MULTISENSOR TRACK INITIATION METHOD THAT ADDRESSES THE MISSING MEASUREMENT PROBLEM

by

Robert James Pawlak

Dissertation submitted to the Faculty of the

Virginia Polytechnic Institute and State University

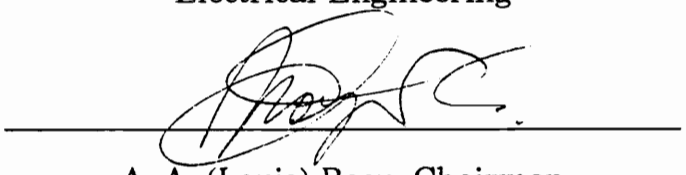
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

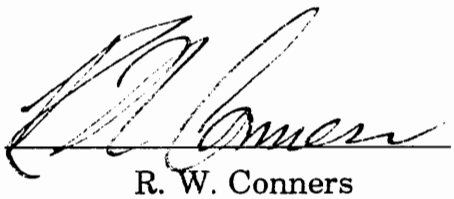
Electrical Engineering

APPROVED:

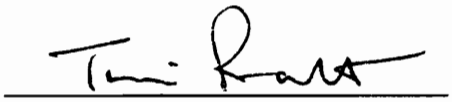


A. A. (Louis) Beex, Chairman

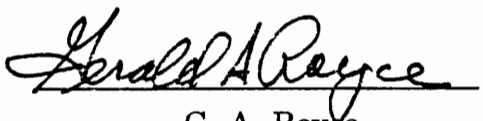
G.S. Brown



R. W. Conners



T. Pratt



G. A. Royce

December, 1992
Blacksburg, Virginia

MULTISENSOR TRACK INITIATION METHOD THAT ADDRESSES THE MISSING MEASUREMENT PROBLEM

by

Robert James Pawlak

Committee Chairman: A. A. (Louis) Beex
Electrical Engineering

(ABSTRACT)

A method for integrating multisensor data for the purpose of track-initiation using horizon infrared and radar data is proposed. This multisensor track initiation (MSTI) method extends contemporary data fusion techniques so as to address the problem of missing measurements. The missing measurement phenomenon occurs due to a variety of reasons, the foremost of which is variation in sensor detection performance due to environmental factors.

The proposed MSTI method requires only the results of spatial feature tests that are performed on sensor data sequences. The formation of data sequences and the derivation of feature tests to integrate horizon radar and infrared data of differing resolutions is addressed.

Results are presented that detail the performance of the MSTI technique when operating on simulated data. It is shown that the statistical performance of the MSTI technique is better than or equal to that of the AND algorithm for

a representative set of scenarios. The sensitivity of the MSTI method to variations in assumed feature test and data sequence statistics is also addressed.

ACKNOWLEDGEMENT

I am grateful to Dr. Louis Beex for his guidance and encouragement throughout the course of this research.

I would additionally like to thank Karl H. Krueger for suggesting alternative approaches to the problem.

I would like to thank Ron Stapleton for his patience, and advice on propagation issues.

I am especially thankful to my wife Michelle Kienholz for her emotional support and understanding (and her graphics assistance on Figure 1.1).

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction

For more than a decade, the surface Navy has been searching for reliable methods of detecting sea-skimming missiles such as the Exocet. The primary defense against this threat today is a radar, which unfortunately has several weaknesses. First, the radar is susceptible to multipath in the region around the horizon; second, the radar is clutter limited in this region; and third, ducting produces changes in propagation conditions and hence in signal to noise ratio (SNR). The inefficacy of this type of system has been demonstrated by the Stark incident [1].

The inclusion of an infrared system to improve the detection performance of existing shipboard radar systems has been proposed based on the nature of the threat. The speed of the incoming missile results in the production of heat and hence of radiation in the infrared bands. This radiation originates from the exhaust plume and from aerodynamic heating. While these aspects of target detection may favor the use of infrared sensors, these (infrared) systems are not without weaknesses in the region of interest: solar glint off wave surfaces produces clutter returns, and atmospheric conditions near the surface of the water induce refraction changes and variation of the extinction coefficient.

The notional sensors considered for the purposes of this research include an azimuth scanning infrared system operating in the 3-5 μm region, and an X-band radar scanning in azimuth with a pencil beam. The proposed characteristics of the target include an IR signature determined by a black body model for aerodynamic heating, and for the radar, Swerling case 2 fluctuations in the radar cross section.

Figure 1.1 gives a graphical description of the radar/infrared detection problem for the sensors in question.

The timely detection of low-elevation targets at the horizon requires the greatest feasible detection ranges from the sensors. Increasing the detection range requires that thresholds in the sensor receivers be lowered. However, the result is an increased number of false alarms from both sensors, caused in part by horizon clutter (this presents a problem, because the increased number of false alarms may force the ship to waste valuable defensive weaponry).

It is believed that the nature of the clutter at the horizon is fundamentally different for radar and infrared systems, and that their clutter processes are uncorrelated[1] at the horizon (because the physical processes

[1] That is, while there may be clutter correlation that occurs for data obtained from a single sensor, there should be none for data obtained from these two completely different sensors. The exception to this is sea birds, which may produce correlated clutter. However, returns from sea birds can be reduced in the radar by the use of sensitivity time control (STC).

producing the clutter are different). Thus, an infrared sensor could serve as a capable adjunct to a radar system.

In addition, research performed by Reibman and Nolte [2] and Barkat and Varshney [20] indicates that the fusion of sensor detections can result in improved system false alarm performance. This is especially useful to reduce the high probability of false alarm that arises frequently in environments such as the area around the horizon.

Once a target has been detected, the timely presentation of a high-resolution three-dimensional (3-D) track (which is a record of an object detected in the field of view) to a fire control[2] system is also important. Ideally, any sequence of detections that initiates a track should consist of both infrared (which provides high angular accuracy, and measurements in azimuth and elevation) and radar (which provides measurements in azimuth, range rate and range only[3]) detections[4]. Thus, high and low-resolution two-dimensional (2-D) information must be fused, as shown in Figure 1.2.

---

[2] A fire control system is utilized to destroy targets, it is not men with buckets putting out a fire (hopefully).

[3] Elevation data are not utilized because of the detrimental effect of multipath on the radar elevation measurement.

[4] This may not always be possible. For instance, if the detection ranges of the sensors differ widely, data from both sensors will not be available. In this case, the result will be a two-dimensional (2-D) track initiation, which is still valuable.

The research in this dissertation focuses on derivation and use of the multisensor track initiation (MSTI) technique to initiate tracks on low-elevation targets by integrating data from a low-resolution scanning radar with data from a high-resolution scanning infrared system. These data are in the form of individual sensor target detections, obtained with the sensors operating at high false alarm rates.

The MSTI technique itself can be summarized as follows: Time-ordered detections consisting of both valid target data and false alarms are presented to the algorithm. The algorithm sequentially examines each new detection that arrives and composes a data sequence that is made up of the current new detection and any other detections that roughly correlate with it in spatial coordinates and time. This data sequence accounts for the missing measurement phenomena (*i.e.*, when sensors are unable to provide detections[5]) by indicating in what order and at what time the detections arrived from the sensors. A set of spatial feature tests[6] is then performed on this data sequence, and the results used to compute the log-likelihood ratio for the data sequence. The log-likelihood ratio is a measure of how "target-like" a data sequence is. The log-likelihood ratio is then compared to two thresholds. If the

---

[5] This may be due to a degraded sensor, or environmental factors such as: clutter, glint, refractive effects or weather conditions.

