutter scenes of interest, \( a \) is close to zero, making the distribution very spiky.

In the previous section we examined the statistics of the measurement log-likelihood ratio as a function of normalized target strength in a number of scenarios. The scenarios included two models for target strength: constant and exponentially fluctuating. The background was either known exactly or else was drawn from a five-point distribution. The processors examined were with the correct likelihood ratio processors for the target and noise models, as well as mismatches due to incorrect target strength, or else an incorrect strength model. The result in each case was a table showing the mean and standard deviation of the measurement log-likelihood statistics. Some conclusions were drawn about the minimum signal necessary to accumulate one log-likelihood unit per scan, assuming a threshold of approximately 50 to be reached in 50 scans. These minimum signals were expressed as signal strength relative to the system noise, since, with a varying background, measuring signal strength relative to the sum of clutter mean and system noise would underestimate the detectability.

### 3.2.1 Three Models of Clutter Encountered

As a target moves across a clutter scene it will encounter varying clutter levels. We wish to model how that level varies in order to accurately assess the total accumulation of log-likelihood over the track. If the background is constant, then the result is easy since each scan
will produce an independent draw of the log-likelihood, so the statistics of the sum are easily determined.

The first model randomly chooses a background level \( x \) and then for \( N \) scans simulates the log-likelihood assuming that background. The second model assumes that on each scan the background level is chosen independently of the previous choices. An intermediate model does things in blocks: in each of \( M \) blocks a background level is chosen for that block, and between blocks the background is chosen independently. For obtaining high PD's, the first model is the most conservative since the target may find itself confronted with a high clutter level for the whole track, making it almost certainly undetectable for the parameters we are using. The PD will be limited by the occurrence of these difficult situations, much less than by the average rate of accumulation. We will now derive the statistics of the sum of the log-likelihoods for these models.

We first introduce some notation:

\[ p(x) \] is the distribution of the clutter mean

\[ \mu_1(x) \] is the mean of the log-likelihood, given the clutter mean \( x \)

\[ \mu_2(x) \] is the second (non-central) moment of the log-likelihood, given the clutter mean \( x \)

\( \nu(x) \) is the variance of the LLR, given the clutter mean \( x \)

\[ \nu(x) = \mu_2(x) - \mu_1^2(x) \]

\( N \) is the number of scans

**Model 1.** We compute the moments of the measurement log-likelihood sum for the \( N \) scans. Given a choice of \( x \), the moments of the sum are

\[ \mu_1(x) = N \mu_1(x) \]

\[ \nu(x) = N \nu(x) = N \mu_2(x) - N \mu_1^2(x) \]

\[ \mu_2(x) = \nu(x) + \mu_1^2(x) \]

\[ = N \mu_2(x) + N(N - 1) \mu_1^2(x) \]
This follows since the mean of the sum is the sum of the means, and the variance of the sum is the sum of the variances. Now we consider the fact that $x$ is chosen randomly, and we compute the moments after this randomization to get:

$$\mu_1 = \int \mu_1(x) p(x) dx$$
$$= N \int \mu_1(x) p(x) dx$$

$$\mu_2 = \int \mu_2(x) p(x) dx$$
$$= N \int \mu_2(x) p(x) dx + N(N - 1) \int \mu_1^2(x) p(x) dx$$

$$v = \mu_2 - \mu_1^2$$
$$= N \int \mu_2(x) p(x) dx + N(N - 1) \int \mu_1^2(x) p(x) dx - N^2 \left[ \int \mu_1(x) p(x) dx \right]^2$$

$$= N \left\{ \int \mu_2(x) p(x) dx - \left[ \int \mu_1(x) p(x) dx \right]^2 \right\} + N(N - 1) \left\{ \int \mu_1^2(x) p(x) dx - \left[ \int \mu_1(x) p(x) dx \right]^2 \right\}$$

**Model 2.** In the second model, consider the measurement log-likelihood on the first scan. It is the same as the previous model with $N=1$, so we know its mean and variance. The mean and variance of the sum of $N$ independent random variables of the same kind is the sum of the means and variances, so for this model:

$$\mu_1 = N \int \mu_1(x) p(x) dx$$

$$v = N \left\{ \int \mu_2(x) p(x) dx - \left[ \int \mu_1(x) p(x) dx \right]^2 \right\}$$

As expected, the first model has a higher variance in the log-likelihood sum. Intuitively, this is because model 1 allows the target to trace a path through very high clutter for its entire length; it also allows paths through very low clutter. The second model assumes that the target will go through independent clutter which avoids the extreme log-likelihood sums by averaging them with other less extreme returns.

**Model 3.** The third model is a hybrid of the first two and may be more realistic for the correlation structure in the clutter scenes. It assumes that the string of $N$ scans is broken into $M$ blocks with $M'$ scans each (so $N = MM'$). Within each block the clutter mean is chosen
independently, and then maintained for the entire block. In that case reasoning similar to the above yields the following:

$$\mu_1 = N \int \mu_1(x)p(x)dx$$

$$\nu = N \left\{ \int \mu_2(x)p(x)dx - \left[ \int \mu_1(x)p(x)dx \right]^2 \right\}$$

$$+ N(M' - 1) \left\{ \int \mu_1^2(x)p(x)dx - \left[ \int \mu_1(x)p(x)dx \right]^2 \right\}$$

This model considers the fact that a target may go into a clutter patch and stay there for a number of scans. A reasonable value for $M$ appears to be 10.

### 3.2.2 Numerical Results

Looking at the above expressions, we see there are three integrals to be done numerically in order to compute the mean and variance of the sum of the log-likelihoods across a complete track. The functions $\mu_1(x)$, $\nu(x)$, and $\mu_2(x)$ are computed by interpolating the tables discussed earlier. Let us define

$$M_1 = \int \mu_1(x)p(x)dx$$

$$V = \int \mu_2(x)p(x)dx - \left[ \int \mu_1(x)p(x)dx \right]^2$$

$$V_m = \int \mu_1^2(x)p(x)dx - \left[ \int \mu_1(x)p(x)dx \right]^2.$$  

Given these values we can compute the mean and variance of the log-likelihood sum for any number of scans for each of the three models above.

If the spikiness parameter were greater than one then it would be possible to use a trapezoidal rule and get an accurate result. However, when the spikiness parameter is less than one the integral is singular at zero. In that case a transformation of variables is used to eliminate the singularity and the resulting integral is done with a trapezoidal rule.

There are two parameters in addition to the spikiness parameter, $a$. They are the clutter-to-noise ratio (CNR) and the signal-to-clutter ratio (SCR). Given values of the these parameters and a clutter mean level of $x$, then the normalized signal strength is
\[
S = \frac{a \cdot SCR}{x + \frac{a}{CNR}} = \frac{SNR}{1 + CNR \cdot \frac{x}{a}}
\]

We use this magnitude to access the previously computed table.

The test cases that we have examined have the CNR and SCR set to

\[
\begin{align*}
CNR &= 6 \text{ dB} = 3.98 \\
SCR &= 5 \text{ dB} = 3.16
\end{align*}
\]

In addition, results are presented below for weaker targets with SCR’s

\[
\begin{align*}
SCR &= 2.5 \text{ dB} = 1.78 \\
SCR &= 0 \text{ dB} = 1.00
\end{align*}
\]

There are four clutter scenes simulated by DTI. The spikiness parameters for those four scenes are:

- 10 kt. upwind: \( a = 0.01156 \)
- 10 kt. crosswind: \( a = 0.01372 \)
- 18 kt. upwind: \( a = 0.04355 \)
- 18 kt. crosswind: \( a = 0.05977 \)

Here are the results for the case where the background is unknown and the target is fluctuating. For each of the three signal levels, the quantities \( M_i, V, \) and \( V_m \) defined above are listed for each of the four clutter scenes. Then the mean and standard deviation of the sum of the measurement log-likelihoods is shown for each of the three models assuming 50 scans. The third model assumes that there are 5 blocks of length 10 scans each. The mean is listed only once since it is the same for all three models.
<table>
<thead>
<tr>
<th></th>
<th>10 kt. upwind</th>
<th>10 kt. cross</th>
<th>18 kt. upwind</th>
<th>18 kt. cross</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>5 dB</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$M_1$</td>
<td>2.67</td>
<td>2.64</td>
<td>2.38</td>
<td>2.27</td>
</tr>
<tr>
<td>$V$</td>
<td>9.66</td>
<td>9.61</td>
<td>8.99</td>
<td>8.68</td>
</tr>
<tr>
<td>$V_m$</td>
<td>0.37</td>
<td>0.42</td>
<td>0.82</td>
<td>0.93</td>
</tr>
<tr>
<td>$\mu_1$</td>
<td>133.5</td>
<td>132.0</td>
<td>119.0</td>
<td>113.5</td>
</tr>
<tr>
<td>Model 1 $\sqrt{v}$</td>
<td>37.28</td>
<td>38.85</td>
<td>49.58</td>
<td>52.08</td>
</tr>
<tr>
<td>Model 2 $\sqrt{v}$</td>
<td>21.98</td>
<td>21.92</td>
<td>21.20</td>
<td>20.83</td>
</tr>
<tr>
<td>Model 3 $\sqrt{v}$</td>
<td>25.49</td>
<td>25.87</td>
<td>28.61</td>
<td>29.20</td>
</tr>
<tr>
<td><strong>2.5 dB</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$M_1$</td>
<td>1.49</td>
<td>1.47</td>
<td>1.33</td>
<td>1.27</td>
</tr>
<tr>
<td>$V$</td>
<td>4.11</td>
<td>4.09</td>
<td>3.79</td>
<td>3.64</td>
</tr>
<tr>
<td>$V_m$</td>
<td>0.12</td>
<td>0.13</td>
<td>0.26</td>
<td>0.30</td>
</tr>
<tr>
<td>$\mu_1$</td>
<td>74.5</td>
<td>73.5</td>
<td>66.5</td>
<td>63.5</td>
</tr>
<tr>
<td>Model 1 $\sqrt{v}$</td>
<td>22.35</td>
<td>22.87</td>
<td>28.75</td>
<td>30.28</td>
</tr>
<tr>
<td>Model 2 $\sqrt{v}$</td>
<td>14.34</td>
<td>14.30</td>
<td>13.77</td>
<td>13.49</td>
</tr>
<tr>
<td>Model 3 $\sqrt{v}$</td>
<td>16.11</td>
<td>16.22</td>
<td>17.51</td>
<td>17.80</td>
</tr>
<tr>
<td></td>
<td>10 kt. upwind</td>
<td>10 kt. cross</td>
<td>18 kt. upwind</td>
<td>18 kt. cross</td>
</tr>
<tr>
<td>----------</td>
<td>--------------</td>
<td>-------------</td>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>0 dB</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$M_1$</td>
<td>0.81</td>
<td>0.80</td>
<td>0.72</td>
<td>0.69</td>
</tr>
<tr>
<td>$V$</td>
<td>1.86</td>
<td>1.84</td>
<td>1.69</td>
<td>1.62</td>
</tr>
<tr>
<td>$V_m$</td>
<td>0.04</td>
<td>0.04</td>
<td>0.08</td>
<td>0.09</td>
</tr>
<tr>
<td>$\mu$</td>
<td>40.5</td>
<td>40.0</td>
<td>36.0</td>
<td>34.5</td>
</tr>
<tr>
<td>Model 1 $\sqrt{v}$</td>
<td>13.82</td>
<td>13.78</td>
<td>16.75</td>
<td>17.36</td>
</tr>
<tr>
<td>Model 2 $\sqrt{v}$</td>
<td>9.64</td>
<td>9.59</td>
<td>9.19</td>
<td>9.00</td>
</tr>
<tr>
<td>Model 3 $\sqrt{v}$</td>
<td>10.54</td>
<td>10.49</td>
<td>10.98</td>
<td>11.02</td>
</tr>
</tbody>
</table>

Table 3.9

We now use these results to compute PD as a function of threshold setting. We assume that the sum of 50 random variables is approximately Gaussian and use the cumulative distribution of the normal to compute the probability that the statistics will be above a threshold. We present charts here for the case where the threshold varies from 25 to 50 log-likelihood units. This is the approximate range of interest based on operationally meaningful false alarm rates and an empirical study of clutter peak values. Additional values can be derived easily from a table of the cumulative normal distribution and the parameters computed in the previous section.

5 dB target

Model 1

<table>
<thead>
<tr>
<th></th>
<th>10 kt. upwind</th>
<th>10 kt. cross</th>
<th>18 kt. upwind</th>
<th>18 kt. cross</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>98</td>
<td>98</td>
<td>91</td>
<td>88</td>
</tr>
<tr>
<td>45</td>
<td>99</td>
<td>98</td>
<td>93</td>
<td>90</td>
</tr>
<tr>
<td>40</td>
<td>99</td>
<td>99</td>
<td>94</td>
<td>92</td>
</tr>
<tr>
<td>35</td>
<td>99.5+</td>
<td>99</td>
<td>95</td>
<td>93</td>
</tr>
<tr>
<td>30</td>
<td>99.5+</td>
<td>99.5+</td>
<td>96</td>
<td>94</td>
</tr>
<tr>
<td>25</td>
<td>99.5+</td>
<td>99.5+</td>
<td>97</td>
<td>95</td>
</tr>
</tbody>
</table>
### Model 2

<table>
<thead>
<tr>
<th></th>
<th>10 kt. upwind</th>
<th>10 kt. cross</th>
<th>18 kt. upwind</th>
<th>18 kt. cross</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>99.5+</td>
<td>99.5+</td>
<td>99.5+</td>
<td>99.5+</td>
</tr>
<tr>
<td>45</td>
<td>99.5+</td>
<td>99.5+</td>
<td>99.5+</td>
<td>99.5+</td>
</tr>
<tr>
<td>40</td>
<td>99.5+</td>
<td>99.5+</td>
<td>99.5+</td>
<td>99.5+</td>
</tr>
<tr>
<td>35</td>
<td>99.5+</td>
<td>99.5+</td>
<td>99.5+</td>
<td>99.5+</td>
</tr>
<tr>
<td>30</td>
<td>99.5+</td>
<td>99.5+</td>
<td>99.5+</td>
<td>99.5+</td>
</tr>
<tr>
<td>25</td>
<td>99.5+</td>
<td>99.5+</td>
<td>99.5+</td>
<td>99.5+</td>
</tr>
</tbody>
</table>

### Model 3

<table>
<thead>
<tr>
<th></th>
<th>10 kt. upwind</th>
<th>10 kt. cross</th>
<th>18 kt. upwind</th>
<th>18 kt. cross</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>99.5+</td>
<td>99.5+</td>
<td>99</td>
<td>98</td>
</tr>
<tr>
<td>45</td>
<td>99.5+</td>
<td>99.5+</td>
<td>99.5+</td>
<td>99</td>
</tr>
<tr>
<td>40</td>
<td>99.5+</td>
<td>99.5+</td>
<td>99.5+</td>
<td>99</td>
</tr>
<tr>
<td>35</td>
<td>99.5+</td>
<td>99.5+</td>
<td>99.5+</td>
<td>99.5+</td>
</tr>
<tr>
<td>30</td>
<td>99.5+</td>
<td>99.5+</td>
<td>99.5+</td>
<td>99.5+</td>
</tr>
<tr>
<td>25</td>
<td>99.5+</td>
<td>99.5+</td>
<td>99.5+</td>
<td>99.5+</td>
</tr>
</tbody>
</table>

### 2.5 dB target

### Model 1

<table>
<thead>
<tr>
<th></th>
<th>10 kt. upwind</th>
<th>10 kt. cross</th>
<th>18 kt. upwind</th>
<th>18 kt. cross</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>86</td>
<td>84</td>
<td>71</td>
<td>67</td>
</tr>
<tr>
<td>45</td>
<td>90</td>
<td>89</td>
<td>77</td>
<td>72</td>
</tr>
<tr>
<td>40</td>
<td>93</td>
<td>92</td>
<td>82</td>
<td>78</td>
</tr>
<tr>
<td>35</td>
<td>96</td>
<td>95</td>
<td>86</td>
<td>82</td>
</tr>
<tr>
<td>30</td>
<td>97</td>
<td>97</td>
<td>89</td>
<td>86</td>
</tr>
<tr>
<td>25</td>
<td>98</td>
<td>98</td>
<td>92</td>
<td>89</td>
</tr>
</tbody>
</table>
### Model 2