[6] A feature is a measured or calculated value for discriminating between the target and false alarms.

log-likelihood ratio passes the highest threshold, a target is declared and a track initiated. If it passes the lowest threshold but not the highest, an additional sensor measurement is requested. If the log-likelihood ratio does not pass either threshold, no action is taken. The formation of data sequences, the performance of spatial tests and the threshold comparisons continue as long as the algorithm is presented with new detections. Thus, the dissertation addresses the formation of data sequences (and their statistical characterization), the derivation of appropriate feature tests and the determination of proper log-likelihood ratio thresholds.

The remainder of this chapter discusses how the sensor detections are obtained, and the background behind the MSTI technique.

1.1 Search and Track System Architecture

Figure 1.3 illustrates the detection process as implemented in many current single-sensor search and track systems. Block A designates the operational components of a complete search and track system. Block B comprises the functional components present in the sensor receiver. As can be seen from Figure 1.1, a target reflects or emits some type of energy in the electromagnetic (or acoustic) spectrum. This target energy, combined with energy reflected or emitted from clutter, is picked up by the sensor as shown in Figure 1.3, which then converts it into an electrical signal. The signal

processor then performs various operations on the signal to discriminate between background and target returns. However, the discrimination process is not always successful.

Any target or clutter returns that pass a detection criterion, usually in the form of some threshold applied to the signal after preprocessing in the receiver, are passed on to a tracker/correlator, which is responsible for initiating tracks, correlating new detections with existing tracks, and operating Kalman filters on existing tracks. Figure 1.4 illustrates the functional blocks of the tracker/correlator.

Many techniques exist for **correlating detections to existing tracks**. Bar-Shalom [3] provides a comprehensive survey, including nearest neighbor-techniques. In this approach, a distance measure is computed (sometimes the Mahalanobis distance) that compares a predicted track with a set of new detections and then determines which, if any, of the detections belong to the old track by selecting the detection with the minimum distance measure. Bar-Shalom [3] discusses another popular technique, the Probabilistic Data Association Filter (PDAF). In this approach, a distance measure is computed for each detection, and all detections having a distance measure below a threshold are considered to be associated with the track. The set of detections associated with a particular track is then used to update the

track filters, with each detection making a weighted contribution to the track (contingent on the perceived reliability of the detection).

Various methods exist for **initiating tracks**, and good overviews of these techniques can be found in Bar-Shalom [3] and in Blackman [4]. By far the most popular technique in use today is the "logic-based" approach, which involves setting up gates (*i.e.*, spatial and temporal regions) around any unpaired target detections (*i.e.*, those detections that are not paired with a track after track correlation); detections falling within a gated region are assumed to be associated. If k detections out of the next N scans fall within this gate, then a new track is initiated. In its basic form, the technique is simple, not processing intensive, and easy to implement. Variations of the technique may permit different numbers of track "dropouts", or missing detections, at different points in the initiation process. It is believed that, with the exception of this research, variations of the logic-based approach have not been proposed for multiple sensor systems.

Multiple-hypothesis track initiators utilize assumed *a priori* target and noise densities as well as assumed system detection and false alarm probabilities to determine the probability of the new measurement 1) being associated with a previous track, 2) corresponding to a new target, and 3) being a false alarm. The hypothesis with the highest probability is then

chosen. Unfortunately, the *a priori* target and noise densities are usually not known in practical situations.

Maximum-likelihood methods have been applied to target detection, track initiation, and association problems. Trunk and Wilson [5] have discussed the application of a maximum-likelihood method to track initiation for unresolved targets, in which multiple targets fall within the same resolution cell yet give rise to only one detection. After applying a linear fit to each set of candidate detections, the researchers tested the goodness of fit for each component detection of the candidate track and used the goodness of fit as a discriminant in the initiation process.

Maximum-likelihood detection methods loosely based on the Viterbi [6] algorithm have also been discussed in the literature [7],[8]. The Viterbi algorithm itself is based on dynamic programming techniques set forth by Bellman [9]. The basic principle behind both the dynamic programming and the Viterbi algorithms is that small portions of the detection problem can be solved as part of a larger one. Arnold [8] has applied this technique to the detection of targets by use of a moving target indication (MTI) radar. The algorithm uses path length functions for azimuth, range, range rate, and signal energy. Unfortunately, some of the assumptions made render the algorithm unsuitable for multisensor applications near the horizon: Signal energy and path smoothness were used as features for computing the path length.

Multipath near the water surface negates the usefulness of this these features by creating large errors in elevation measurements, as well as nulls in the SNR response. In addition, no provisions for adjusting the total path length threshold were made to account for high false alarm rate data or missing measurements. Finally, Barniv [7] has discussed the application of dynamic programming techniques to the detection of targets for a synthetic aperture radar (SAR) system staring at a fixed point on the ground. He used the approach to detect extremely dim (SNR<1) moving targets within the field of view.

While the above algorithms [7],[8] differ in their use of dynamic programming techniques, they do provide a good idea of how the method applies to the single sensor detection problem. The algorithm in its basic form operates as follows: First a candidate set of time-ordered detections is obtained. The path length or log-likelihood ratio $L_{i,i+1}$ is then computed for each pair of candidate associated detections at time t_i and t_{i+1}:

$$L_{i,i+1}=\log\left(\frac{\Pr(z|Tgt)}{\Pr(z|Clutter)}\right) \qquad (1\text{-}1)$$

where z is a set of feature measurements synthesized from raw measurements made at t_{i+1}, and t_i (this may include azimuth, elevation, range, range rate, and SNR). The pair with the highest log-likelihood is taken to be the best association. The algorithm then compares detections at time t_{i+1} and t_{i+2}. This

continues until some stopping criterion is satisfied (say, over a fixed time window). If N is the number of detections, then the total path length L_p or log-likelihood ratio is defined as:

$$L_p = \sum_{i=1}^{N-1} L_{i,i+1} \tag{1-2}$$

L_p is then compared to a threshold. If it passes the threshold, a target is declared. However, these techniques do not address the problem of missing measurements, as the MSTI approach does.

1.2 Information Fusion in the Search and Track System

The addition of multisensor data to the system complicates the entire search and track process because measurements taken by one sensor may have no analog when compared to a second sensor (*i.e.*, the radar can make a range and range-rate measurement, whereas the infrared sensor cannot). If one assumes that difficulties such as these can be resolved, then there are basically three ways in which the multisensor data can be combined in a search and track system.

The first approach fuses the actual sensor observations before they undergo any signal processing, as shown in Figure 1.5. The second approach

fuses the sensor detections[7], as detailed in Figure 1.6. The third approach fuses the sensor tracks, as shown in Figure 1.7 (note: the contents of the boxes A, B and C correspond to that in Figure 1.3).

Although the first approach, fusion of sensor observations, minimizes the loss of information, it is virtually impossible to retrofit to existing sensor systems. This method, discussed by Van Trees [10], requires enormous communication bandwidths as well (in cases where the sensors are not collocated) and is therefore generally considered impractical. The third approach, fusion of sensor tracks, while being the easiest to implement and to retrofit, does not provide any data fusion advantage until *after* the target has been declared and is being tracked.

The fusion of sensor detections, however, permits detection gain from the different sensors while eliminating a substantial amount of redundancy in the track processor. Furthermore, it has the advantage of being reasonably easy to retrofit to existing systems. Scott [11] has graphically analyzed some of the tradeoffs involved in using sensor detections instead of observations. He found that under certain conditions (such as both hypotheses equally likely, two independent and identically distributed Gaussian measurements, and a

---

[7]It is important to note that the target has not been declared (hypothesis H_1 true) by the multisensor system; *i.e.*, individual sensors transmit detections (including a large number of false alarms) to a fusion center, which is then responsible for declaring targets.

fused decision made according to the minimum probability of error criterion), fusion using the sensor detections produced a probability of error equal to that of the individual sensors. For the search and track problem in question, these circumstances almost never occur. He also showed that under all circumstances, fusion of sensor detections could not provide performance equal to that of fusing sensor observations for the two-measurement case. This was an effect of the detection process, which is simply a binary quantization of the input measurement.

The central assumption in all of the analyses performed by Scott is that the joint sensor measurement probability density function (PDF) is known, which is not necessarily the case in a multisensor system. For example, in a radar system, the distributions of clutter returns cannot always be characterized exactly. Furthermore, even when the PDF is known for all of the measurements, the use of the entire PDF in forming a discriminant boundary increases the sensitivity of the fusion process to small errors in the PDF characterization. Conversely, the fusion of detection information makes weaker assumptions (that is, that the probabilities of detection and false alarm of a test are known for individual sensor data) when forming a discriminant boundary. The fusion of sensor detections uses assumptions that are similar to that for the family of robust detectors known as *sign detectors*. The *sign detector* makes a detection decision based only on which side of the median

measurements lie. The assumption for fusing sensor decisions is similar, in that declaration decisions are based only on whether measurements lie on either side of a threshold. For an overview of robust detection methods, see Gibson and Melsa [12].

The present research concentrates on fusion of sensor detections, because of its relative ease of implementation and because it makes weaker assumptions regarding the measurement statistics.

1.3 Detection Fusion Methods

The detection fusion methods discussed in the literature can be coarsely grouped into four different categories:

1. Thresholds fixed *a priori* at the sensors and optimized at the fusion processor (a central decision making function that bases its decisions on peripheral sensor decisions);
2. Thresholds optimized at the sensors and fixed *a priori* at the fusion center;
3. Thresholds at the fusion center and at the sensors optimized concurrently; and
4. Methods based on the sequential probability ratio test (SPRT), as proposed by Wald [13].

For the methods enumerated above, the typical detection fusion system is organized as shown in Figure 1.8. For category 4 techniques, thresholds at the sensor and at the fusion center vary as new measurements are presented to the sensors (not explicitly shown in Figure 1.8). The sensors make a detection u_i based on an observation y_i. The detection/decision rules at the sensors and at the fusion center are in the form of likelihood ratio tests, which are made with a probability of detection (P_{Di}) and a probability of false alarm (P_{Fi}).

The fusion center then makes a decision to declare based on a log-likelihood ratio test. It does this by comparing a weighted sum of the peripheral sensor decisions (u_i) to a threshold; Figure 1.9 depicts this process at the fusion center. If a minimum probability of error criterion is assumed, the weights (ω_i) are determined by the following equations (from Chair and Varshney [14]):

$$\omega_0 = \log \frac{P_1}{P_0} \qquad \textbf{(1-3)}$$

where the *a priori* probability of hypothesis H_i occurring is P_i, and

$$\omega_i = \begin{cases} \log\dfrac{1-P_{Mi}}{P_{Fi}}, & \text{if } u_i = +1 \\ \log\dfrac{1-P_{Fi}}{P_{Mi}}, & \text{if } u_i = -1 \end{cases} \qquad \textbf{(1-4)}$$

P_{Mi} is the probability of miss associated with the decision i, and P_{Fi} is the associated probability of false alarm. N in Figure 1.8 and Figure 1.9 represents the number of sensors, and Λ in Figure 1.9 is the log-likelihood ratio at the fusion center. The likelihood ratio, which in this case is a weighted sum of the sensor decisions, can be expressed as follows:

$$\begin{aligned} \Lambda &= \log\frac{P(u|H_1)}{P(u|H_0)} \\ &= \sum_{i=1}^{N} \omega_i u_i \end{aligned} \qquad \textbf{(1-5)}$$

Where **u** is the vector of sensor decisions $[u_1, u_2, ..., u_N]$. A decision to declare at the fusion center is then made by use of the following inequality:

$\Lambda > \omega_0$ Declare H_1

$\Lambda \leq \omega_0$ Declare H_0

As outlined by Chair and Varshney [14] and as illustrated by the log-likelihood ratio test, the *a priori* probabilities of the hypotheses H_0 and H_1 are necessary for the calculation of the fusion rule. However, this represents a shortcoming for many practical systems, in which the exact values of P_0 and

P_1 are not known; the latter has lead to methods [15] in which the Neyman Pearson criterion is employed for calculating the ω_0 term.

Several papers have been published that address techniques belonging to category 1. In 1980, Nahin and Pokoski [16] discussed general detection fusion. Chair and Varshney [14] later addressed optimum detection fusion by formulating a log-likelihood ratio test that utilized a weighted sum of the individual sensor detections. These weights were based on the operating characteristics of the sensors. Viswanathan and Aalo [17], [18] analyzed the asymptotic performance (as $N \rightarrow \infty$) of the "k-out-of-N" rule for sensors with identical thresholds and measurement statistics for the case of correlated and uncorrelated local decisions. Thomopoulos *et al.* [15] analyzed the distribution of the log-likelihood ratio for the case of identical sensors. They showed that the decision rule at the fusion center for this case could be generalized to a "k-out-of-N" rule.

Category 2 techniques fix thresholds at the fusion center but optimize them at the sensors, as discussed by Tenney and Sandell [19]. Barkat and Varshney [20] also discussed the generalized "k-out-of-N" decision rule as it applied to a multiple radar constant false alarm rate problem; they analyzed cases in which the decision rules at the fusion center were either AND or OR and optimized the thresholds at the sensors. Thomopoulos *et al.* [21] proved that if the decision rule at the fusion center followed the Neyman-Pearson

criterion, then the optimal thresholds at the sensors were based on the likelihood-ratio test. Srinivasan [22] had come to the same conclusion earlier but used a different approach based on the method of Lagrange multipliers. In addition, he analyzed the performance of the distributed system for the AND, OR, and majority logic cases.

Category 3 techniques optimize thresholds at the fusion center and at the peripheral sensors concurrently. Reibman and Nolte [23] illustrate that this approach to the detection fusion problem involves the simultaneous solution of a set of coupled, non-linear equations. The solution of these equations is difficult, and the authors made several simplifying assumptions to present an example of the technique[8]. Izzo and Paura [24] later called attention to these issues in their correspondence.

In the final category, the sequential probability ratio test, as first discussed by Wald [13], is employed. This test provides a convenient way to include multiple measurements as a way to resolve sensor ambiguities. Using this method, future decisions are based on the results of past decisions. Teneketzis [25] was one of the first to apply this method to the multisensor detection problem, although he quickly determined that the solution was mathematically intractable. Chair, Hoballah, and Varshney [26] formulated

---

[8] The example presented assumed identical noise statistics and identical thresholds at each sensor. In doing this, they arrived at the same results as Chair and Varshney [14].

a method based on the concept of bilevel thresholds, in which the likelihood ratio is tested against two thresholds. If the likelihood ratio is higher than the first threshold, the fusion center declares H_1. If the likelihood ratio is higher than the second but below the first, another set of measurements is taken. If the likelihood ratio is below the second threshold, no more measurements are taken, and H_0 is declared. This method requires that two thresholds be computed at the fusion center as well as one at each sensor for each new measurement that arrives. The addition of Neyman-Pearson constraints to the system results in an optimization problem involving a set of coupled nonlinear equations.