<table>
<thead>
<tr>
<th></th>
<th>10 kt. upwind</th>
<th>10 kt. cross</th>
<th>18 kt. upwind</th>
<th>18 kt. cross</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>95</td>
<td>94</td>
<td>88</td>
<td>84</td>
</tr>
<tr>
<td>45</td>
<td>98</td>
<td>97</td>
<td>94</td>
<td>91</td>
</tr>
<tr>
<td>40</td>
<td>99</td>
<td>99</td>
<td>97</td>
<td>95</td>
</tr>
<tr>
<td>35</td>
<td>99.5+</td>
<td>99.5+</td>
<td>98</td>
<td>98</td>
</tr>
<tr>
<td>30</td>
<td>99.5+</td>
<td>99.5+</td>
<td>99.5+</td>
<td>99</td>
</tr>
<tr>
<td>25</td>
<td>99.5+</td>
<td>99.5+</td>
<td>99.5+</td>
<td>99.5+</td>
</tr>
</tbody>
</table>

### Model 3

<table>
<thead>
<tr>
<th></th>
<th>10 kt. upwind</th>
<th>10 kt. cross</th>
<th>18 kt. upwind</th>
<th>18 kt. cross</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>93</td>
<td>92</td>
<td>82</td>
<td>77</td>
</tr>
<tr>
<td>45</td>
<td>96</td>
<td>96</td>
<td>89</td>
<td>85</td>
</tr>
<tr>
<td>40</td>
<td>98</td>
<td>98</td>
<td>93</td>
<td>90</td>
</tr>
<tr>
<td>35</td>
<td>99</td>
<td>99</td>
<td>96</td>
<td>94</td>
</tr>
<tr>
<td>30</td>
<td>99.5+</td>
<td>99.5+</td>
<td>98</td>
<td>97</td>
</tr>
<tr>
<td>25</td>
<td>99.5+</td>
<td>99.5+</td>
<td>99</td>
<td>98</td>
</tr>
</tbody>
</table>

### 0 dB target

### Model 1

<table>
<thead>
<tr>
<th></th>
<th>10 kt. upwind</th>
<th>10 kt. cross</th>
<th>18 kt. upwind</th>
<th>18 kt. cross</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>24</td>
<td>23</td>
<td>20</td>
<td>18</td>
</tr>
<tr>
<td>45</td>
<td>37</td>
<td>35</td>
<td>29</td>
<td>27</td>
</tr>
<tr>
<td>40</td>
<td>51</td>
<td>50</td>
<td>40</td>
<td>37</td>
</tr>
<tr>
<td>35</td>
<td>65</td>
<td>64</td>
<td>52</td>
<td>48</td>
</tr>
<tr>
<td>30</td>
<td>77</td>
<td>76</td>
<td>63</td>
<td>60</td>
</tr>
<tr>
<td>25</td>
<td>86</td>
<td>86</td>
<td>74</td>
<td>70</td>
</tr>
</tbody>
</table>
Model 2

<table>
<thead>
<tr>
<th></th>
<th>10 kt. upwind</th>
<th>10 kt. cross</th>
<th>18 kt. upwind</th>
<th>18 kt. cross</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>16</td>
<td>14</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>45</td>
<td>32</td>
<td>30</td>
<td>16</td>
<td>12</td>
</tr>
<tr>
<td>40</td>
<td>52</td>
<td>50</td>
<td>33</td>
<td>27</td>
</tr>
<tr>
<td>35</td>
<td>71</td>
<td>69</td>
<td>54</td>
<td>47</td>
</tr>
<tr>
<td>30</td>
<td>86</td>
<td>85</td>
<td>74</td>
<td>69</td>
</tr>
<tr>
<td>25</td>
<td>94</td>
<td>94</td>
<td>88</td>
<td>85</td>
</tr>
</tbody>
</table>

Model 3

<table>
<thead>
<tr>
<th></th>
<th>10 kt. upwind</th>
<th>10 kt. cross</th>
<th>18 kt. upwind</th>
<th>18 kt. cross</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>18</td>
<td>17</td>
<td>10</td>
<td>7</td>
</tr>
<tr>
<td>45</td>
<td>33</td>
<td>31</td>
<td>20</td>
<td>17</td>
</tr>
<tr>
<td>40</td>
<td>51</td>
<td>50</td>
<td>35</td>
<td>30</td>
</tr>
<tr>
<td>35</td>
<td>69</td>
<td>68</td>
<td>53</td>
<td>48</td>
</tr>
<tr>
<td>30</td>
<td>84</td>
<td>82</td>
<td>70</td>
<td>65</td>
</tr>
<tr>
<td>25</td>
<td>92</td>
<td>92</td>
<td>84</td>
<td>80</td>
</tr>
</tbody>
</table>

Table 3.10

In the previous section we investigated the average rate of log-likelihood accumulation and its variance over a target track. Those results were appropriate for a target that encounters a constant estimated clutter mean level. When the clutter mean is varying it is possible to compute the target strength relative to the system noise or else relative to the sum of the system noise and the average clutter mean level. The first method is a weak upper bound on detection performance, but is useful for some studies. The second method underestimates performance, quite significantly in spiky clutter. The current work has quantified the effect of a varying clutter level on detection performance. Three models were presented for the type of clutter encountered along a track.

Numerical results were computed for the expected log-likelihood accumulation for three target strengths in the four clutter scenes generated by DTI. For fixed SCR and CNR, the average performance is better in spikier clutter. The 10 kt scenes are spikier than the 18 kt. scenes, and the cross wind scenes are slightly spikier than the upwind scenes. For a CNR of 6 dB and a SCR of 5 dB, the average accumulation over 50 scans ranges from 113 to 133 units. This is much above the clutter peaks encountered empirically. The likelihood ratios in the
runs done so far are approximately \(5 \times 10^8\) due to that setting for the probability of a target appearing in a cell on each scan. This is approximately \(-17\) log-likelihood units. With this as the floor, the target peak should accumulate to between \(10^{40}\) and \(10^{50}\). From this analysis, this target should be very easy to detect in all four clutter scenes.

The 2.5 dB target average accumulation is between 63 and 74 for the 50 scans. This is high enough above the clutter that this target should be detectable more than 50% of the time based on this analysis. The standard deviation of the aggregate log-likelihood is significant enough that extremely high PD’s are not expected, especially in the clutter scene with 18 kt. winds and a crosswind look direction.

The 0 dB target’s average accumulation is between 34 and 40. These are close to the thresholds examined to limit false alarms to 1 per hour in a circle out to 10 nautical miles. This analysis indicates that these targets will be detected less than 50% of the time, and often only 25%. Weaker targets would have even a lower PD.

### 3.3 Clutter Peak Analysis

This section presents the results of analysis of local peak values produced by runs with four large simulated clutter scenes. The results are of interest since the target detection process depends on targets producing local peaks which are high compared to the peaks produced by clutter.

#### 3.3.1 Simulated Data Scenes Considered

DTI provided four sets of radar returns. Each consisted of 10,240 range bins (in one direction) over 512 scans. Each range bin is 0.3 meters and scans occur at 0.2 second intervals. The first case has wind of 10 knots blowing toward the radar. The second has a 10 knot wind blowing across the sensor line of sight. The third and fourth cases have winds of 18 knots blowing toward the sensor and across the sensor line of sight respectively.

#### 3.3.2 Processing Methodology

The files were each processed through the UCTP system consisting of the clutter tracker and the LRT. The complete LRT state space at each time step was saved. This state space is 10,240 range cells by 100 velocity cells. There were 512 times steps. Thus the output
file consisted of 524,288,000 real numbers. With each real number requiring four bytes, the result is a two gigabyte file. These files were saved on tape for future reference.

Using the complete state space dump, local peaks are detected by a separate program. The output file can be considered a function of three positive integers, \( p, v \) and \( t \) for position index, velocity index and time index respectively. \( f(p,v,t) \) is defined to be a local peak if

\[
f(p,v,t) \geq f(p',v',t') \forall p, v, t\ s.t. \ p - p' \leq \delta p, |v - v'| \leq \delta v, |t - t'| \leq \delta t
\]

The values for \( \delta p, \delta v \) and \( \delta t \) were 45, 10 and 10 respectively. Values of \( t \) which were within \( \delta t \) of the start or end of the run were not considered. A region of this size is used so that a build up of likelihood is not counted twice.

To elaborate on the last statement, a target is expected to build up likelihood by being reinforced over time with a series of small increases consistent with the motion model. Clutter can build up in a similar manner. For short periods of time, random sequences of returns can be similar to a target. Also a very large return at a single point can be reinforced for several different velocities over the next few time steps, producing local peaks which are very close together.

Files of the natural logs of the peak values together with the associated \((p,v,t)\) were produced for each of the four cases. It is these files which were used in the analysis of the distribution of peak values.

### 3.3.3 Local Peak Distributions

The distribution of the values for the local peaks appeared to be similar to a shifted gamma distribution. An unshifted gamma distribution has density function

\[
f(x) = \begin{cases} 
  x^{\alpha-1} e^{-\frac{x}{\beta}} & x \geq 0 \\
  \frac{1}{\Gamma(\alpha)\beta^\alpha} & x < 0 
\end{cases}
\]

while the shifted version has density

\[
f(x) = \begin{cases} 
  (x - \delta)^{\alpha-1} e^{-\frac{x-\delta}{\beta}} & x \geq \delta \\
  \frac{1}{\Gamma(\alpha)\beta^\alpha} & x < \delta 
\end{cases}
\]
For this analysis, $\delta$ was taken to be (minimum-0.001). Maximum likelihood estimators for $\alpha$ and $\beta$ for an unshifted gamma distribution are given in Kotz and Johnson (1982) as

$$\bar{X} = \sum_{i=1}^{N} X_i / N, \bar{X} = \left( \prod_{i=1}^{N} X_i \right)^{1/N}$$

$$\ln(\alpha) - \Gamma'(\alpha) / \Gamma(\alpha) = \ln(\bar{X} / \bar{X})$$

$$\beta = \bar{X} / \alpha$$

Fortunately, the same reference provides a rational approximation for $\alpha$

$$\alpha = \begin{cases} 
\frac{0.5000876 + 0.1648852 M - 0.0544276 M^2}{M} & 0 \leq M \leq 0.5772 \\
\frac{8.098919 + 9.00950 M + 0.977573 M^2}{M(17.79728 + 11.968547 M + M^2)} & 0.5772 \leq M \leq 17 \\
\frac{M}{M} & M > 17 
\end{cases}$$

$$M = \ln(\bar{X} / \bar{X})$$

Using this method one obtains the values for $\alpha$ and $\beta$ given in Table 2.11 below.

<table>
<thead>
<tr>
<th>Distribution Parameters for Standard Processing</th>
<th>$\alpha$</th>
<th>$\beta$</th>
<th>$N$</th>
<th>min</th>
<th>max</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 kts upwind</td>
<td>5.414</td>
<td>1.123</td>
<td>13511</td>
<td>-13.516</td>
<td>9.208</td>
</tr>
<tr>
<td>10 kts crosswind</td>
<td>5.877</td>
<td>0.990</td>
<td>13712</td>
<td>-13.032</td>
<td>8.874</td>
</tr>
<tr>
<td>18 kts upwind</td>
<td>5.909</td>
<td>0.796</td>
<td>12617</td>
<td>-14.176</td>
<td>3.302</td>
</tr>
<tr>
<td>18 kts crosswind</td>
<td>6.217</td>
<td>0.696</td>
<td>12570</td>
<td>-13.695</td>
<td>4.222</td>
</tr>
</tbody>
</table>

Table 3.11

3.3.4 Extrapolation to Four False Alarm Rates

Once the distribution is determined, it is possible to estimate the peak values likely to occur over longer time periods, more range bins and more directions. This type of extrapolation is, of course, inherently risky. Care should be used in applying these results.

Anderson (1997) provides a mathematical description of a numerical method for use with gamma distributions. There are 102.4 seconds and 10,240 feet in each data set. A single direction over 10 nautical miles is about 60,000 feet or 6 times as large. An hour has 3600 seconds, so another factor of 36 is required. Thus, if there are $N$ local peaks in the data set, there should be about $36*6*N$ peaks in one beam over an hour. The “1/beam/hour” column gives the $1/(36*6*N)$ level of the shifted gamma distribution. Similar calculations are made for
the other columns with the convention that each beam is two degrees and hence there are 180 beams over the coverage area.

<table>
<thead>
<tr>
<th>UCTP Processing Log-Likelihood Peak Levels</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td>1/beam/hour</td>
</tr>
<tr>
<td>1/beam/day</td>
</tr>
<tr>
<td>1/hour</td>
</tr>
<tr>
<td>1/day</td>
</tr>
<tr>
<td>10 kts upwind</td>
</tr>
<tr>
<td>10 kts crosswind</td>
</tr>
<tr>
<td>18 kts upwind</td>
</tr>
<tr>
<td>18 kts crosswind</td>
</tr>
</tbody>
</table>

Table 3.12

3.3.5 Estimated Receiver Operating Characteristics

The Receiver Operating Characteristics (ROC) curve shows the tradeoff between false alarm rate (FAR) and probability of detection (PD) obtained from a processor by varying the threshold level. We can combine the results from Section 3.2 on the log-likelihood accumulated by a target along a track with the results above on the observed clutter peak levels. The result is shown in the Figures 3.2–3.4 for the three different target strengths considered: 5 dB, 2.5 dB, and 0 dB. For each of these, we have graphed the PD vs. FAR for the four clutter scenes studied. Looking at Figure 3.2 we see that the 5 dB target is easily detected (above 99%) across all the scenes, down to the FAR of 1 per day over the whole surveillance region. For the 2.5 dB target (Figure 3.3), we see similar results: PDs above 95% over the whole set of clutter scenes and FARs. With the 0 dB target (Figure 3.4), we see some probability of not detecting the target. In the 10 kt upwind scene, the PD varies from 75% with a FAR of 1 per beam per hour, down to 35% with a FAR of 1 false alarm per day over the whole region. In the 10 kt cross wind scene, the corresponding PDs are 80% and 50%. The two 18 kt scenes show very similar behavior. The PD is 85% for 1 false alarm per beam per hour, down to 65% with 1 false alarm per day over the whole region. All of these analyses assume a 10 second periscope exposure time.

3.4 Log-likelihood Accumulation for Alternate Data Types and Processors

One approach for comparing different processing streams in the problem of radar detection of periscopes against the competing sea clutter and system noise is to examine the
log-likelihood accumulations which would result were one to employ optimally the same data as used in those various processing streams. For example if no estimation of the local clutter is attempted and only the radar intensity values are used, then however they might be used in the actual processing under consideration we can assert that the performance can never be better than what would result from forming a log-likelihood from that data and accumulating it across scans. Similarly, if the intensity values are thresholded against a fixed threshold, producing for each range-azimuth cell a count of threshold exceedances over a given number of trials, then whatever the actual use made of this count it can never achieve better performance than that which would be achieved from forming a proper log-likelihood from the exceedance counts. As a general theme and approach we believe that it is legitimate to compare the average target-present accumulations of log-likelihood associated with each data type (determined by the particular pre-processing, ancillary data used, editing, etc.) employed in a given processing schema: A data type which offers a buildup of a given number of log-likelihood units on average which exceeds another data type by a given amount represents a superior choice, quantitatively expressed by that log-likelihood difference, and this degree of superiority prevails even if additional processing based upon independent features of the observations (such as, in a more specific context, spatial scatter about straight-line paths) is contemplated. This difference from a certain viewpoint is more fundamental than the actual performance differences between the two processing streams given the various degrees of unintentional additional losses in each. A corresponding comparison in terms of detection sensitivity may be formed from this difference as well: If one data type permits a log-likelihood buildup at a given average rate at a lower value of target strength than another type would require to produce the same rate of accumulation then the first type achieves greater detection sensitivity by that difference.

We shall actually consider in this section the analytic forms for the average log-likelihood buildup for five cases. Common to all these cases are the assumptions that the observations are statistically independent and that the system noise, clutter, and target intensities at any point are exponentially distributed and add incoherently; additionally, the mean clutter level point statistics are assumed to be gamma distributed as we shall see presently, and the system noise has a mean value of one. Additional assumptions associated with each case treated will be stated within the further description of those cases. The cases are as follows:

1. No knowledge of the local mean clutter is available, but distribution for the mean clutter is known exactly. The target strength may or may not be known: Both
situations are examined. Given this degree of knowledge, the measurements at each possible target location \( x \) consist of the set of \( N \) independent intensities \( I(x,t) \) observed at discrete times within a time interval during which the target could not have moved outside of the given range-azimuth radar resolution cell.