The combination of key concepts from the sequential probability ratio test and dynamic programming techniques shall be detailed in Chapter 2, where the MSTI algorithm will be discussed in more detail.

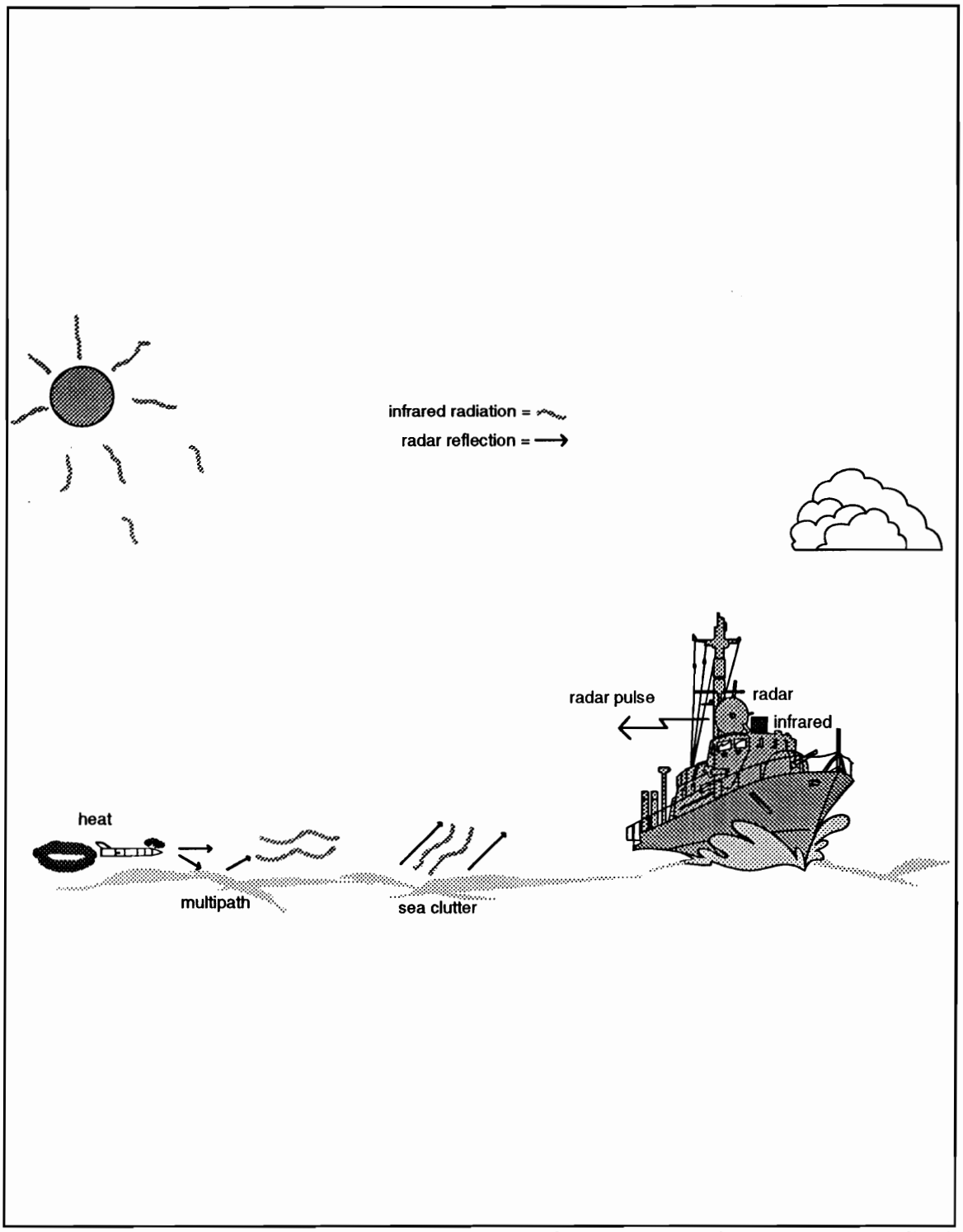


Figure 1.1 Graphical illustration of the multisensor track initiation (MSTI) problem