2. Given the same degree of knowledge as in Case 1, the data are first thresholded against a constant value. The resulting data then consist for each location \( x \) of the number of threshold exceedances observed, \( M(x) \).

3. The local mean clutter \( C(x,t) \) is known exactly through methods neither relevant nor described here. The data consist therefore for each location \( x \) the values \( C(x,t) \) and \( I(x,t) \). Again, the cases where the target strength is known and unknown are both examined.

4. The local mean clutter is known exactly as in Case 3, however the data \( I(x,t) \) are normalized by division by the local quantity \( 1 + C(x,t) \) to arrive at a new set of data \( J(x,t) = I(x,t) / (1 + C(x,t)) \). The local clutter values are then "forgotten", discarded, and the detection decision must be made entirely from the normalized data set.

5. The circumstances are as described in Case 4, however the field \( J(x,t) \) is thresholded and only the number of exceedances \( M(x) \) at each location is available for making the detection decision.

Common Statistical Assumptions

Here we summarize the statistical model already described qualitatively in the introduction. First, the probability density for obtaining a given intensity \( I(x,t) \) is conditioned on the local clutter level \( C(x,t) \) and the target level \( T(x) \) in that cell. Cells containing no target have a target value of zero. The density is therefore given by

\[
L(I(x,t) | C(x,t), T(x)) = \frac{1}{1 + C(x,t) + T(x)} \exp\left(- \frac{I(x,t)}{1 + C(x,t) + T(x)}\right).
\]

Secondly, the clutter mean level is distributed globally according to a gamma distribution:

\[
p(C) = \frac{C^{a-1}}{b^a \Gamma(a)} \exp\left(- \frac{C}{b}\right).
\]
The quantity \( a \) is the “spikiness” parameter, typically ranging from 0.01 to 0.05, and the global mean clutter is given in terms of the parameters \( a \) and \( b \) by the product: \( \overline{C} = ab \). We shall set \( a=0.01 \) in this analysis. The additive system noise is exponential with mean level of one.

A situation which can easily arise is where the different observations result in independent draws on the system noise and the exponential distributions but correlated draws on the local clutter. An exact treatment of this case is not available from the foregoing analysis; however an approximation which is suggested is to simply imagine that we decimate the \( N \) observations so as to arrive at truly independent conditions and use the resulting smaller value of \( N \) as the effective number of observations. This is obviously suboptimal, in that it abandons the advantage of some averaging over the system noise as well as the “speckle noise.” Nonetheless, it might serve to provide an estimate of comparative performances for this situation.

The equations obtained here are most readily employed to determine performance through means of numerical evaluations. These evaluations of numerical consequences will be provided in the following sections wherein each of the above five cases will be treated separately.

### 3.4.1 Case 1: No Clutter Estimation

In the present section we detail the results for the Case 1, a description of which is the following: No knowledge of the local mean clutter is available, but distribution for the mean clutter is known exactly. The target strength may or may not be known, both situations being examined. Given this degree of knowledge, the measurements at each possible target location \( x \) consist of the set of \( N \) independent intensities \( I(x,t) \) observed at discrete times within a time interval during which the target could not have moved outside of the given range-azimuth radar resolution cell.

According to this general model the measurement log-likelihood ratio for Case 1 is given by

\[
\ln \mathcal{L}(I \mid \hat{T}) = \ln \left( \frac{\int_0^\infty dCp(C)L(I \mid C,\hat{T})}{\int_0^\infty dCp(C)L(I \mid C,0)} \right),
\]
where \( \hat{T} \) is the assumed target strength, and the respective likelihoods involved in the ratio are given by

\[
L(I \mid T) = \int_0^\infty dCp(C)L(I \mid C, T) = \int_0^\infty dC \left( \frac{C}{b} \right)^{a-1} \frac{1}{b\Gamma(a)} \exp\left(-\frac{C}{b}\right) \frac{1}{1+C+T} \exp\left(-\frac{I}{1+C+T}\right)
\]

\[
L(I \mid 0) = \int_0^\infty dCp(C)L(I \mid C, 0) = \int_0^\infty dC \left( \frac{C}{b} \right)^{a-1} \frac{1}{b\Gamma(a)} \exp\left(-\frac{C}{b}\right) \frac{1}{1+C} \exp\left(-\frac{I}{1+C}\right).
\]

The average measurement log-likelihood buildup each update for a true target strength of \( T \) is given by

\[
\mathbb{E}[\ln L(I \mid \hat{T}) \mid T] = \int_0^\infty dI L(I \mid T) \ln L(I \mid T).
\]

When the true target strength is known, one sets \( \hat{T} = T \).

The likelihood function \( L(I \mid T) \)

The likelihood function is evaluated by a change of variable. We write \( C = bx^{1/a} \) and change the variable of integration to \( x \). Because the behavior of the integrand with \( x \) is quite different according as whether \( x \) is greater than or less than one, we break the integral into two parts and obtain

\[
L(I \mid T) = \int_0^1 dx \frac{1}{\Gamma(a+1)} \exp\left(-x^{1/a}\right) \frac{1}{1 + bx^{1/a} + T} \exp\left(-\frac{I}{1 + bx^{1/a} + T}\right)
\]

\[
+ \int_1^{x_{\text{max}}} dx \frac{1}{\Gamma(a+1)} \exp\left(-x^{1/a}\right) \frac{1}{1 + bx^{1/a} + T} \exp\left(-\frac{I}{1 + bx^{1/a} + T}\right),
\]

where in the second integral the upper limit \( x_{\text{max}} \), in actual evaluation set to 25, is adequate to obtain sufficient precision of the final answer. Insensitivity of the likelihood functions to the mean clutter value at our assumed small spikiness parameter value is seen in Figure 3.5. This shows the target-absent likelihood function versus measured intensity for two global mean clutter values, 5 and 20.
From the figure we see that the likelihood function indeed does not depend dramatically on the overall mean clutter, the upper curve corresponding to a factor of 4 increase in that quantity compared with the lower one. Accordingly, we shall next show only a single figure of the likelihoods for various target strengths, setting the mean clutter level to 10 for the illustration. Figure 3.6 shows the likelihood function for target strengths of 5 through 45 in steps of 5, as well as the target-absent case.
The Log-likelihood for various assumed mean target strengths

We now consider the behavior of the measurement log-likelihood for various assumed mean target strengths $\hat{T}$. By definition, the log-likelihood for a target assumed to have the strength $\hat{T}$ is given by

$$\ln L(I | \hat{T}) = \ln \left( \frac{L(I | \hat{T})}{L(I | 0)} \right),$$

where the likelihood functions, target-present and target-absent, which comprise the numerator and denominator of the above expression were discussed above. When we examine the behavior of this function for the range of assumed target values from 5 to 50 and for an assumed global mean clutter value of 10 as before, we observe an interesting behavior which might not appear intuitively appealing; the log-likelihood is exhibited in Figure 3.7.
The log-likelihood functions all start around minus 3, pass through zero for intensity values a few times the noise level and reach a maximum lying between 3.5 and 3.75 at an intensity value between approximately one and two times the assumed target strength. Then the likelihood descends asymptotically towards zero, the descent fastest for the smaller assumed target strength. Thus, for example, for an assumed target level of 5 the log-likelihood is very little different from zero for any intensity observation in excess of about 75. The explanation of this behavior is really quite simple. Consider the distribution function for target absent. This is of course given by

\[ P(I \mid 0) = \int_0^\infty dx \mathcal{L}(x \mid 0) = \int_0^\infty dt \frac{\exp(-t^{1/\alpha})}{\Gamma(\alpha + 1)} \exp\left(-\frac{I}{1 + bt^{1/\alpha}}\right). \]

This intensity distribution function is plotted in Figure 3.8. As we see from the graph, the intensity-to-noise ratio is less than 3 approximately 90 percent of the time. It is less than 10 approximately 97.5 percent of the time. On the other hand, in the 2.5 percent or so of cases where the intensity-to-noise ratio is greater than 10, the distribution falls off relatively slowly: It is actually one-percent probable that the intensity will exceed 100, for example. These few-percent probable large values are associated with the mean clutter’s assuming large values out in the tail of its gamma distribution.
Now the explanation of the measurement log-likelihood’s peaking and then falling off asymptotically to zero is clear: For moderately large values (say 10 to 50) of the intensity the presence of a target is the most credible explanation of the data. On the other hand, when the intensity is very large (say greater than 100), the target alone cannot provide a convincing explanation of the observation. The clutter must probably have played an important role – assuming a much larger value than the usual less-than-ten condition – in order that the intensity could be that large. But if greatly aggravated clutter is required to provide a credible explanation of the large intensity value then presence or absence of a target really changes very little. The target might or might not be present: It is not easy to ascertain when the clutter is so large. The falloff towards zero of the measurement log-likelihood reflects precisely this point: When the intensity is very large, the clutter is probably also quite aggravated; and under such conditions the measurement is simply not very informative – hence the log-likelihood tends towards zero.

**Figure 3.8: Target-absent Distribution Function**

![Figure 3.8: Target-absent Distribution Function](image)

**Expected log-likelihood contribution with target present**

Finally we ask the question which determines average detection performance: What is the mean contribution each update of the log-likelihood under the circumstances described in
this case? More precisely, we ask: What is the mean log-likelihood contribution when a target of strength $T$ is present and one assumes a target strength of $\hat{T}$?

Clearly, the expected value of log-likelihood is given by

$$E[\ln L(I \mid \hat{T}) \mid T] = \int_0^\infty dI \, L(I \mid T) \ln L(I \mid \hat{T}).$$

In order to evaluate this integral approximately we define a $300 \times M$ array $L_{I\tau} = L(I, T(\tau))$ with the index $I$ which corresponds to measured intensity running from 1 to 300 and the index $\tau$ corresponding to true target strength running from 1 to $M$. In practice we have set $M = 10$ and defined $T(\tau) = 5\tau$. Similarly, we have defined a similarly dimensioned log-likelihood array $1_{I\tau} = \ln L(I \mid \hat{T}(\tau))$ with $\hat{T}(\tau) = 5\tau$. The approximate evaluation of the above integral (noting that the integrand falls off significantly and can be neglected for $I > 300$) is therefore obtained as the finite sum:

$$E[\ln L(I \mid \hat{T}(\hat{\tau})) \mid T(\tau)] = F_{\hat{T}\hat{\tau}} = L^T I \Delta I = L^T I.$$

The results of this evaluation of expected performance in the present regime is given in the figure below. There we show the expected log-likelihood accumulation per update as a function of true target strength. Plotted in Figure 3.9 are the expected accumulations when the assumed target strength is set to 10, 20, 30, 40, and 50. (At very small true target strengths the curves are ordered with the lowest assumed strength at the top and the largest at the bottom. Obviously they cross to the reverse order as one considers large true values of target strength.)

Although selecting an assumed target strength larger than the target actually present loses some performance, it appears from examining the above set of graphs that a choice of $\hat{T} = 10$ or perhaps 15 achieves a reasonable compromise when the target strength is unknown: An assumption in this range loses only slightly in performance for weaker targets while still maintaining sufficient performance on targets in the 20-50 range to achieve probable detections.\(^1\)

---

\(^1\) This assumes that roughly an accumulation of 25 is necessary to enforce the required false alarm rate and that the number of independent updates available is somewhat larger than 10.
3.4.2 Case 2: No Mean Clutter Estimate With Thresholded Data

In the present section we detail the results for the Case 2 presented earlier, a description of which is the following: No knowledge of the local mean clutter is available, but distribution for the mean clutter is known exactly. The target strength may or may not be known, both situations being examined. Given this degree of knowledge, the radar intensities $I(x,t)$ observed at discrete times at each possible target location $x$ within a time interval during which the target could not have moved outside of the given range-azimuth radar resolution cell are first compared against a fixed threshold. The data on which a declaration decision must be based consist exclusively of the reports at each $x$ of $M(x)$ threshold exceedances out of the $N$ independent opportunities.

When the intensity data are subjected to a threshold test, $I > I_\theta$?, then the number of exceedances, $M$, out of $N$ attempts is distributed according to a binomial distribution. We have

$$L(M | \hat{T}) = \frac{N!}{M!(N-M)!} p^M(I_\theta | \hat{T})(1 - p(I_\theta | \hat{T}))^{N-M}$$
where \( p(I_\theta | \hat{T}) \) is the single-attempt probability of exceeding the threshold for an assumed target of strength \( \hat{T} \). This quantity is given by
\[
P(I_\theta | \hat{T}) = \frac{1}{\Gamma(a+1)} \int_0^\infty dx \exp(-x^\alpha) \exp\left(-\frac{I_\theta}{b x^\alpha + \hat{T}}\right).
\]

The probability which applies under the null hypothesis of no target present in the cell is therefore simply given by setting the target strength parameter to zero:
\[
P(I_\theta | \emptyset) = P(I_\theta | 0) = \frac{1}{\Gamma(a+1)} \int_0^\infty dx \exp(-x^\alpha) \exp\left(-\frac{I_\theta}{1 + bx^\alpha}\right).
\]

According to this general model the measurement log-likelihood ratio for Case 2 is given by
\[
\ln \mathcal{L}(M | \hat{T}) = M \ln \left(\frac{P(I_\theta | \hat{T})}{P(I_\theta | 0)}\right) + (N - M) \ln \left(\frac{1 - P(I_\theta | \hat{T})}{1 - P(I_\theta | 0)}\right)
\]

where \( \hat{T} \) is the assumed target strength.

The average measurement log-likelihood buildup over all the updates for a true target strength of \( T \) is found by replacing the number of exceedances \( M \) in the log-likelihood function given above with its expected value conditioned on target presence. The result of this replacement is to yield an average log-likelihood accumulation
\[
E[\ln \mathcal{L}(M | \hat{T}) | T] = P(I_\theta | T) \ln \left(\frac{P(I_\theta | \hat{T})}{P(I_\theta | 0)}\right) + (1 - P(I_\theta | T)) \ln \left(\frac{1 - P(I_\theta | \hat{T})}{1 - P(I_\theta | 0)}\right).
\]

When the true target strength is known, one sets \( \hat{T} = T \). Clearly, this performance depends upon the choice of a threshold. When the threshold selected is either too small or too large the outcome ceases to be informative and the expected accumulation per update falls off towards zero. We next discuss the selection of a threshold value.

**The threshold setting \( I_q \)**

In order to examine the behavior of mean log-likelihood accumulation as a function of the threshold setting we temporarily assume that the log-likelihood function is matched perfectly to the actual target strength; that is, the true target strength is either assumed to be known or else fortuitously agrees with the value assumed in forming the log-likelihood.
In Figure 3.10 we have plotted the average log-likelihood accumulation per update as a function of threshold setting for ten cases of true target strengths, ranging in steps of 5, from 5 to 50. This figure clearly shows that the accumulation peaks at an optimal choice of threshold which grows somewhat with true target strength, that the dependence of this optimal threshold upon target strength is not very strong, and that the value attained at the optimal point grows (not unexpectedly) with target strength. A more detailed examination of the figure indicates that in the absence of perfect knowledge of the target strength a choice of $I_o$ roughly equal to 5 serves overall quite well, losing performance only slightly with respect to the optimal values matched to the respective target strengths and even permitting the weakest targets to be processed with moderate loss.

**Figure 3.10: Average Log-likelihood vs Threshold**

---

**Expected log-likelihood accumulation per update**

Based upon the results of the last section we shall assume that a threshold given by $I_o = 5$ has been selected. (Of course if true target strength were known we would select a slightly different threshold to optimize results for the case encountered.) We still must make an assumption about target strength in order to translate the resulting number of threshold
exceedances into a log-likelihood. The mean accumulation of the resulting test statistic for a
given actual target strength is shown in Figure 3.11 for a number of choices of the assumed
target strength, again ranging from 5 to 50. The ordering of the curves is that the lowest values
of assumed target strength achieve the best results for very small true target strengths, and the
reverse ordering occurs at the largest values of true strength.

Examination of the figure reveals that the expected accumulations are not extremely
sensitive to the selection of an assumed target strength. Furthermore, as was the outcome for
the Case I analysis, the selection of an assumed target strength somewhere in a range from 10
to 15 appears to represent a good compromise, achieving performance with moderate losses all
up and down the range of true target strength. More precisely, if a target strength in the upper
part of the range 15-50 is anticipated then selecting a value closer to 15 is the indicated
preference, whereas if targets less than 15 are more likely to be encountered, then adoption of
the lower value of 10 is a better choice.

Whether we read the expected performance off the graph shown in Figure 3.11 or else
assume perfect knowledge of the target strength and read the expected performance as the
maximum value achieved as a function of threshold as shown in Figure 3.10, the results are
very similar.

**Figure 3.11: Expected Log-Likelihood Contributions per Update**
Performance attained by non-adaptive threshold method

The performance attained in Case 2 is shown directly by the foregoing graph for the fixed threshold of 5; as we have said, for the case where the target strength is known a slight improvement in performance can be achieved by adjusting the threshold somewhat and by reading the resulting performance as the peak value in Figure 3.10. We shall make very little of this distinction, however, inasmuch as it is seen in Figure 3.10 to amount to a difference which is generally less than a tenth of a log-likelihood unit.

Selecting the assumed target strength to lie in the 10-15 range as recommended above results in an overall performance versus target strength given by the “second” curve (counting from the top at the y-axis) in the figure. Observing in particular the performance for a true strength of 10, we find that the mean accumulation per update for this case is approximately 1.35; and comparing this with the results achieved for Case 1 for this same target strength we see that the mean rate of accumulation for the present case is approximately two-thirds of that achieved for Case 1. Alternatively, to achieve the same performance in Case 2 as that achieved for a target of strength 10 in Case 1 (around 2 log-likelihood units per update) would require a target having a strength of about 25. This corresponds to a loss in sensitivity of 4 dB. We may therefore characterize the counting of fixed threshold exceedances as being 4 dB worse in detection sensitivity, achieving only 2/3 of the likelihood growth, and discarding this degree of information formerly present in the data, than the retention of intensity information.

3.4.3 Case 3: Perfectly Known Mean Clutter

In the present section we detail the results for the Case 3 presented earlier, a description of which is the following: Accurate knowledge of the local mean clutter is provided by means we need not specify here. This local mean, which varies from observation to observation, is drawn from a global gamma distribution. It along with the measured radar intensity comprises the information used in the processing and decision as to whether or not a target is present. The target strength may or may not be known, both situations being examined. Given this degree of knowledge, the measurements at each possible target location x consist of the set of N independent intensities I(x,t) observed at discrete times within a time interval during which the target could not have moved outside of the given range-azimuth radar resolution cell as well as the local mean clutter values C(x,t).
The measurement log-likelihood ratio for Case 3, conditioned on a local clutter value $C$ and an assumed target strength $\hat{T}$ is given by

$$\ln \mathcal{L}(I \mid \hat{T}, C) = \frac{\hat{T}}{(1 + C)(1 + C + \hat{T})} - \ln \left(\frac{1 + C + \hat{T}}{1 + C}\right).$$

We see that the log-likelihood for this case is simply a linear function of observed intensity.

The average measurement log-likelihood buildup each update for a true target strength of $T$ is given by

$$\mathbb{E}[\ln \mathcal{L}(I \mid \hat{T}, C) \mid T] = \frac{1}{\Gamma(a + 1)} \int_0^{\infty} dx \exp(-x^{\frac{1}{a}}) \left(\frac{\hat{T}(1 + bx^{\frac{1}{a}} + T)}{(1 + bx^{\frac{1}{a}})(1 + bx^{\frac{1}{a}} + \hat{T})} - \ln \left(\frac{1 + bx^{\frac{1}{a}} + \hat{T}}{1 + bx^{\frac{1}{a}}}\right)\right),$$

where we have performed the same change of integration variable (from $C$ to $x$) as in the last two sections treating Cases 1 and 2. When the true target strength is known, one sets $\hat{T} = T$; otherwise, the test statistic is the log-likelihood for the assumed target strength instead of for the true target strength.

**Expected log-likelihood accumulation per update**

We have evaluated the expected log-likelihood accumulation per update for Case 3 and present the results in Figure 3.12. In that figure we plot the mean log-likelihood per update as a function of true target strength and show two curves: One curve, the one which crosses the y-axis higher than the other, corresponds to an assumed target strength of 10; the other curve corresponds to an assumed target strength of 50. Clearly, there is very little sensitivity over this range of the accumulation rate upon the assumed target strength. Furthermore, the rate is easily seen from the equations to be exactly linear with true target strength.

Setting the assumed target strength to 10 as we have found to be favorable for the earlier cases considered, we find that for a target of strength 10 the rate of log-likelihood buildup is 7.11 per update. This corresponds to a rate which is just over 3.5 times faster than for Case 1 or 4.7 times faster than for Case 2. Expressed alternatively, in order to achieve the same performance as would be achieved for Case 1 on a 10 dB target, it is only necessary that the target have a strength of about 4 dB using Case 3 processing – a 6 dB advantage with respect to Case 1, or a 10 dB advantage with respect to Case 2.
3.4.4 Case 4: Data Normalized by Mean Clutter

In this section we treat a somewhat different situation than before, the Case 4 defined in Section 3.5.1. In Case 4 we assume as in Case 3 that the local clutter $C(x,t)$ is known perfectly. The radar intensity measurements at each possible target location $x$, denoted earlier by $I(x,t)$, observed at discrete times within a time interval during which the target could not have moved outside of the given range-azimuth radar resolution cell\textsuperscript{2} as well as the local mean clutter values $C(x,t)$ are used to produce a new normalized intensity field which is normalized by the local clutter-plus-noise, $J(x,t) = I(x,t) / (1 + C(x,t))$. Following this normalization the local clutter information is discarded and the detection decision must be based purely on the set of normalized intensities. The target strength may or may not be known, both situations being examined as before.

\textsuperscript{2} This simplification is actually one of description only: For time durations sufficient that the target could move from resolution cell to resolution cell, we can accumulate log-likelihoods along hypothetical target tracks and accomplish the same accumulations as though the target did not move.
Consider the result of normalizing the intensity by a perfect estimate of the local clutter-plus-noise: We write

\[
J = \frac{I}{1 + C}.
\]

The joint density that the local clutter value be given by \( C \) and that the normalized intensity be given by \( J \), conditioned on target level, is equal to

\[
\rho(J, C \mid T) = \rho(C) L(J \mid C, T) = \rho(C) \frac{1 + C}{1 + C + T} \exp\left(\frac{-(1 + C)J}{1 + C + T}\right).
\]

The likelihood for Case 4, conditioned on an assumed target strength \( \hat{T} \), is given by

\[
L(J \mid \hat{T}) = \frac{1}{\Gamma(a+1)} \int_0^\infty dx \left(1 - \frac{\hat{T}}{1 + bx^{1/a} + \hat{T}}\right) \exp(-x^{1/a}) \exp\left(-\left(1 - \frac{\hat{T}}{1 + bx^{1/a} + \hat{T}}\right)J\right),
\]

where we have performed the same change of integration variable (from \( C \) to \( x \)) as in the analysis of Cases 1, 2, and 3.

Observing that the likelihood function under the null hypothesis reduces to \( \exp(-J) \), the measurement log-likelihood ratio for this case is given by

\[
\ln \mathcal{L}(J \mid \hat{T}) = J + \ln \left(\frac{1}{\Gamma(a+1)} \int_0^\infty dx \left(1 - \frac{\hat{T}}{1 + bx^{1/a} + \hat{T}}\right) \exp(-x^{1/a}) \exp\left(-\left(1 - \frac{\hat{T}}{1 + bx^{1/a} + \hat{T}}\right)J\right)\right).
\]

(3.1)

The average measurement log-likelihood buildup each update for a true target strength of \( T \) is given by

\[
\mathbb{E}[\ln \mathcal{L}(J \mid \hat{T}) \mid T] = \int_0^\infty dJ \mathcal{L}(J \mid T) \ln \mathcal{L}(J \mid \hat{T}).
\]

When the true target strength is known, one sets \( \hat{T} = T \); otherwise, the test statistic is the log-likelihood for the assumed target strength instead of for the true target strength.

**Likelihood functions**

We have evaluated the likelihood function of the normalized intensity for several different assumed values of target strength and present the results in Figure 3.13. The lowest
curve which stands out from the rest is the target-absent likelihood function. The other five curves, reading from the bottom to the top refer to assumed target values of 10, 20, 30, 40, and 50. They appear closely clustered on the graph, but of course this is only in relation to the target-absent likelihood.

Figure 3.13: Likelihood Function for Case 4
Measurement log-likelihood function

The measurement log-likelihood ratio function has been computed as a function of normalized intensity for a range of assumed target strengths. This function shows a very weak dependence upon assumed target strength, and we only show two cases in Figure 3.14: The upper curve at large normalized intensities corresponds to an assumed target strength of 50; the lower-slope curve corresponds to an assumed target strength of 10. As we can see from the figure, the leading contribution to the measurement log-likelihood versus normalized intensity, particularly for the smaller of the two example assumed target levels, is the first term of equation 3.1: the normalized intensity $J$ itself. From the graph, the resulting log-likelihood function is not easily distinguishable from a linear function of normalized intensity.

**Figure 3.14: Log-Likelihood Function**

![Graph showing log-likelihood function with normalized intensity on the x-axis and log-likelihood on the y-axis. Two curves, one for each assumed target strength (10 and 50), are shown.]

Expected log-likelihood accumulation per update

Finally, we calculate the expected log-likelihood accumulation per update as a function of true target strength. As in Case 3, the results are so insensitive to assumed target strength that we show only two cases in Figure 3.15: The curve of greater slope corresponds to an assumed target strength of 50; the curve of slightly lesser slope corresponds to an assumed
strength of 10. Again, the relationship between expected log-likelihood accumulation per update and actual target strength is quite linear, as was found for Case 3. Particularly surprising, however, is the fact that the predicted rate of accumulation is virtually the same here as for Case 3: We find a predicted value of 7.17 at a target strength of 10, for example; this is to be compared with the value 7.11 found for the same strength in Case 3. We recognize that the Case 4 results cannot possibly be better than those of Case 3 (because all the information from Case 4 is available to Case 3 in addition to that of the absolute level of local clutter), and the small difference actually measured is well within the presumed accuracy of our numerical estimates. Nonetheless, the fact that these two cases are so similar is not presently explained.

**Figure 3.15: Mean Log-Likelihood Accumulation Per Update**

![Graph showing mean log-likelihood accumulation per update for different target strengths.](image)

**True Target Strength**

**Concluding Remark for Case 4**

We have seen two facts:

1. The performance of the system using normalized intensity data, well-informed during normalization as to the correct local mean clutter-plus-noise, is essentially equivalent to that of the system which retains the intensity as well as the exact local mean clutter values.

2. The measurement log-likelihood function is well represented for this case as a linear function of normalized intensity.
Now if we posit one additional ingredient, namely that the local clutter-plus-noise might be well estimated by a suitable\(^3\) space-time average about the cell in question, then one would obtain a surprising result: *Under the conditions seen to hold and the additional ingredient conjectured we would conclude that simply normalizing the data by a suitable moving space-time average would result in an almost exactly optimal test statistic – linearly related to the log-likelihood.*

### 3.4.5 Case 5: Data Normalized by Mean Clutter and Thresholded

We treat in this section a situation which bears some of the same elements of Cases 2, 3, and 4: Like Cases 3 and 4, the true local clutter values are assumed known perfectly; like Case 4 the local clutter plus noise is only used to normalize the intensity and then discarded; and finally, like Case 2, the actual continuous data (obtained in the present instance after normalization) is thresholded and only the number of exceedances is available for making the target declaration decision. The true target strength may or may not be known, both situations being examined as before.

First we write

\[
L(M | \hat{T}) = \frac{N!}{M!(N - M)!} p^M(J_o | \hat{T})(1 - p(J_o | \hat{T}))^{N - M},
\]

where \(p(J_o | \hat{T})\) is the single-attempt probability of exceeding the threshold for an assumed target of strength \(\hat{T}\). This quantity is now given by

\[
P(J_o | \hat{T}) = \frac{1}{\Gamma(a+1)} \int_0^\infty dx \exp(-x^{1/a}) \exp \left( - \left( \frac{\hat{T}}{1 + bx^{1/a} + \hat{T}} \right)^{1/a} J_o \right),
\]

where we have performed the same change of integration variable (from \(C\) to \(x\)) as in the analysis of Cases 1, 2, 3, and 4. When the assumed target strength is set to zero we obtain the probability of exceedance under the null hypothesis of target absent. This reduces to the simple form

\[
P(J_o | \emptyset) = P(J_o | 0) = \exp(-J_o).
\]

Given these values for the probabilities, the log-likelihood function after observing \(M\) exceedances from \(N\) independent trials is given by

---

\(^3\) In this case “suitable” might require omitting a neighborhood about the cell in question in order to eliminate “target capture” effects.
\[
\ln \mathcal{L}(M \mid \hat{T}) = M \ln \frac{P(J_{\phi} \mid \hat{T})}{P(J_{\phi} \mid 0)} + (N - M) \ln \frac{1 - P(J_{\phi} \mid \hat{T})}{1 - P(J_{\phi} \mid 0)}.
\]

The average measurement log-likelihood buildup each update for a true target strength of \( T \) is therefore

\[
E[\ln \mathcal{L}(M \mid \hat{T}) \mid T] = P(J_{\phi} \mid T) \ln \frac{P(J_{\phi} \mid \hat{T})}{P(J_{\phi} \mid 0)} + (1 - P(J_{\phi} \mid T)) \ln \frac{1 - P(J_{\phi} \mid \hat{T})}{1 - P(J_{\phi} \mid 0)}.
\]

When the true target strength is known, one sets \( \hat{T} = T \); otherwise, the test statistic is the log-likelihood for the assumed target strength instead of for the true target strength.

**The threshold setting \( J_q \)**

In order to examine the behavior of mean log-likelihood accumulation as a function of the threshold setting we temporarily assume that the log-likelihood function is matched perfectly to the actual target strength; that is, the true target strength is either assumed to be known or else fortuitously agrees with the value assumed in forming the log-likelihood.

In Figure 3.16 we have plotted the average log-likelihood accumulation per update as a function of threshold setting for ten cases of true target strengths, ranging in steps of 5, from 5 to 50. This figure clearly shows that the accumulation peaks at an optimal choice of threshold which is close to the true target strength and that the value attained at the optimal point grows (not unexpectedly) with target strength.

We can obtain additional insight into the behavior of this threshold approach by calculating the probability per opportunity of exceeding the optimally set threshold when a target of the assumed strength is actually present. Figure 3.17 shows this target-present threshold exceedance probability as a function of target strength. We observe that over the range from 5 to 50 the optimal setting results in a very narrow range of probabilities between 0.35 and 0.41. We shall have further occasion to mention this fact in the discussion of the overall performance of the method.
Figure 3.16: Threshold Dependence of Average Log-Likelihood

Figure 3.17: Optimal Threshold Passage Probability Settings
Expected log-likelihood accumulation per update

Based upon the results above we shall assume that a threshold given by $J_\theta = \hat{T}$ has been selected. The mean accumulation of the resulting test statistic for a given actual target strength is shown in Figure 3.18 for a number of choices of the assumed target strength, again ranging from 5 to 50. The ordering of the curves is that the lowest values of assumed target strength achieve the best results for very small true target strengths, and the reverse ordering occurs at the largest values of true strength.

Examination of the figure shows that performance is very much dependent on the target strength assumed: If a target strength of 10 is assumed, for example, the threshold setting which goes with that choice optimizes the log-likelihood accumulation for a target of strength 10 but gives up the potential much larger performance on targets much greater than 10. The reason is clear from the behavior seen in Figure 3.17. If a threshold setting for a target strength of 10 gives a probability of about 0.38 that such a target would produce a threshold exceedance at each opportunity then no target no matter how strong could produce more than about two and one-half times that number on average; this bounds the likelihood accumulation for an arbitrarily large target to roughly 2.5 times that of a target of strength 10 when that threshold setting has been selected. On the other hand, if one selects a threshold setting appropriate to a much larger target strength then each such threshold exceedance is worth more in log-likelihood units – allowing the larger accumulations for larger targets – but the probability of achieving exceedances is so reduced for the smaller targets that they no longer enjoy large enough mean numbers of exceedances effectively to build log-likelihood on average.
**Performance attained by thresholding the normalized intensity**

The performance attained in Case 5 as shown directly by the foregoing graph for various assumed target strengths is, as we have said, highly dependent on this choice. Fixing attention on the target strength given most attention for comparative purposes, a strength of 10, results in the performance versus target strength given by the “second” curve (counting from the top at the y-axis) in the figure. Observing in particular the performance for a true strength of 10, we find that the mean accumulation per update for this case is approximately 3.14; and comparing this with the results achieved for Cases 3 and 4 for this same target strength, we see that the mean rate of accumulation for the present case is approximately 44 percent of that achieved for Cases 3 and 4. Alternatively, to achieve the same performance in Case 5 as that achieved for a target of strength 10 in Cases 3 and 4 (around 7.2 log-likelihood units per update) would require a target matched to the strength assumption having a strength of about 22. This corresponds to a loss in sensitivity of 3.4 dB. We may therefore characterize the counting of threshold exceedances of perfectly normalized intensity data as being 3.4 dB worse in detection sensitivity at a target level of 10, achieving only 44 percent of the average log-likelihood growth there, and discarding this degree of information formerly present in the data. Perhaps of equal importance, however, is the fact already noted that the thresholding operation
abandons any opportunity for achieving greatly improved performance for much larger targets than those assumed in the detection process.