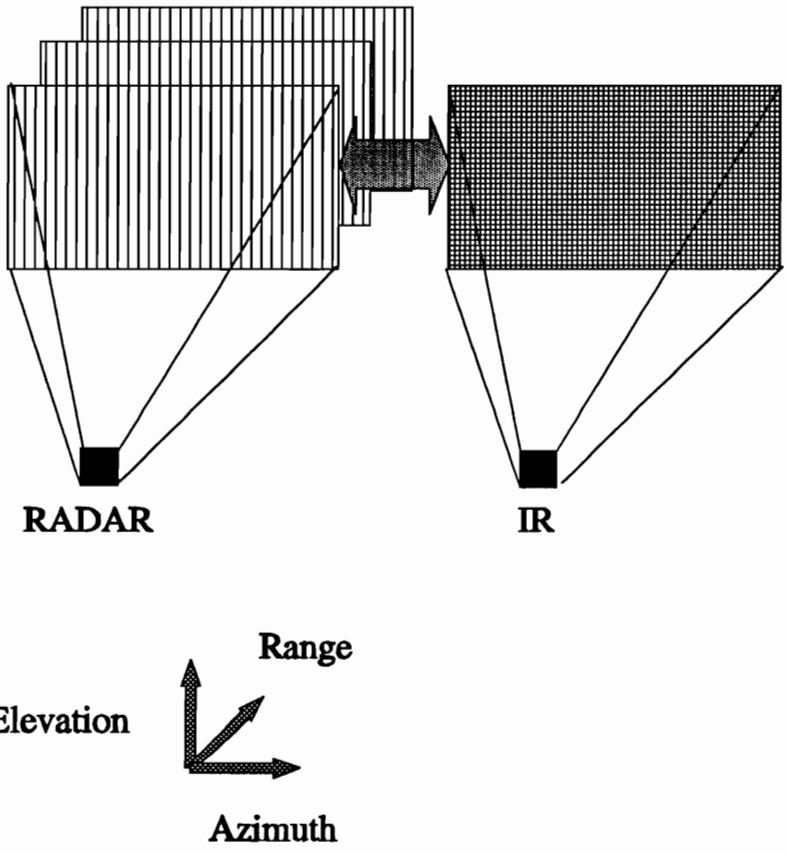


Figure 1.2 Infrared/Radar fusion problem

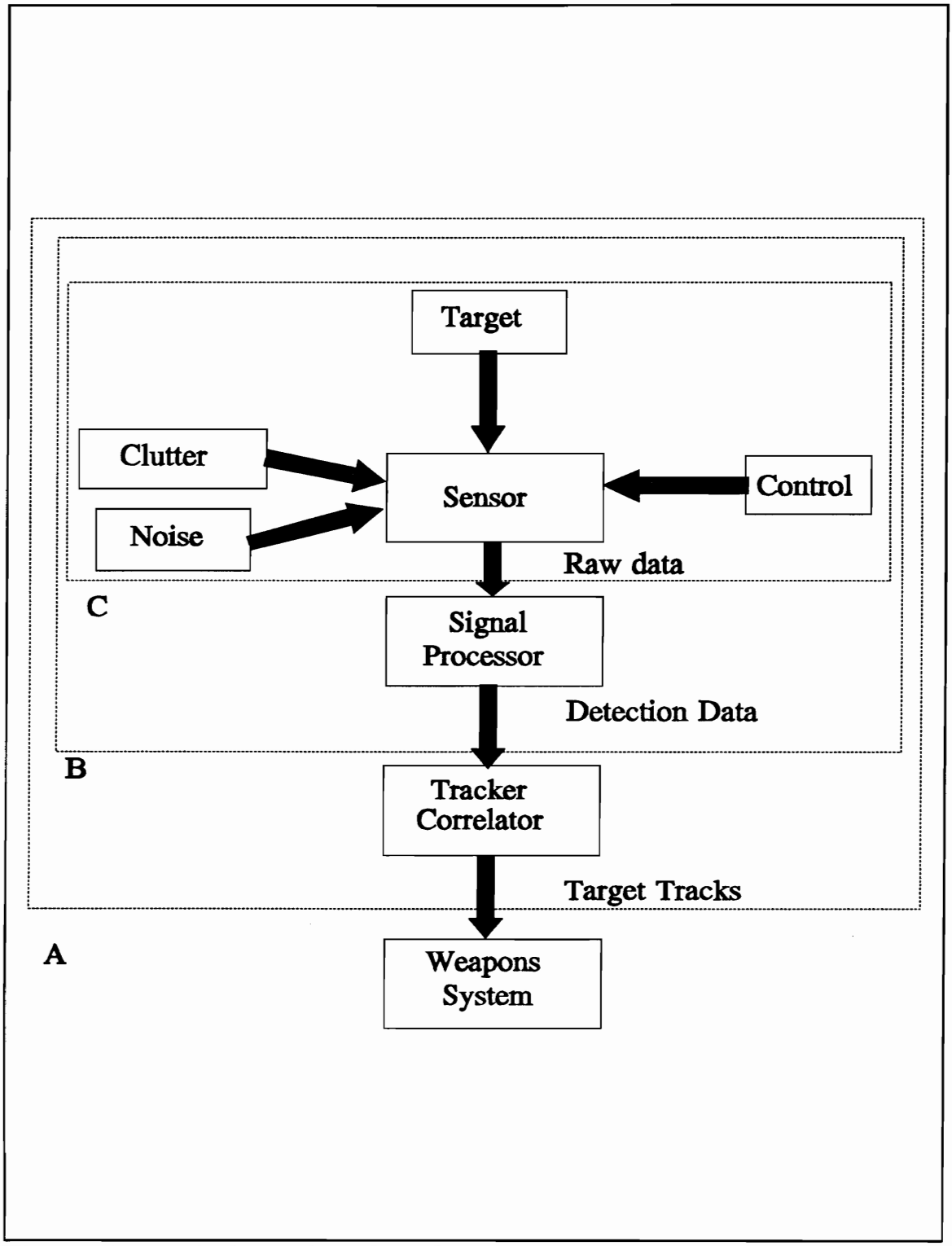


Figure 1.3 Typical search and track system

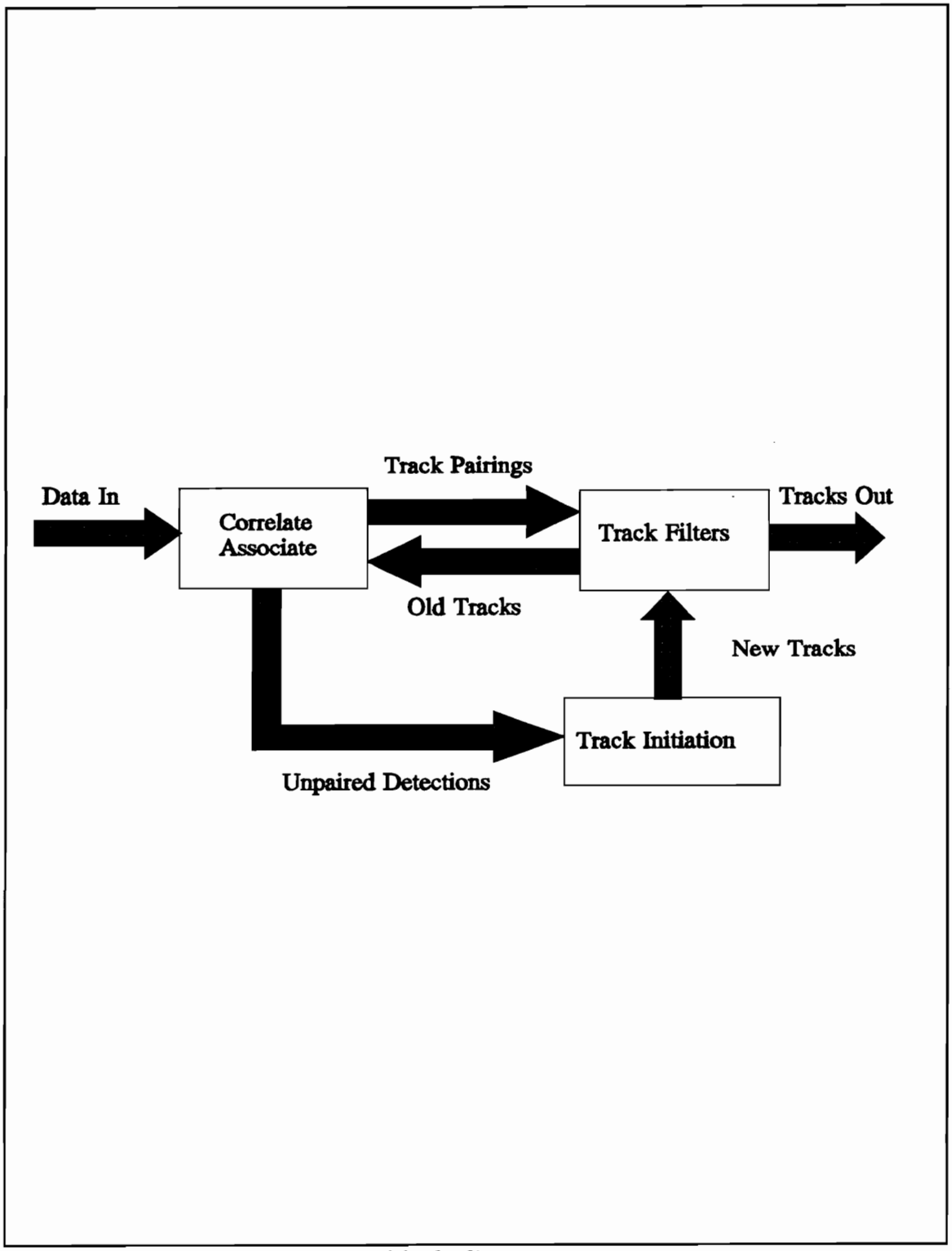


Figure 1.4 Tracker/Correlator block diagram

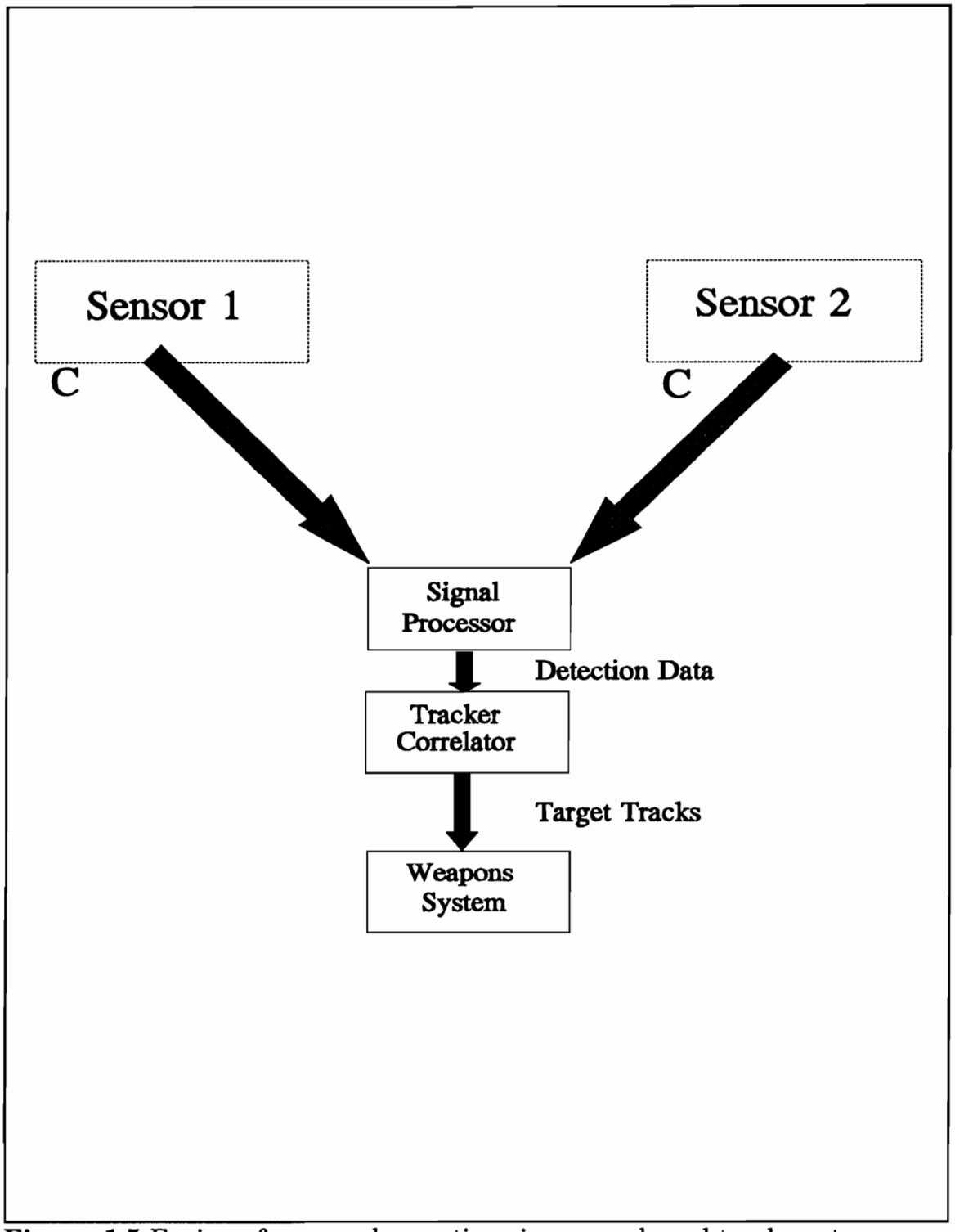


Figure 1.5 Fusion of sensor observations in a search and track system