3.4.6 Conclusions

This section summarizes the results of five cases described in Section 3.5.1 which discussed a method for bounding the performance of different processing streams in the problem of radar detection of periscopes. The overall method described there was to compare the average target-present accumulations of log-likelihood associated with each data employed in each given processing schema. Two follow-on sections dealt with the numerical results for two cases where the local mean clutter was completely unknown. The first of these dealt with the situation where the radar intensity comprised the information to be processed; the second, with that where one thresholded each intensity value and only retained for each cell information about how many threshold exceedances occurred out of the \( N \) independent opportunities. A third follow-on section treated the results for the Case 3 described earlier. In that situation accurate knowledge of the local mean clutter is provided by means we need not specify. This local mean, which varies from observation to observation, is drawn from a global gamma distribution. It along with the measured radar intensity comprised the information used in the processing and decision as to whether or not a target was present. In the section immediately preceding this one we treated a somewhat different situation, the Case 4, where we assumed as in Case 3 that the local clutter \( C(x,t) \) is known perfectly. The radar intensity measurements at each possible target location \( x \), denoted earlier by \( I(x,t) \), observed at discrete times within a time interval during which the target could not have moved outside of the given range-azimuth radar resolution cell as well as the local mean clutter values \( C(x,t) \) were used to produce a new normalized intensity field which was normalized by the local clutter-plus-noise, \( J(x,t) = I(x,t) / (1 + C(x,t)) \). Following this normalization the local clutter information was discarded and the detection decision was based in Case 4 purely on the set of normalized intensities. Finally, in Case 5 the true local clutter values were assumed known perfectly as in Case 3; like Case 4 the local clutter plus noise was only used to normalize the intensity and then discarded; and like Case 2 the actual continuous data (obtained in the Case 5 instance after normalization) was thresholded and only the number of exceedances was available for making the target declaration decision. The true target strength may or may not be known in any of these treatments, both situations being examined.
Performance results

The following Figure 3.19 shows the average log-likelihood accumulation per update versus true target strength for all five cases. For comparison, we have chosen an assumed target strength of 10; however only Case 5 is particularly sensitive to this choice, as discussed in the previous section. We shall discuss the various numerical and qualitative comparisons among the various cases in the next section, however in a certain sense the curves in Figure 3.19 speak for themselves.

![Figure 3.19: Comparison of Performance for Five Cases](image)

We have seen that the Cases 3 and 4, which make different uses of the local clutter information perform best of all: Each achieves a growth rate of about 7.15 log-likelihood units per update on a target of strength 10. Next, the Case 5 also uses the local clutter estimate but as a result of discarding information only achieves an average growth of 3.14. Trailing behind, Case 1 achieves a rate of 2 log-likelihood units per update on a 10 strength target, and Case 2 trails all others with an average rate of about 1.35. These comparative rates are illustrated graphically below in Figure 3.20.
One may view the consequences of different log-likelihood accumulation rates in various ways. One such way is illustrated in Figure 3.21. Based upon an assumed requirement to accumulate approximately 22 log-likelihood units in order to enforce a false-alarm rate of about 1 per hour, we can determine and compare the number of independent observations on the target track required in each case in order to arrive at a detection probability (for a 10 strength target) of around 50 percent at this false alarm rate. The Cases 3 and 4 achieve this detection probability in about 3 independent scans, Case 5 requires around 7, Case 1 would need 11, and Case 2 would require about 16 independent scans of data to achieve the same required log-likelihood build-up needed for detection.

These comparisons are illustrated in Figure 3.21.
Another comparison is provided by Figure 3.22. We have already pointed out that Cases 3 and 4 achieve detectable performance at reasonable false alarm rates after only three independent observations on average. (This is also the minimum number of observations required to establish an estimate of target velocity and kinematic scatter about a uniform motion trajectory.) One might ask: How many false alarms would be produced by the other cases if they were also set to provide a 50 percent probability of detecting the target (as part of that set)? We approximate the dependence of false alarm probability upon log-likelihood accumulation via an exponential dependence. The results, as shown in Figure 3.22, are that while Cases 3 and 4 are producing approximately 1 false alarm per hour, Case 5 is producing around 50 per second, Case 1 is producing about 1400 candidates per second, and Case 2 would result in an average of 10,000 false candidate detections per second at the same credibility level as the target.
The general conclusions drawn from these numerical comparisons and from the behavior seen over the course of the analysis are briefly summarized in the following section.

**General overall conclusions**

Having made these numerical comparisons we summarize with a set of qualitative conclusions drawn from the study. These are not presented in any special order.

- The performance advantages of processing the radar data with good information about the underlying local clutter are clear and significant: At 10 dB SNR levels growth rates of log-likelihood are typically around 4 times what could be achieved without that knowledge.

- Cases 3 and 4 produce performances which cannot be distinguished at the level of accuracy achieved in this analysis. This equivalence of result has not yet been shown to arise from any analytical equivalence.
• Only the Case 3 and Case 4 processing provides good performance over a wide range of actual target strengths without actually knowing the true target strength. These cases afford growth rates which are essentially proportional to the true target strength, whereas the other data types exhibit strong saturation of growth rate for targets which are larger than the assumed strength.

• At the small spikiness parameter value studied the dependence upon actual mean clutter level is quite small: The signal-to-noise is much more an important parameter than the mean clutter to noise or the target-to-clutter ratio. The situation is approximately described in these terms: Most of the time the clutter is negligible; however in that 5 percent or so of the time when it is larger than the system noise it is potentially so large as to be able to explain any large measurement, absent an actual estimate of the local clutter. Those instances of data do not generally convey very much information about target presence or absence even when local clutter estimates are available. The principal leverage achieved by local clutter estimates arises from knowing when the clutter does not fall into the unusual class, thereby allowing full exploitation of the information available during those situations.

• When one considers the consequences of thresholding, both in Case 2 and Case 5, one finds that unacceptable losses attend this reckless discarding of information. Roughly speaking, the abandonment of actual values observed in favor of counting exceedances discards about half of the information content – half of the log-likelihood accumulation rate.

• Although not involving any knowledge of the local clutter the Case 1 performance is quite close to being adequate for periscope detection. Expectation of having 11 independent hits on a target which exhibits a 10 second exposure, requiring roughly 1 second decorrelation of the clutter, does not seem to be unrealistic.

• To achieve the performance just described for Case 1 involves the use of a highly non-linear and unusual log-likelihood function which is sensitive to assumed target strength and which actually peaks at moderate intensity values. Furthermore, it requires some knowledge of the underlying clutter statistics.

• The comparisons arrived at over the sequence of section and described here are valid even when other features are available for false alarm assessment provided one can approximate those features as statistically independent of the intensity data. That is, if
additional data provides additive log-likelihood – spatial scatter about a hypothetical track, for example – then these contributions studied here and their relative sizes are still pertinent for determination of comparative performance – for example for determining the numbers of candidates passed to discrimination by each type of intensity processor.
CHAPTER 4

4. UCTP SOFTWARE

In order to demonstrate the UCTP theory outlined in Chapters 1 and 2 a number of software modules were written at Metron. This chapter describes the software components. Initial development was done using the Nodestar software developed by Metron for the Spotlight project. Likelihood functions for the periscope detection radar were coded into the Nodestar framework. After numerous test runs, we decided to split the UCTP software off from the Nodestar development. Many of the advanced features of Nodestar (world maps, land avoidance, contact association logic, etc.) were not needed for UCTP, so by developing UCTP as a separate program we were able to reduce the code base and make the program easier to develop. With the more compact code we were able to introduce algorithmic and computer science optimization to make the code run faster. This became important for the large scale performance benchmarking described in Chapter 5.

Section 4.1 describes the clutter tracker and Section 4.2 outlines the target likelihood ratio tracker. In Section 4.3 we describe the graphical user interface that was developed for the UCTP project. The purpose of the GUI is to orchestrate runs from setting the parameters, choosing which programs to run, and then viewing the results. A number of smaller stand-alone programs were developed for the data analysis. Section 4.4 describes these programs.

4.1 Clutter Tracker

The clutter tracker software computes the covariance of log-intensity in a clutter field, computes the optimal filter coefficients, and performs the two-dimensional finite impulse
response (FIR) filter using those coefficients. The first two steps are accomplished by a executable called hmult_SUN, while the last step is performed by another executable called ct_SUN. Those software modules are described in this section.

4.1.1 Inputs

The inputs to both hmult_SUN and ct_SUN are the same. These are passed in through a configuration file. There are 14 parameters. The first four are file names. CLUTTER is the file used for estimating the covariances and designing the filter coefficients. TARGET is the file that will be used for performing the FIR filter. TARGET and CLUTTER may be the same file. ESTIMATE is the file where the clutter estimate will be written. HMULTIPLIERS is the file used to store the filter coefficients in binary form; hmult_SUN writes to this file and ct_SUN reads from this file. The next three parameters refer to the size of the data files. N RANGE1 is the number of range bins in the CLUTTER file, N RANGE2 is the number of range bins in the TARGET file, and NTIMES is the number of time steps in both files. The next five parameters define the size of the filter mask. They are denoted M, N, A, and B. These parameters follow the notation in Section 2.3.2. The next parameter is TCinM. If this parameter is set to zero, the target cell is not included in the filter mask; if it is non-zero, the cell is included. This flag is needed by hmult_SUN. The final two parameters, LOGSPECKLE_MEAN and LOGSPECKLE_VAR, are the mean and the variance of the speckle. LOGSPECKLE_MEAN is used by ct_SUN when converting the filtered conditioned data back into the clutter mean intensity. LOGSPECKLE_VAR is used in hmult_SUN to estimate the residual variance, and also in the filter weight computation if the target cell is included.

4.1.2 Class Structure

There are five C++ classes in the clutter tracker code. The first one is a configuration class common to hmult_SUN and ct_SUN called Config. Its private data contains the 14 parameters described in the previous section. The primary method is one that parses an input file looking for keywords denoting a parameter followed by the value of the parameter.

The second class, also used by both hmult_SUN and ct_SUN, is TimeBlockSet. It is used to store a two-dimensional array read in from disk. There are two parameters for the dimensions of the array. After reading the array in binary form, the constructor computes the mean of the data, computes and stores the logarithm of the data in an array, and also computes the mean of the log-data. Before storing the first array it truncates all data below a certain threshold.
The third class, called VarCov, is used by hmult_SUN. It operates on a TimeBlockSet object to compute the covariance as a function of lag in each of the two dimensions. It contains parameters A, B, M, and N. The covariances are computed for time lags between 0 and (M+N) inclusive, and time lags of -(A+B) to (A+B) inclusive. If any of the parameters is negative, the above expressions are modified by using zero for the value of that parameter. An operator is defined in the class to allow access to the covariances with a double subscript, even though the array is stored linearly.

The next class is HMultipliers. The constructor is called with a VarCov object which is stored as the first member variable and also used to compute the other member variables. The other members are the optimal filter weights, the system of equations used in computing the weights, the condition number of that matrix, and error codes from the CLAPACK matrix solver.

The final class is ClutterEstimate used by ct_SUN. Its constructor is called with the name of a file, a TimeBlockSet, and four size parameters. An external Config object is also used to relay information about mask size and speckle mean. A set of filter weights is read from the file and then applied to the data in the TimeBlockSet over a subrectangle defined by the last four input parameters. The second array (log-data) from the TimeBlockSet is used for the filtering. The estimate stored with the ClutterEstimate member variable is the filtered values converted back to clutter intensity (not log-intensity) after adding in the mean value from the TimeBlockSet object and subtracting the speckle mean.

4.1.3 Functions

In addition to the usual type of functions like member access, destructor, output, and two-dimensional array access, there are a few non-trivial functions in the above classes that handle the UCTP clutter processing. Four of those functions are briefly described.

4.1.3.1 Covariance Calculation

The constructor of the VarCov class computes the covariance of the TimeBlockSet passed in as the last parameter. The covariances are computed for positive time lags up to (M+N), and for range lags of -(A+B) to (A+B). Since the covariances are symmetric function of time and range, it is unnecessary to calculate the covariance for negative time lags. For a fixed time and range lag, the covariance is computed by taking all pairs of points separated by those lags over the full TimeBlockSet. For computational efficiency, the points are indexed by
two pointers moving through the arrays. The total number of sample pairs is counted as the sum is accumulated, so an accurate average can be calculated.

4.1.3.2 Filter Coefficient Calculation

The constructor in the HMultipliers class computes the optimal filter coefficients from the covariance data supplied by the input VarCov object. It allocates memory for a system of equations and fills the entries as shown in equation 2.1 from Section 2.3.2. If the target cell is not in the mask, then entries in the system of equations are zeroed out in such a way to force that filter coefficient to be zero without affecting the other weights. The system of equations is solved using the CLAPACK library of numerical linear algebra routines. After reformatting the system in a bridge module, the routine “dgesvx_” is called. This is a general routine that does not make use of any special structure (such as being Toeplitz block-Toeplitz) in the system. The condition number of the system is passed back from dgesvx, and is accessible for debugging in the HMultipliers object.

4.1.3.3 Residual Variance

Immediately after computing the optimal filter weights, the estimator variance is computed. As shown in Section 2.3.2, the residual variance is calculated from the inner product of the optimal weights and the right hand side of the system.

4.1.3.4 Evaluation of Residual Variance for Arbitrary Filter Weights

For general filter weights, the residual variance is a quadratic function of the weights with respect to the covariance matrix. The function compute_residual_var in the HMultipliers class assumes that the VarCov member is correct and that storage has been allocated in the class for the filter weights. The weights are read in from a file passed to the routine. The calculation proceeds as in Section 2.3.2.

4.1.4 Outputs

As the clutter tracker code runs, there are a number of outputs. The first set of outputs goes to the standard output. As the program is running, status messages are written to cout showing the progress of the processing. As the configuration file is read, the parameters are echoed to cout for debugging. In TimeBlockSet, all values below the threshold are written to cout. As the covariances are computed in the constructor of VarCov, the lags, the covariance, and number of data points are written. In the HMultiplier class, messages are
displayed when matrix elements are zeroed out if the target cell is not included in the mask. The residual variance is also written to cout to be used by the likelihood ratio tracker. If the residual variance for general filter coefficients is calculated, then the residual variance is also written to cout. During ct_SUN, a status message is written every 100 scans.

The other set of outputs is to files. The optimal filter coefficients are written to two files, one in binary form and the other in ASCII form. The other output is the clutter estimate written in binary form.

4.2 Likelihood Ratio Tracker

The software for the LRT comes in a set of ten C++ modules. Most of them have corresponding header files. The basic contents are specified in the table below.

<table>
<thead>
<tr>
<th>Module Name</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>main.C</td>
<td>Initialization and main event loop including reading of clutter tracker output and radar scans</td>
</tr>
<tr>
<td>config.C</td>
<td>Describes class containing externally specified parameters. Contains all access for configuration parameters</td>
</tr>
<tr>
<td>radarscan.C</td>
<td>Describes RadarScan, the base class for cell-based processing. Includes new target processing. Includes evaluation processing</td>
</tr>
<tr>
<td>nyquist_radarscan.C</td>
<td>Describes base class for Nyquist processing which is derived from the class RadarScan. Contains the motion model and new target processing</td>
</tr>
<tr>
<td>velstate.C</td>
<td>Describes the velocity sheet class. Includes motion model</td>
</tr>
<tr>
<td>clutter_calc.C</td>
<td>Contains the processing for generating the likelihood ratios and mapping them onto the LRT grid structure</td>
</tr>
<tr>
<td>nr.h</td>
<td>Header file for access to “Numerical Recipes” Used in Nyquist Processing</td>
</tr>
<tr>
<td>nrutil.c</td>
<td>“Numerical Recipes” code used in Nyquist processing</td>
</tr>
</tbody>
</table>
complex.c  "Numerical Recipes" code used in Nyquist processing
four1.c  Fourier transform used in Nyquist processing

Table 4.1

4.2.1 Main Event Loop

The main event loop (in module main.c) is very similar for the Nyquist and cell based options. The initial call to pure_motion performs the motion update to the time of the next radar scan. Function pop_like does the new target introduction adjustment to the likelihood ratio. Comp_likelihood reads in the radar scan and the clutter tracker output, calculates the measurement likelihood and stores the result for each velocity sheet. The call to fuse_likelihood actually incorporates the measurement likelihood into the likelihood ratio. The remaining calls in the loop generate evaluation output.