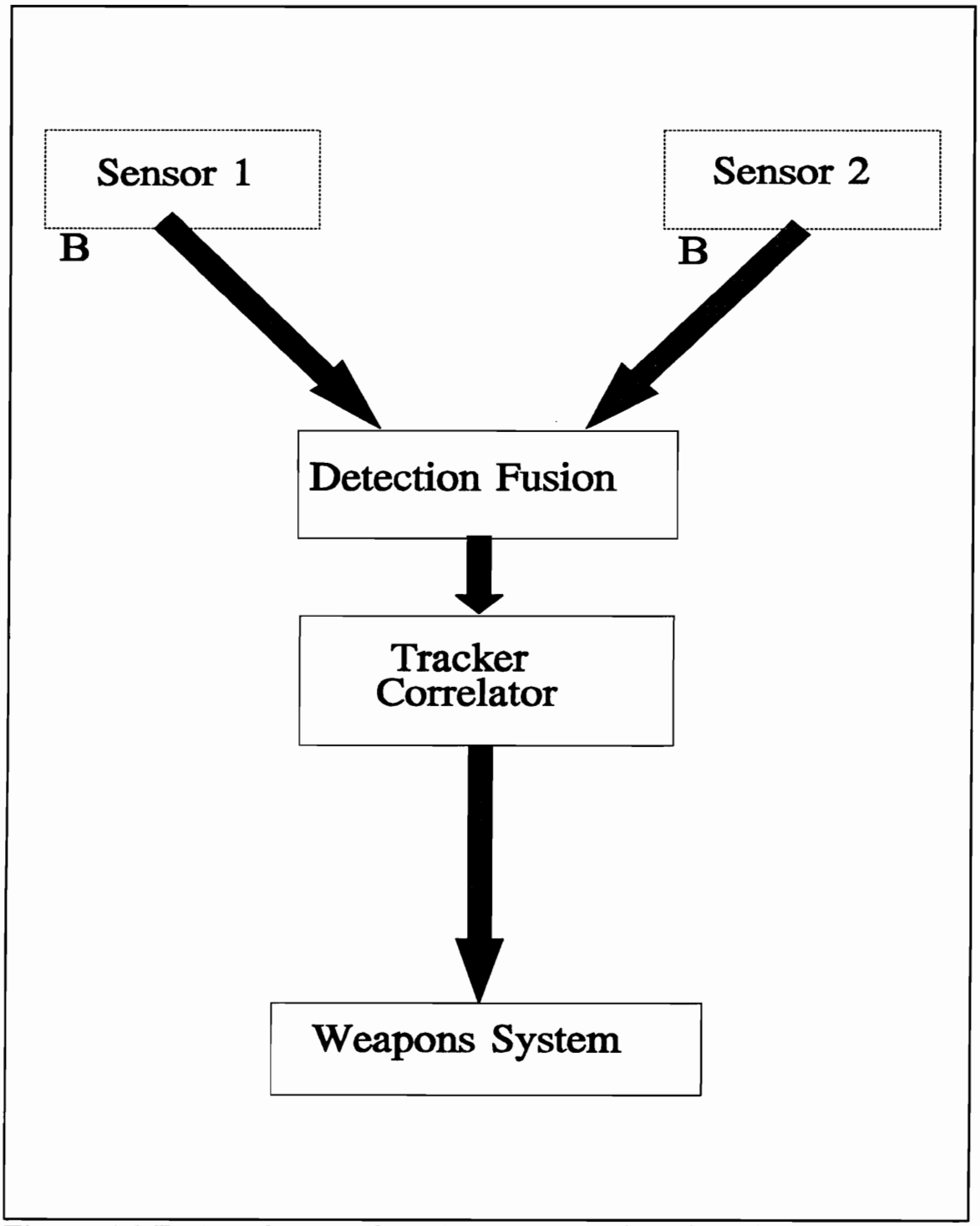


Figure 1.6 Fusion of sensor detections in a search and track system

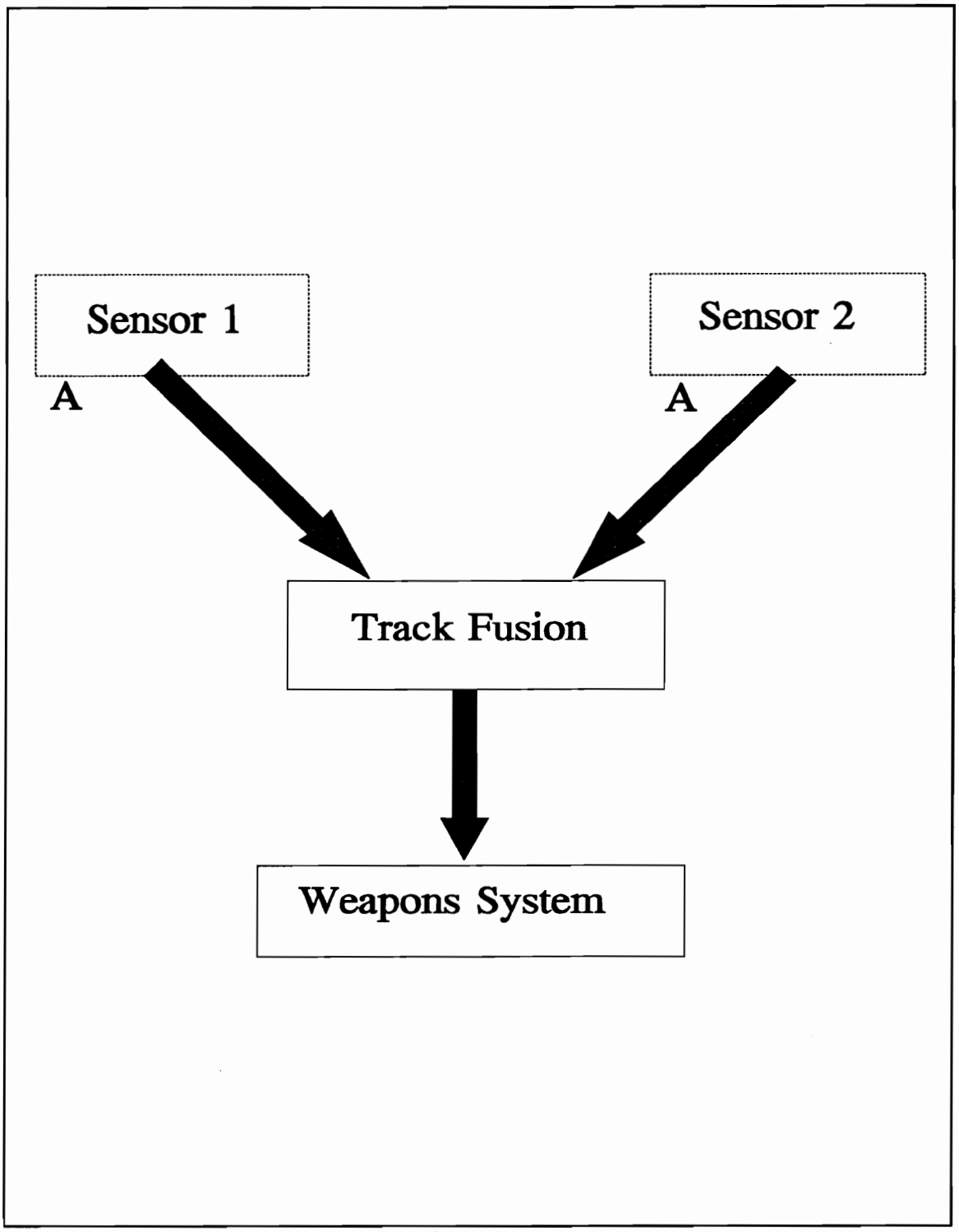
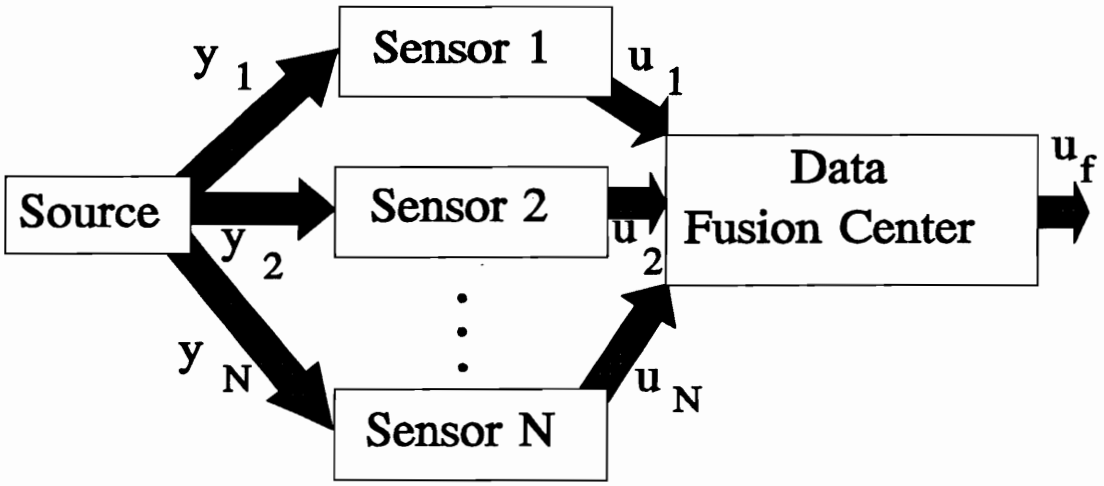