4.2.2 Measurement Likelihood Ratio

Foremost among the calculation devices in the LRT is the use of the logarithm of the likelihood instead of the likelihood itself. This has several advantages in the processing of new likelihood ratios. One advantage is that the LRT may suffer numerical overflow problems on some machines when the processing is done in standard form. These difficulties vanish when the logarithms are processed. The likelihood ratio is calculated as quotient. The log of each component can be more efficiently calculated and then the difference can be taken to give the log of the ratio. Several exponentiations and a quotient are thereby saved. Since the velocity sheets are independent, there is not a problem with having log likelihood rather than likelihood even for the motion model.

One function that does not work more easily in logarithmic form is the new target introduction calculation. This is done in the function pop_like in module radarscan.c and module nyquist_radarscan.c. Additionally, some of the evaluation items, e.g. marginals, are not readily calculated in logarithmic form.

4.2.2.1 Single / Wide / Nyquist

There are several options for the target model in the LRT. The simpler model is that a target will only appear in one radar cell at a time. This option is more prone to discretization errors as a target shifts between cells. A more accurate model assumes that the target signature will appear in several cells. This latter option requires several parameters to specify how the
target signature is expected to spread and one parameter to specify the fineness of an auxiliary
grid used in the mapping from the continuous likelihood function to the discrete grid.

4.2.2.2 Residual Variance Usage

The residual variance is set for LRT processing through the configuration options,
normally through the configuration file. It is used in the functions targetloglike and
noiseloglike found in clutter_calc. It provides a scaling parameter for discrete
distribution modeling the uncertainty in the estimated clutter.

4.2.2.3 Static / Moving Velocity Sheets

The velocity sheets for the LRT are all carried on the same size grid with the same
number of cells. Each velocity sheet carries a parameter which gives the fractional offset which
has not yet been used to shift values. Traditional Nodestar only uses the fractional offset for the
motion update, but does not account for the fractional offset when it generates likelihood
functions. In the LRT, the problem involves detecting phenomenon which are localized within
a few cells. The fractional shift is potentially significant for this problem. Accounting for the
fractional shifts when generating the likelihood ratio for the velocity sheets is optional. The
LRT can either generate one likelihood ratio for the grids as if the fractional offset were zero or
it can generate a more accurate mapping based on the fractional offsets for each sheet.

4.2.2.4 Radar to Nodestar Cell Conversion

The radar scan and the clutter tracker output are carried on a fixed position grid. The
internal LRT state space covers the same range, but need not have the same number of cells as
the radar grid. In either case the cell or cells in the radar input which correspond to the internal
LRT cells are determined as follows. For the LRT cell, find the center position. Find the radar
scan or clutter tracker output grid cell which contains that center position. For wide target
processing, a range of cells is determined by a similar process. This is found in the processing
options of clutter_calc.C.

4.2.3 Motion Update

Cell-Based. The cell based motion model is accessed through a series of calls. The
highest level is the function pure_motion of class RadarScan. It in turn calls
translate. The latter function calls move of class VelocityState for each velocity
sheet in turn. The hierarchy of calls is inherited from Nodestar where the motion model is more complex since it involves the possibility of transferring between velocity sheets.

The motion itself involves shifting values of the likelihood ratio within a velocity sheet. This is done by first determining the distance to be moved, given the velocity, in units of grid cells. Then adding the previous fractional shift. The closest integer to the resulting shift is determined. The values are shifted by that integer amount and the fractional remainder becomes the new fractional offset. Note that cells which do not have a value shifted into them are reset to the initial value.

**Nyquist.** The motion update for the Nyquist processing option is significantly more complicated and involves Fourier transforms. The motion update is initiated by calling pure_motion which calls MotionUpdate. MotionUpdate, in turn, calls translate which processes the entire state space using Fourier transform methods. Once again, values which are vacated by the motion are filled in with the (logarithm of the) initialization value.

### 4.3 Graphical User Interface

The general software design of the UCTP interface is meant to allow the user to enter and save the parameters for each step of the detection process. The main panel (Figure 4.1) divides the program into the three main sections:

1) Selecting the clutter file and injecting the target(s)

2) Generating the H Coefficients and running the Clutter Tracker

3) Running the LRT
The Setup Batch Run button allows the user to select several directories, start several runs and allow the processes to run overnight (Figure 4.2). The final button, Image Processing, provides a way to display both the initial clutter data files and the result data files.
Selecting a Working Directory. The Directory menu allows the user to either select an existing working directory or to create a new directory. From the directory window, the user can create an empty new directory or he can create a new directory which has copies of the parameter setup files from an existing directory. Figure 4.3 shows the directory selecting window. To create a new directory type the new directory’s name, with an ending ‘/’, into the lower text box and press the New Directory button.

![Set Directory](image)

Figure 4.3

Figure 4.4 shows the New Directory Setup window. When a new directory is created, all the relevant files from the source directory are displayed in their appropriate text boxes. The user is then allowed to change any of the file names. The Apply button copies the relevant files into the new directory.
4.3.1 Selecting the Clutter File and Injecting the Target(s)

After selecting a working directory, the user can either load an existing injection file, usually named inject.inp, or create a new injection file. To load an injection file, select Load Data Gen File from the file menu. The check box to the left of the Data Generation button will now contain a check mark and the name of the loaded file will be on the right. Pressing the Data Generation button will bring up the window in Figure 4.5.

Within this window the user can shorten a raw clutter data file and inject the target(s). All of this information is saved into the inject.inp file. The Inject Target File Setup button calls the configuration wizard. This is an external program that allows you to create or edit all of the relevant setup files like inject.inp, Config.Dat, and CT_Config.Dat.
4.3.2 Generating the H Coefficients and Running the Clutter Tracker

To generate the H coefficients or to run the clutter tracker, press the Clutter Tracker button. Here again, the user can load an existing file or create a new one. The clutter tracker setup window is shown in Figure 4.6.

The Select H-Mult File button allows the user to select any H Coefficient file. The Setup Clutter Tracker Files button calls the configuration wizard to edit or create the CT_Config.Dat configuration file. The Gen Multipliers button generates new H coefficients and the Run CT button actually runs the clutter tracker.
4.3.3 Running the LRT

Pressing the LRT button in the main window brings up the likelihood ratio tracker window (Figure 4.7). Once again the Setup LRT Files button opens the configuration wizard where the user can edit or create the Config.Dat configuration file. The Run LRT button will run the tracker.

4.3.4 Image Processing

The Image Processing button in the main window opens the Image Processing window (Figure 4.8). In this window the user can open several data files, both raw clutter files and
results files, and view them. The scales on the left adjust the color map used to display the data. The controls in the lower right corner concern the data file. The user must enter the size of the data file he wishes to display. The return button must be pressed for changes in the text boxes to take effect. The user can also select to take the log of the data by checking the Log check box. Finally, the zooming controls allow the user to zoom in on the target in the display window.

![Image Processing](image)

**Figure 4.8**

The image files can be loaded and closed from the File menu. Currently, only four image files may be open simultaneously.

The following picture is an example of the raw clutter file (Figure 4.9).
4.4 Utility Programs

A number of utility programs were written for data analysis. This section briefly describes those programs. The programs are for injecting simulated targets into clutter scenes, finding peaks in the likelihood ratio surface caused by clutter patches, manipulating files, finding the peak likelihood ratio produced by an injected target, and extrapolating observed clutter peaks to other false alarm rates.

4.4.1 Target Injection

The target injection code is written in FORTRAN, unlike the clutter tracker and likelihood ratio tracker which are written in C++. The injection code reads an input file containing parameters describing the target. Values in the data file are replaced with target intensities along the target’s trajectory.
The single target injection input file contains parameters specifying the target’s intensity, correlation time, model, scan rate, speed, start time, start range, and exposure time. The file also contains the clutter level, noise level, range resolution of the radar, data file, and random seed. The target intensity, $\text{SCR}$, is given in decibels relative to the clutter level in the input file. The clutter level, $c$, and system noise levels, $n$, are also specified, so that the target strength, $t$, is given by

$$t = \sqrt{10^{\left(\frac{\text{SCR} + c}{10}\right)} \cdot n}$$

The target amplitude along the trajectory is a correlated random draw based on the correlation time in the input file. The target model can be either exponential or a Rice-squared distribution. The Rice-squared distribution parameter has an approximate range of 0.1 to 20, where low values represent a nearly constant target and higher values represent a more widely varying target. The radar scan rate is 300 rpm or 5 Hz. The target speed is given in meters per second, where positive speeds are in the positive range direction. The starting range and time determine the target’s initial position in the radar scene. The radar range resolution determines the width of the target in range, where the general form of the target signature is a sinc squared.

The injection code first generates the correlated random variates to use for the target intensities. Then an index into the data array is determined for the start range and time. The target wings are included out to ten times the radar resolution in range. Finally, for each scan in which the target is present, the complex clutter and target are assigned random phases and added. The resulting value is inserted in the data file, replacing the old clutter value. Statistics on the target intensities, clutter values, and phase assignments are calculated and output to an auxiliary file.

The multiple target injection code uses the single target injection code. New parameters in the input file include the number of targets to inject, number of scans in the data, and the range of velocities. Assuming a 512 bin radar scene, the targets are injected into alternating eighty foot wide bins. The target bins are separated by ten second intervals and have a sixty-six foot range border around the perimeter of the scene so that they do not interfere with one another in the likelihood calculation. The targets are injected sequentially along the velocity range specified in the input file with different random seeds.

### 4.4.2 Local Peaks

The clutter peaks in the data scenes that are processed without targets are found using a local maximum finding utility written in C++. The reason for finding a local maximum over
some neighborhood is to count the likelihood buildup over a clutter patch only once. This code extracts the local maximum value of a cell in a two dimensional state space given parameters to govern the height and width of the local max neighborhood. The maximum value of the observation likelihood over the velocity hypotheses is found within the likelihood ratio tracker, thereby reducing the state space significantly in size and from three to two dimensions: range and time. The final two dimensions are searched using a neighborhood size specified either by standard input or in an input file. Each cell in the state space is first assumed to be a local maximum. If a cell containing a larger value is found within the neighborhood window, then the original cell is no longer considered a maximum. The cells near the edges of the data are only compared to cells within the portion of the window that overlaps the data. The utility outputs the value, range bin and scan number of the local maximum. A second utility file removes clutter peaks near the edges of the data using the list output by the original local maximum utility.

4.4.3 Data File Handling

Several C utilities are used to handle the data files. The first utility converts the data from PC to Sun formats by swapping the byte order. Two other utilities can shorten or append the file. The first utility is used when the data file is larger than necessary: e.g. when only part of the file is needed to encompass the target injection sites, even after the targets are spaced. The second utility is used when the data file is not long enough to hold all of the required target bins after spacing. In that case, the file is appended to itself to generate a long enough scene in time to hold all of the targets for processing. A final utility is used to clip the target levels after injection to the maximum clutter value that appears in the scene.

4.4.4 Target Maximums

Determining the target maximum occurs within the likelihood ratio tracker, but was originally a separate utility. It is included in the LRT so that the state space does not need to be output for each run, and, given limited disk partition size, multiple runs on multiple processors can be executed simultaneously. The algorithm is similar to the local max finding, with the exception that the target parameters are known so the search is confined to a small box in range, time, and velocity around the target track. The target parameters and window size are first read by the LRT. After each scan is processed by the LRT, it is determined whether or not a target is present in the scan given the target parameters. If a target is present for that scan, the range bin of the target’s center is determined again by referencing the list of target parameters. Finally, the velocity sheets, range cells, and scans within the pre-defined search area are
checked to determine the maximum value of the target in the neighborhood of the target cell. The largest value of likelihood accumulated for each target is output to a file.

4.4.5 Peak Extrapolation

The threshold necessary for a given false alarm rate is determined by a small C++ program that fits observed clutter peaks to an exponential form and then analytically extrapolates to the desired false alarm rate. The algorithm was implemented as follows: assume that the largest $M$ of $N$ values are distributed with a shifted exponential distribution. If $a$ is the shift and $\lambda$ is the exponential parameter, then the density has the form,

$$\lambda \cdot e^{-(\lambda(x-a))}$$ (4.1)

The maximum likelihood estimate for $a$ is the minimum of the $M$ largest values. The maximum likelihood estimate for $\lambda$ is $M/k$ where $k$ is the sum of deviations from $a$ over the $M$ largest values. Considering two sets, 0 and 1, of clutter peak values following the same distribution but generated from files of different sizes, where set 1 is $B$ times larger than set 0. Then if $M$ values are larger than threshold $b$ in set 0, $BM$ peaks larger than $b$ should appear in set 1. These peaks will be drawn from the shifted exponential with the same parameters as the original set. Supposing at most $k$ false alarms are desired in set 1, then there is a level $L$ such that only $k$ of $BM$ peaks are above $L$: i.e. only one of $BM/k$ peaks are above $L$. The threshold, $L$, is found by solving the inverse shifted exponential given above in equation 4.1. Therefore, given the relative dimensions in range, time, and number of beams of two scenes, the false alarm rate in one is calculated from the false alarm rate of the other. The threshold level necessary to achieve this rate is determined the parameters of the shifted exponential fit to the clutter peak distribution.
CHAPTER 5

5. DETECTION RESULTS ON EXPERIMENTAL DATA

This chapter presents results from extensive performance benchmarking runs done with the UCTP system on six experimental clutter scenes and a range of simulated injected periscope signatures. The objective of this analysis was to determine the signal strengths needed for the UCTP system to detect a target as a function of range and sea state. The results are presented in a series of plots of probability of detection versus signal to clutter-plus-noise ratio in the six clutter scenes. The analysis was performed at two false alarm rates. The first rate was 1 false alarm per day over all radar beams out to 10 nm. The second one was a factor of $10^6$ higher, based on the assumption of an exogenous false alarm limiter that could, by statistical pattern recognition techniques, reduce the false alarm rate by a factor of $10^6$. In this form the UCTP results can be compared to the front end of other processing systems that utilize the false alarm limiter.

In Section 5.1 we describe the six clutter scenes and their characteristics. Section 5.2 describes in detail the steps in the benchmarking process. Results are presented for the thresholds necessary to achieve the two false alarm rates mentioned above as a function of signal strength in the six clutter scenes. Section 5.3 contains the probability of detection results for all the cases considered.

5.1 Description of Data

The clutter data we used was obtained from an experimental radar mounted on a cliff looking out into the ocean. Data was acquired in two sea states: sea state 3 and sea state 4. The
radar was sampled at approximately 1 foot resolution for 512 samples over a single azimuth. The center of the range interval was set to one of three ranges: 3.5 nautical miles, 6.5 nm, and then 9.5 nm. The pulse rate of the radar was near 2 kHz, but the data used for analysis was downsampling to 5 Hz in order to simulate data along a single azimuth from a scanning high pulse rate radar with a 5 Hz scan rate. Portions of artifact-free data were found, amounting to roughly 15–20 minutes (4,500–6,000 pulses) for each scene. One of the scenes (sea state 3, range 6.5 nm) was shorter (only 4 minutes long).

We can comment on the qualitative features of the data as a function of range. The 3.5 nm scene contains very strong, clearly defined clutter patches. In the 6.5 nm scene, the clutter patches are visible, but they are not as dense as in the previous scene. The speckle between the patches caused by system noise is much stronger than in the 3.5 nm scene. The receiver gain was set higher in the 6.5 nm scene to maintain a fixed mean. The trends continue in the 9.5 nm scene.

The following three tables present some of the characteristics of the scenes. The mean clutter-plus-noise was obtained by averaging all the data in the file. The system noise level was estimated by choosing a small area containing no clutter patches and averaging the data there.

Length of each scene in minutes:

<table>
<thead>
<tr>
<th>Range</th>
<th>Sea State 3</th>
<th>Sea State 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.5 nm</td>
<td>15.4</td>
<td>22.0</td>
</tr>
<tr>
<td>6.5 nm</td>
<td>3.9</td>
<td>23.0</td>
</tr>
<tr>
<td>9.5 nm</td>
<td>15.4</td>
<td>23.0</td>
</tr>
</tbody>
</table>

Table 5.1

Mean Clutter-Plus-Noise in dB

<table>
<thead>
<tr>
<th>Range</th>
<th>Sea State 3</th>
<th>Sea State 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.5 nm</td>
<td>18.6</td>
<td>22.6</td>
</tr>
<tr>
<td>6.5 nm</td>
<td>24.9</td>
<td>21.9</td>
</tr>
<tr>
<td>9.5 nm</td>
<td>24.8</td>
<td>25.2</td>
</tr>
</tbody>
</table>

Table 5.2
Estimated System Noise in dB

<table>
<thead>
<tr>
<th></th>
<th>Sea State 3</th>
<th>Sea State 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.5 nm</td>
<td>16.1</td>
<td>20.6</td>
</tr>
<tr>
<td>6.5 nm</td>
<td>24.1</td>
<td>20.3</td>
</tr>
<tr>
<td>9.5 nm</td>
<td>24.3</td>
<td>24.7</td>
</tr>
</tbody>
</table>

Table 5.3

5.2 Processing Method

Our primary focus over the past nine months has been assessing the performance of the UCTP system. This section describes the methods used for the benchmarking. There are three phases in the analysis: clutter only runs, target runs with known signal level, and then target runs with unknown signal level. Those three phases are described in this section.

5.2.1 Clutter Peaks

The first phase of the analysis studied the clutter scenes with no targets injected in order to determine the fluctuations of the detection statistic, and then to set thresholds. The analysis is broken into five steps:

Step A1: Clutter Tracker. We first ran the clutter tracker on each scene. The clutter tracker computed the data covariances, optimal filter coefficients, and residual variance of the estimator. This step develops the two-dimensional FIR filter used to estimate the mean clutter. The clutter tracker then filters the log-intensity data to produce the clutter mean estimate. For our performance benchmarking we used a filter mask which extended ±5 scans and ±10 range bins from the cell to be estimated.

The residual variance is computed from the clutter tracker and then used in the likelihood ratio tracker. The results for the residual variance are:
### Table 5.4

<table>
<thead>
<tr>
<th></th>
<th>Sea State 3</th>
<th>Sea State 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.5 nm</td>
<td>0.485</td>
<td>0.334</td>
</tr>
<tr>
<td>6.5 nm</td>
<td>0.296</td>
<td>0.387</td>
</tr>
<tr>
<td>9.5 nm</td>
<td>0.428</td>
<td>0.239</td>
</tr>
</tbody>
</table>

**Step A2: Target Tracker.** For each scene we chose a range of target strengths where the expected detection performance would vary from 0% to 100%. The range for each scene was determined iteratively. The numerical value of the likelihood ratio outlined above depends on the assumed target strength in the processor. For each assumed signal level we ran the target tracker with the clutter data as input, and the clutter mean estimate from Step A1. The clutter tracker did not have to be run each time because the clutter estimate does not depend on the assumed signal strength. We limited the range of target levels to less than 20 dB because the raw radar had a dynamic range of 42 dB and the mean clutter levels were approximately 20 dB. The log-speckle variance was discussed in Section 4.1. The numbers used for the six scenes are shown in the following table.

### Table 5.5

<table>
<thead>
<tr>
<th></th>
<th>Sea State 3</th>
<th>Sea State 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.5 nm</td>
<td>0.95</td>
<td>1.24</td>
</tr>
<tr>
<td>6.5 nm</td>
<td>1.39</td>
<td>1.11</td>
</tr>
<tr>
<td>9.5 nm</td>
<td>1.19</td>
<td>1.45</td>
</tr>
</tbody>
</table>

**Step A3: Local Peaks.** As the target tracker was running, we monitored the detection statistic, which for this analysis was the maximum value of the likelihood ratio surface over the complete state space. The goal was to count how many distinct times the detection statistic rose above a threshold as a function of the threshold level. In order to avoid double counting threshold exceedances, we considered only values that were local maximums over some neighborhood in time, range, and velocity. The output of this step was a sorted list of peak values of the detection statistic.

**Step A4: Peak Value Parametric Form.** For each scene and each assumed target level, we had a list of the highest peaks. In order to extrapolate exceedance counts to lower false alarm rates we estimated a parametric form for each set. The chosen form was exponential with two parameters—a shift and a scale factor—discussed in Section 4.4.
Step A5: Threshold Extrapolation. The desired false alarm rate for the UCTP system was set to 1 per day over the complete search area. In one case we assumed that an exogenous false alarm limiter was employed on small sections of the intensity data to statistically discriminate wide clutter patches from the narrow periscope tracks after being cued by the UCTP system. This assumption was made to allow a comparison to other systems that included a false alarm limiter. We assumed the false alarm limiter reduced the false alarm rate by a factor of $10^6$ without affecting the target detections. Converting the 1 per day rate over the full search area (256 beams, and 60,000 range bins) to 512 range bins in a single azimuth, leads to 1 false alarm allowed every 43 minutes. The extrapolation was consequently a factor of approximately 3 (or 10 for the sea state 3, range 6.5 scene) from the size of the clutter scene. Without the false alarm limiter, the UCTP system's false alarm rate by itself was $10^6$ lower than this number, meaning that one false alarm would be allowed for every 43 million minutes of data processed on a single beam over 512 range bins. Since this is such a large jump from the available data, there is greater uncertainty in the PD results.

Computer Run Time. The approximate timing of the runs on a Sun Sparc floating point processor operating at 60 MHz was as follows. Calculating the filter coefficients and performing the likelihood ratio processing are nearly linear functions of the number of scans in a data set. To calculate the filter coefficients of data scenes with 5100 scans, i.e. 17 minutes long, takes 14 minutes. The clutter estimation stage for the same size scene takes less than 2 minutes. Also, the likelihood ratio tracker takes approximately 54 minutes. Thus all three stages take slightly more than four times real time. Assuming the filter coefficients do not need to be calculated constantly, the clutter estimate and likelihood ratio tracking perform at three and a third times real time for this processor.

5.2.2 False Alarm Rate Extrapolation

In order to extrapolate to smaller false alarm rates than seen in the clutter scenes, we fit an analytic form to the observed clutter peak distribution. We took the top 10 peaks from each run and approximated their distribution by

$$
\lambda \cdot e^{\lambda(x-a)}
$$

We could then extrapolate to the two desired false alarm rates.
5.2.2.1 Extrapolation Parameters

Tables 5.6–5.11 show the parameter fits generated from the clutter runs.

Sea State 3, Range 3.5 nm

<table>
<thead>
<tr>
<th>SCNR</th>
<th>Lambda</th>
<th>Shift</th>
</tr>
</thead>
<tbody>
<tr>
<td>-2</td>
<td>3.13</td>
<td>-12.10</td>
</tr>
<tr>
<td>0</td>
<td>2.33</td>
<td>-11.66</td>
</tr>
<tr>
<td>2</td>
<td>2.81</td>
<td>-11.03</td>
</tr>
<tr>
<td>4</td>
<td>3.12</td>
<td>-10.63</td>
</tr>
<tr>
<td>6</td>
<td>2.41</td>
<td>-10.50</td>
</tr>
<tr>
<td>8</td>
<td>1.73</td>
<td>-10.34</td>
</tr>
<tr>
<td>10</td>
<td>1.57</td>
<td>-10.17</td>
</tr>
<tr>
<td>12</td>
<td>1.66</td>
<td>-10.07</td>
</tr>
<tr>
<td>16</td>
<td>2.12</td>
<td>-9.84</td>
</tr>
<tr>
<td>20</td>
<td>0.99</td>
<td>-10.17</td>
</tr>
</tbody>
</table>

Table 5.6

Sea State 3, Range 6.5 nm

<table>
<thead>
<tr>
<th>SCNR</th>
<th>Lambda</th>
<th>Shift</th>
</tr>
</thead>
<tbody>
<tr>
<td>-4</td>
<td>6.01</td>
<td>-11.12</td>
</tr>
<tr>
<td>0</td>
<td>4.12</td>
<td>-10.10</td>
</tr>
<tr>
<td>2</td>
<td>2.52</td>
<td>-9.62</td>
</tr>
<tr>
<td>4</td>
<td>2.71</td>
<td>-8.78</td>
</tr>
<tr>
<td>6</td>
<td>2.31</td>
<td>-7.4</td>
</tr>
<tr>
<td>8</td>
<td>1.23</td>
<td>-5.95</td>
</tr>
<tr>
<td>10</td>
<td>1.00</td>
<td>-3.73</td>
</tr>
<tr>
<td>12</td>
<td>1.42</td>
<td>-1.34</td>
</tr>
<tr>
<td>16</td>
<td>0.93</td>
<td>-0.04</td>
</tr>
</tbody>
</table>

Table 5.7
Sea State 3, Range 9.5 nm

<table>
<thead>
<tr>
<th>SCNR</th>
<th>Lambda</th>
<th>Shift</th>
</tr>
</thead>
<tbody>
<tr>
<td>-4</td>
<td>8.38</td>
<td>-11.02</td>
</tr>
<tr>
<td>0</td>
<td>4.74</td>
<td>-9.83</td>
</tr>
<tr>
<td>2</td>
<td>4.75</td>
<td>-8.74</td>
</tr>
<tr>
<td>4</td>
<td>2.09</td>
<td>-7.13</td>
</tr>
<tr>
<td>6</td>
<td>0.90</td>
<td>-5.27</td>
</tr>
<tr>
<td>8</td>
<td>0.54</td>
<td>-3.28</td>
</tr>
<tr>
<td>10</td>
<td>0.53</td>
<td>-0.56</td>
</tr>
<tr>
<td>12</td>
<td>0.39</td>
<td>1.21</td>
</tr>
<tr>
<td>16</td>
<td>0.44</td>
<td>4.06</td>
</tr>
</tbody>
</table>

Table 5.8

Sea State 4, Range 3.5 nm

<table>
<thead>
<tr>
<th>SCNR</th>
<th>Lambda</th>
<th>Shift</th>
</tr>
</thead>
<tbody>
<tr>
<td>-6</td>
<td>3.03</td>
<td>-11.33</td>
</tr>
<tr>
<td>-2</td>
<td>2.77</td>
<td>-10.27</td>
</tr>
<tr>
<td>2</td>
<td>1.62</td>
<td>-8.62</td>
</tr>
<tr>
<td>4</td>
<td>1.42</td>
<td>-6.93</td>
</tr>
<tr>
<td>6</td>
<td>1.28</td>
<td>-5.23</td>
</tr>
<tr>
<td>8</td>
<td>0.62</td>
<td>-4.23</td>
</tr>
<tr>
<td>10</td>
<td>0.45</td>
<td>-2.74</td>
</tr>
<tr>
<td>12</td>
<td>0.53</td>
<td>-0.47</td>
</tr>
<tr>
<td>14</td>
<td>0.57</td>
<td>0.98</td>
</tr>
<tr>
<td>18</td>
<td>0.63</td>
<td>2.20</td>
</tr>
</tbody>
</table>

Table 5.9
Sea State 4, Range 6.5 nm

<table>
<thead>
<tr>
<th>SCNR</th>
<th>Lambda</th>
<th>Shift</th>
</tr>
</thead>
<tbody>
<tr>
<td>-4</td>
<td>6.05</td>
<td>-11.13</td>
</tr>
<tr>
<td>0</td>
<td>2.45</td>
<td>-9.81</td>
</tr>
<tr>
<td>4</td>
<td>1.07</td>
<td>-8.31</td>
</tr>
<tr>
<td>6</td>
<td>1.07</td>
<td>-7.27</td>
</tr>
<tr>
<td>8</td>
<td>1.03</td>
<td>-5.95</td>
</tr>
<tr>
<td>10</td>
<td>1.06</td>
<td>-4.44</td>
</tr>
<tr>
<td>12</td>
<td>0.85</td>
<td>-3.31</td>
</tr>
<tr>
<td>14</td>
<td>1.15</td>
<td>-1.50</td>
</tr>
<tr>
<td>16</td>
<td>0.78</td>
<td>-0.91</td>
</tr>
<tr>
<td>20</td>
<td>0.43</td>
<td>-1.12</td>
</tr>
</tbody>
</table>

Table 5.10

Sea State 4, Range 9.5 nm

<table>
<thead>
<tr>
<th>SCNR</th>
<th>Lambda</th>
<th>Shift</th>
</tr>
</thead>
<tbody>
<tr>
<td>-8</td>
<td>6.60</td>
<td>-11.04</td>
</tr>
<tr>
<td>-4</td>
<td>3.07</td>
<td>-10.21</td>
</tr>
<tr>
<td>0</td>
<td>1.96</td>
<td>-8.30</td>
</tr>
<tr>
<td>2</td>
<td>2.13</td>
<td>-6.55</td>
</tr>
<tr>
<td>4</td>
<td>1.78</td>
<td>-4.70</td>
</tr>
<tr>
<td>6</td>
<td>1.10</td>
<td>-2.63</td>
</tr>
<tr>
<td>8</td>
<td>0.97</td>
<td>-0.24</td>
</tr>
<tr>
<td>10</td>
<td>0.70</td>
<td>2.47</td>
</tr>
<tr>
<td>12</td>
<td>0.45</td>
<td>4.80</td>
</tr>
<tr>
<td>16</td>
<td>0.32</td>
<td>6.70</td>
</tr>
</tbody>
</table>

Table 5.11

5.2.2.2 Threshold Settings

The first set of results shows the thresholds determined in step A5. Figure 5.1 shows the threshold for log-likelihood (natural logarithm) as a function of signal to clutter-plus-noise ratio (SCNR) for the three scenes in sea state 3. Figure 5.2 shows the same quantities for the
three scenes in sea state 4. For each scene we determined the clutter-plus-noise level by averaging the intensity data over the whole scene. We see that the thresholds increase slightly with assumed signal level. For very low signal levels the thresholds are near -10 log-likelihood. For the sea state 3 scenes, at a given SCNR the thresholds increase with range from the radar; the thresholds for the 3.5 nm scene are the lowest for all signal levels. For the sea state 4 scenes the ordering from lowest to highest is 6.5 nm, 3.5 nm, and then 9.5 nm. Comparing across sea states, the 9.5 nm thresholds are very similar, and the 6.5 nm thresholds are also very similar. The greatest difference is in the 3.5 nm scenes, where the sea state 3 thresholds are much lower than the sea state 4 thresholds for the same SCNR.

Figures 5.3 and 5.4 show the results for the lower false alarm rate, where the exogenous false alarm limiter is not used so the UCTP must by itself maintain 1 false alarm per day. The figures have roughly the same characteristics, but note that the vertical scale is different. In the second two figures the scale runs from -10 to 70. The thresholds are higher in each case. For example, with a signal of 10 dB in the sea state 3, range 3.5 nm scene, we needed to set the threshold at approximately -7.5 log-likelihood units at the higher false alarm rate, but needed to lift it to +1 to enforce the lower false alarm rate. The absolute changes occur in the 9.5 nm scenes.

5.2.3 Target Detection With Known Signal Level

The second phase worked with the clutter scenes after injecting realistic submarine periscope signatures. It also consisted of five steps.

Step B1: Target Injection. Another participant in the UCTP program (Dynamics Technology, Inc.) provided FORTRAN code to inject target signals. The target response adds in random phase with the clutter. The point statistics of the target are Rice-squared. We did two sets of runs, one with the Rice parameter $R$ equal to 3.0 and the other with $R$ set to 0.3. The Rice parameter is the ratio of the coherent signal to the incoherent. The correlation time of the target was 1 second.

For each scene we simulated 100 periscope signatures with 5 second duration using the model described above. The 100 signatures were injected into the clutter scenes in locations separated far enough apart so the targets did not interfere with each other in the likelihood ratio surface. After adding the target response to the clutter data, the intensity data was clipped at the same level the hardware had clipped the clutter data (approximately 40 dB).
Step B2: Clutter Tracker. The next step was to run the clutter tracker on the scenes with the injected targets. For these runs we did not recompute the data covariances and derive new filter coefficients. This is reasonable because when the UCTP system is operating aboard a ship there will very rarely be periscopes in the data, and certainly not 100 periscopes. We ran the filtering section of the clutter tracker over the scene with the injected targets since the returns from a periscope will be present in the filter mask, and will slightly elevate the mean estimate. The altered clutter mean estimate will affect the likelihood ratio calculation.

Step B3: Target Tracker. We then ran the target tracker using data from step B1 and the estimate from step B2. Within the target tracker the assumed signal level was set equal to the true level used for the injection in step B1.

Step B4: Target Peak Levels. For each injected target we found the maximum value in the likelihood ratio surface caused by that target during its presence.

Step B5: Probability of Detection. Using the threshold determined in step A5, we counted the number of targets that produced likelihood ratio values above that level. The fraction above the level became the probability of detection. Results from these runs are presented in Section 5.3.

5.2.4 Target Detection With Unknown Signal Level

The previous phase analyzed the detection performance when the target tracker used a signal level matched to the true signal level. In a real system, the target tracker will not know the true signal level. It must pick a single level, and then use that for all targets. The analysis of this realistic case is the subject of the third phase.

Step C1: Signal Level. For each scene we choose a signal level to use in the target tracker. For injected targets with that same level the results will be identical to the those found in Phase B. For other signal levels, the processor will be slightly mismatched and therefore slightly suboptimal. We wanted to maintain the same performance for targets near 50% probability of detection, so for each scene we examined the results from Phase B and chose a signal level closest to 50% PD. Tables 5.12 and 5.13 show the signal levels chosen.
Signal Level Chosen for R=3.0 Runs in dB:

<table>
<thead>
<tr>
<th></th>
<th>Sea State 3</th>
<th>Sea State 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.5 nm</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>6.5 nm</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>9.5 nm</td>
<td>6</td>
<td>8</td>
</tr>
</tbody>
</table>

Table 5.12

Signal Level Chosen for R=0.3 Runs in dB:

<table>
<thead>
<tr>
<th></th>
<th>Sea State 3</th>
<th>Sea State 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.5 nm</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>6.5 nm</td>
<td>6</td>
<td>8</td>
</tr>
<tr>
<td>9.5 nm</td>
<td>10</td>
<td>10</td>
</tr>
</tbody>
</table>

Table 5.13

Step C2: Target Tracker. We then reran the target tracker on the same data and estimated mean files used in step B3. The only difference was that assumed target strength in the target tracker.

Step C3: Target Peak Levels. For each target we found the peak in the likelihood ratio surface due to that target.

Step C4: Probability of Detection. The final step was to count the number of targets that produced likelihoods above the threshold. The threshold used was the one from Phase A associated with the assumed target strength, not the threshold for the true target strength. The combination of different numerical values of the likelihood ratio and different thresholds altered the PD results from Phase B. The differences are greatest for targets with the largest mismatch between the true signal level and the assumed signal level. Results from these runs are presented in the next section.
5.3 Probability of Detection Results

The final output of the performance benchmarking is a set of curves showing the probability of detection versus signal strength the six clutter scenes, with both known (matched) and unknown (potentially mismatched) signal levels, and two false alarm rates. Those curves are presented and discussed in this section, first for the case with the exogenous false alarm limiter and then for the case without.

5.3.1 With Exogenous False Alarm Limiter (UCTP FAR = 12 / sec)

Figures 5.5 and 5.6 show the results for the R=3.0 case with known signal level (Phase B described above). At each of the three ranges we see the PD’s rising monotonically with signal strength. Measured with respect to the sum of mean clutter and noise power, the 3.5 nm scene presents the easy case to detect the target, the 6.5 nm scene the next, and the 9.5 nm scene the most difficult. Looking at Figure 5.1 we see that this is also the order of the thresholds: 3.5 nm has the lowest thresholds and 9.5 nm the highest. The SCNR needed to detect the targets ranges from 2 dB to 6 dB in sea state 3. In sea state 4, the graphs are close together. The 50% PD points are 4.5 to 6.5 dB. The 9.5 nm scene presents the most difficult detection environment, while the other two are roughly the same. Looking at Figure 5.2 we see that the order of scenes is the same for the thresholds and the 50% PD points.

Figure 5.7 and 5.8 show the analogous results with Rice parameter R=0.3. We see that these targets are slightly more difficult to detect than their counterparts with R=3.0. Another feature is that the curves rise more slowly with signal strength. The reason for this is that the fluctuating signal may be lower than average over a majority of the 25 scans. Some targets, even with high mean signal strength, may not appear very strong in a particular random realization. The 50% PD points in sea state 3 are in the range of 3.5 to 10 dB. In sea state 4 (Figure 5.8) we see the same effect of the curves flattening out. The 50% PD points in sea state 4 range from 6.5 to 10 dB.

The next four figures present results for the case where the signal level is unknown and must be set at one level in each scene (Phase C). Figures 5.9 and 5.10 treat the R=3.0 case. Table 5.12 shows the signal levels chosen. These are approximately the 50% PD signal levels, so the results with known and unknown signal level agree at that point. Comparing Figures 5.5 and 5.9 we see that with a fixed signal level the curves are much steeper. In the 3.5 nm scene (solid line) the signal level chosen was 2 dB. For true signal levels below 2 dB the PD’s
degrade slightly. Above 2 dB we would expect the results to degrade slightly as well (since the processor is mismatched there), but we see that the PD’s improve a little bit. The reason for this is that the thresholds are increasing with signal level, so by using a lower signal level in the processor, we can set the thresholds slightly lower, and even though the target likelihood accumulation is less, the net effect is that more targets pass the lower threshold. Similar results hold in the sea state 4 scenes. The 50% PD points are listed in Table 5.14.

Signal Level Needed for 50% and UCTP FAR = 12 / sec in R=3.0 Runs (dB):

<table>
<thead>
<tr>
<th></th>
<th>Sea State 3</th>
<th>Sea State 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.5 nm</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>6.5 nm</td>
<td>3.5</td>
<td>5</td>
</tr>
<tr>
<td>9.5 nm</td>
<td>6</td>
<td>7</td>
</tr>
</tbody>
</table>

Table 5.14

Figures 5.11 and 5.12 show analogous results for the R=0.3 case. In this analysis the signal levels chosen in each scene were listed in Table 5.13. In these figures we do not see the graphs becoming steeper with fixed signal level. The fluctuating target is less sensitive to a mismatch in assumed and true target strengths. In each case the graphs are little changed. The signal levels needed for 50% PD are listed in Table 5.15.

Signal Level Needed for 50% and UCTP FAR = 12 / sec in R=0.3 Runs (dB):

<table>
<thead>
<tr>
<th></th>
<th>Sea State 3</th>
<th>Sea State 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.5 nm</td>
<td>3.5</td>
<td>9</td>
</tr>
<tr>
<td>6.5 nm</td>
<td>5.5</td>
<td>6.5</td>
</tr>
<tr>
<td>9.5 nm</td>
<td>9.5</td>
<td>10</td>
</tr>
</tbody>
</table>

Table 5.15

One effect we are starting to see however is the degradation in PD for true target strengths that are much larger than the assumed target strength. The sea state 3, range 3.5 nm shows this effect. The measurement log-likelihood ratio is similar, but not identical, to the log-likelihood ratio shown in Figure 3.7: for received signals much larger (factors of 10-100) than the estimated clutter mean, the log-likelihood drops off since the most credible explanation is that the data is a clutter spike rather than a target. For the sea state 3, range 3.5 nm scene, we have set the assumed signal level to 4 dB. When targets appear with a mean level of 15 dB and
higher, the stronger returns in the target signal are discounted, making the total target likelihood along the track decrease. At the same time, the thresholds are rising with signal strength, and the combination of these two effects reduces the PD. One way out of this problem would be to run two processors side-by-side, one set for the weaker targets and one set for a stronger target (say 15 dB in the sea state 3, range 3.5 nm scene). The false alarm rate might double, but we could increase the thresholds slightly in each processor. By running the two processors we could have higher PD over a wider range. Another option is to incorporate a probability distribution on target strength and compute the associated measurement likelihood ratio.

5.3.2 Without Exogenous False Alarm Limiter (UCTP FAR = 1 / day)

The other case examined is with the UCTP system running by itself and maintaining one false alarm per day over the full surveillance region. As noted in Section 5.2.2.2, the thresholds are much higher. The figures are arranged in the same order as in the previous section.

Figures 5.13 through 5.16 present results for the known signal cases. The first two are for Rice parameter $R=3.0$. We see that the curves are more jagged than the corresponding curves are the higher false alarm rate. This is mainly due to the threshold curves. Remember that the extrapolation parameters were computed independently for each signal level and we are extrapolating six orders of magnitude so slight differences in the lambda parameters are magnified. As expected, the signal levels needed to achieve similar performance have increased. For example, comparing Figures 5.13 and 5.5, we see that in sea state 3 the 50% PD points have increased by 3, 6.5, and 8 dB, in the 3.5, 6.5, and 9.5 nm scenes, respectively. The 9.5 nm scene suffers the most because its thresholds have been raised the most. For sea state 4, the 50% PD points have moved to the right by 7.5, 5, and 9 dB for ranges 3.5, 6.5, and 9.5 nm, respectively. For the $R=0.3$ case, we see that performance has degraded as well. Visually extrapolating from the results shown, we see that an additional 5-10 dB signal strength is needed to compensate for the much lower false alarm rate.

The last set of four figures show the results for the unknown signal level case. In these runs we kept the signal level used in the processor the same as before; these levels are shown in Tables 5.12 and 5.13. The levels do not correspond to the 50% PD points for the low false alarm rate, but since the thresholds are generally rising quickly with assumed signal strength, it might be better to choose a low assumed signal strength.
Figures 5.17 and 5.18 show the results for \( R=3.0 \). The curves are no longer jagged since we are using the same threshold across all true signal levels. Table 5.16 shows the signal level needed to obtain 50% PD in these cases.

Signal Level Needed for 50% and UCTP FAR = 1 / day in \( R=3.0 \) Runs (dB):

<table>
<thead>
<tr>
<th></th>
<th>Sea State 3</th>
<th>Sea State 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.5 nm</td>
<td>5</td>
<td>8.5</td>
</tr>
<tr>
<td>6.5 nm</td>
<td>6</td>
<td>9.5</td>
</tr>
<tr>
<td>9.5 nm</td>
<td>11</td>
<td>10</td>
</tr>
</tbody>
</table>

Table 5.16

We see in these curves the drawback in choosing a low assumed signal level. As discussed in the last section, the detection performance degrades for signals much larger than the assumed level. For example, in the sea state 3, range 3.5 nm case, the assumed level is 2 dB. The PS increases for signal levels up to 10 dB, and then it decreases after that. This is also seen in the 3.5 and 6.5 nm, sea state 4 scenes when the signal level is above 14 dB versus the assumed level of 6 dB.

Figures 5.19 and 5.20 show the results for \( R=0.3 \). The results shown are similar to the known signal level cases since the fluctuating target is less sensitive to target strength mismatch and the range of levels shown is not far from the assumed levels in Table 5.13. The 50% PD signal levels are shown in Table 5.17

Signal Level Needed for 50% and UCTP FAR = 1 / day in \( R=0.3 \) Runs (dB):

<table>
<thead>
<tr>
<th></th>
<th>Sea State 3</th>
<th>Sea State 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.5 nm</td>
<td>7</td>
<td>&gt;18</td>
</tr>
<tr>
<td>6.5 nm</td>
<td>9</td>
<td>13.5</td>
</tr>
<tr>
<td>9.5 nm</td>
<td>&gt;16</td>
<td>15.5</td>
</tr>
</tbody>
</table>

Table 5.17

To summarize the differences between the two false alarm rates considered we can compare Table 5.14 to Table 5.16 and Table 5.15 to Table 5.17 and see how much more signal level is needed to achieve a 50% probability of detection when we lower the false alarm rate by a factor of one million. For the more constant target \( (R=3.0) \) we see that the increased levels
are 3, 2.5, and 5 dB in sea state 3 for ranges 3.5, 6.5, and 9.5 nm, respectively, and in sea state 4 the signal needs to be increased 3.5, 4.5, and 3 dB at the same set of ranges. This translates to approximately one half to one dB for each order of magnitude change in FAR. For the more fluctuating target ($R=0.3$), the results are 3.5, 3.5, more than 6.5 dB in sea state 3 and more than 9, 7, and 5.5 dB in sea state 4, at ranges of 3.5, 6.5, and 9.5 nm, respectively. In this case the change is signal level is 1 to 1.5 dB for each order of magnitude change in FAR. This might be improved by choosing slightly higher signal levels than we did for the fluctuating targets.
REFERENCES


FIGURES

Figures 3.1 through 3.4 and Figures 5.1 through 5.20 are presented in the following pages.
Figure 3.1: Target Capture Effect for Fluctuating Target
Figure 3.2: ROC Curves -- 5 dB Target

Probability of Detection

False Alarm Rate

- 10 kt upwind
- 10 kt. cross
- 18 kt upwind
- 18 kt cross

1 FA / day / 10 nm  1 FA / hr / 10 nm  1 FA / day / beam  1 FA / hr / beam
Figure 3.3: ROC Curves -- 2.5 dB Target

Probability of Detection

False Alarm Rate

- 10 kt upwind
- 10 kt cross
- 18 kt upwind
- 18 kt cross
Figure 3.4: ROC Curves -- 0 dB Target

False Alarm Rate

Probability of Detection

- - - - 10 kt upwind
- - - - 10 kt cross
- - - - 18 kt upwind
- - - - 18 kt cross

1 FA / day / 10 nm  1 FA / hr / -10 nm  1 FA / day / beam  1 FA / hr / beam
Figure 5.1: UCTP Threshold Settings
Sea State 3

Likelihood Ratio Threshold

Signal to Clutter-Plus-Noise Ratio (dB) Used in UCTP Processor
Figure 5.2: UCTP Threshold Settings
Sea State 4

Likelihood Ratio Threshold

Signal to Clutter-Plus-Noise Ratio (dB)
Used in UCTP Processor
Figure 5.4: UCTP Threshold Settings, Sea State 4
FAR = 1 / day

Likelihood Ratio Threshold

Signal to Clutter-Plus-Noise Ratio (dB)
Used in UCTP Processor

- 3.5 nm
- 6.5 nm
- 9.5 nm
Figure 5.5: Sea State 3, Known Signal Level, $R = 3.0$
Figure 5.6: Sea State 4, Known Signal Level, $R = 3.0$
Figure 5.7: Sea State 3, Known Signal Level, R = 0.3
Figure 5.8: Sea State 4, Known Signal Level, R = 0.3

- Probability of Detection
- Signal to Clutter-Plus-Noise Ratio (dB)

Graph showing the relationship between probability of detection and signal to clutter-plus-noise ratio for different signal levels (3.5 nm, 6.5 nm, 9.5 nm).
Figure 5.9: Sea State 3, Unknown Signal Level, R = 3.0

Signal to Clutter-Plus-Noise Ratio (dB)

Probability of Detection

- 3.5 nm
- 6.5 nm
- 9.5 nm
Figure 5.10: Sea State 4, Unknown Signal Level, $R = 3.0$
Figure 5.11: Sea State 3, Unknown Signal Level, $R = 0.3$
Figure 5.12: Sea State 4, Unknown Signal Level, $R = 0.3$
Figure 5.13: Sea State 3, Known Signal Level, R = 3.0
FAR = 1 / day

- Probability of Detection

Signal to Clutter-Plus-Noise Ratio (dB)

- 3.5 nm
- 6.5 nm
- 9.5 nm
Figure 5.14: Sea State 4, Known Signal Level, R = 3.0
FAR = 1 / day

Probability of Detection

Signal to Clutter-Plus-Noise Ratio (dB)
Figure 5.15: Sea State 3, Known Signal Level, R = 0.3
FAR = 1 / day

- 3.5 nm
- 6.5 nm
- 9.5 nm
Figure 5.16: Sea State 4, Known Signal Level, \( R = 0.3 \)
FAR = 1 / day

![Graph showing the probability of detection as a function of signal to clutter-plus-noise ratio (dB) for different signal levels.]
Figure 5.17: Sea State 3, Unknown Signal Level, R = 3.0
FAR = 1 / day

Probability of Detection

Signal to Clutter-Plus-Noise Ratio (dB)

- 3.5 nm
- 6.5 nm
- 9.5 nm
Figure 5.18: Sea State 4, Unknown Signal Level, R = 3.0
FAR = 1 / day

Probability of Detection

Signal to Clutter-Plus-Noise Ratio (dB)
Figure 5.19: Sea State 3, Unknown Signal Level, \( R = 0.3 \)
\[ \text{FAR} = 1 / \text{day} \]

- Probability of Detection

- Signal to Clutter-Plus-Noise Ratio (dB)
Figure 5.20: Sea State 4, Unknown Signal Level, $R = 0.3$
FAR = 1 / day

Probability of Detection

Signal to Clutter-Plus-Noise Ratio (dB)

- 3.5 nm
- 6.5 nm
- 9.5 nm