Figure 1.7 Track fusion in a search and track system



y_i = sensor measurements

u_i = sensor decisions/detections

u_f = fusion center decision

Figure 1.8 Typical post-detection fusion system

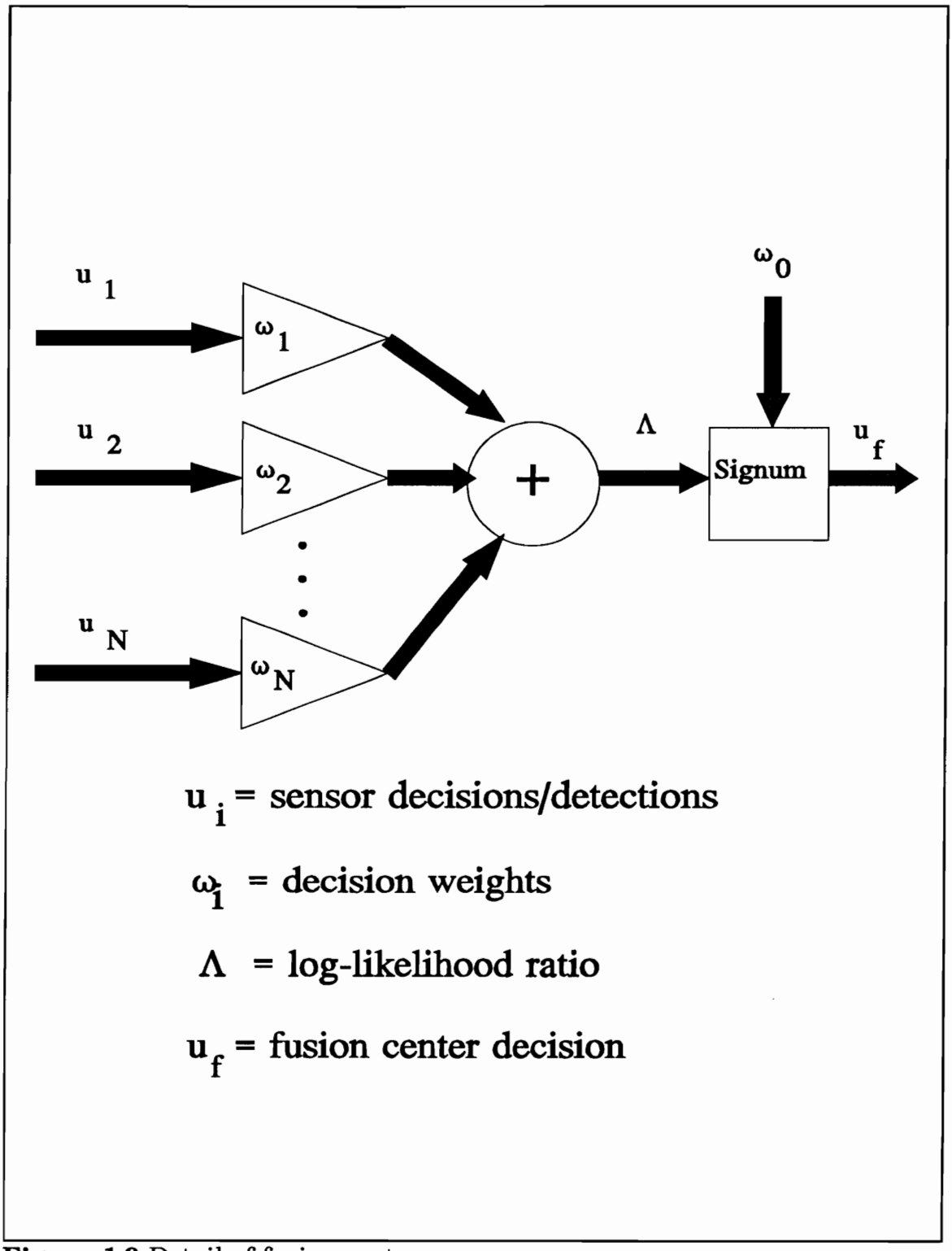


Figure 1.9 Detail of fusion center

Chapter 2: The Multisensor Track Initiation Technique

2.1 Overview

The multisensor track initiation technique (MSTI) method combines important concepts from the theories of maximum-likelihood track initiation and detection fusion.

From Arnold [8] it is seen that the principle of optimality and the total path length are important concepts when applied to the track initiation problem. In equation (1-2), the total path length was defined as the sum of the intermediate path lengths for a particular data sequence (with a data sequence being a time ordered list of multisensor detections). These path lengths are computed via features that determine how target-like a particular data sequence is. However, three problems impair the direct application of maximum likelihood techniques. First, using a maximum likelihood criterion for the selection of the best data sequence is not practical in a high false alarm rate environment due to processing constraints. Second, there is no mechanism for determining the optimum threshold for the log-likelihood ratio when there are missing data (*i.e.*, when data sequences of differing lengths must be compared). This is not a factor when considering the synthetic aperture radar

system [7], but it is when considering the problem at hand[9]. Third, depending on the strength of the feature tests used, the maximum likelihood criterion may result in the exclusion of those data sequences that contain multi-sensor data. For instance, take the case where we are trying to test whether a radar and an infrared data pair are indicative of the same target being present. The only measurement these data have in common is azimuth (see Figure 1.2). The radar has relatively low angular resolution (see appendix C), whereas the infrared sensor has high angular resolution. A possible test for determining whether these two data points originated from a target could be a threshold on their angular difference. If the difference is small, and the data arrive within a short time interval, we expect that they did indeed originate from a target. However, the larger angular resolution of the radar places an upper limit on the probability of false alarm (P_F) of the test. Now suppose that we wish to test two infrared data points. We expect that the difference will be even smaller, because of the increased accuracy of the measurements. Thus the P_F for this test will be lower than that for the radar and infrared data pair. If the P_F desired at the fusion center is very low, we may end up excluding those data sequences containing radar/infrared data pairs.

---

[9] In the problem being considered, the radar has a possibly asynchronous scan rate, and only detections are transmitted to a fusion center. This means that there may or may not be data present at a particular azimuth or range. This cannot be predicted, thus decisions must be made using only the available data.

Given that these problems are significant, the concepts of path length and optimality can be applied to the problem at hand with the following enhancements:

1. A data sequence is formed by examining all the data within a spatio-temporal gate of the most recent data point originating from either sensor. This is done as follows:

The fusion processor (a device which implements the MSTI algorithm) receives data from both the infrared sensor and the radar concurrently, which are scanning in azimuth. Assume that the data point received at time T and spatial location S is being examined. The fusion processor employs a "scan back" approach, as shown in Figure 2.1, in which it examines all data within a spatial and temporal region of T and S; this approach is often referred to as spatio-temporal gating. The gating procedure is used because we do not wish to examine the likelihood of associations for which it is known *a priori* that these associations will have a likelihood approaching zero.

As a result of the spatial and temporal gating, the fusion processor limits its input to the set of data composed of all the detections falling within the gated region. It then filters the data using an order statistic filter of order s based on the signal to noise ratio (SNR) of each data point[10]. For the case

[10] This may require dynamic scaling of the SNR. The order statistic filter is used because of its robust nature, and because the true conditional probability density functions of the SNR are in general unknown.

shown in Figure 2.1, the data sequence {A,B,C,D} would pass the spatiotemporal gating criterion. If s=3, then the three highest SNR data points would be extracted from {A,B,C,D}; let us assume that these are {A,B,D}.

2. Feature tests similar to those used by Arnold [8] are then applied to the data. However, these feature tests are in fact likelihood-ratio tests[11] and are based on a Markov assumption, in keeping with the principle of optimality. The Markov assumption is convenient in that it allows us to form the log-likelihood ratio by using feature tests based on sequentially occurring data pairs. Discrete tests are used for increased speed and better sensitivity characteristics. This is illustrated as follows: consider some high false alarm rate data. Every new data point that is presented to the fusion processor must have a set of feature tests performed on it. The feature tests suggested by Arnold [8] use transcendentals and possibly some divisions. A feature test requiring only a subtraction and a comparison would be much faster. Chapter 3 discusses the derivation of feature tests having these characteristics.

The log-likelihood ratio computation is now based on the feature tests. The fusion processor computes the log-likelihood ratio for the data sequence ({A,B,D}) as follows:

---

[11] Arnold [8] uses a classical log-likelihood ratio computation, whereas the MSTI method uses detection fusion methods for computation of the log-likelihood ratio.

$$\Lambda=\sum_{i=1}^{N} \omega_i u_i \quad , \; u_i \in \{-1,0,1\} \tag{2-1}$$

Equation (2-1) is similar to equation (1-5), except here u_i refers to the i^{th} out of N feature tests computed for the data sequence rather than the i^{th} sensor decision. The weights ω_i are computed as in equation (1-4). Note that the domain of u_i now includes 0, which is used to denote the absence of a particular feature. Also note that the ω_0 term is absent; for purposes of the log-likelihood ratio test, ω_0 is computed on the basis of the Neyman-Pearson criterion.

3. Two thresholds are applied to the log-likelihood ratio instead of one, in a manner similar to that of the sequential probability ratio test. A target is declared, and a track initiated, if the log-likelihood ratio exceeds the first threshold. A cue (i.e. a request for a second set of sensor measurements) is requested if the second threshold, but not the first is exceeded. No action is taken if neither threshold is exceeded. Both thresholds are determined via Neyman-Pearson constraints. This is done because typically, search and track sensors are limited in the number of cues that they can respond to in a given period of time due to minimum search frame requirements. In addition, it is necessary to consider other system level requirements[12] when performing the

[12] Such as the fact that a track initiation using 3-D data is more valuable than one that originates from 2-D data only.

threshold optimization under Neyman-Pearson constraints. The use of these constraints in the optimization problem is discussed in Chapter 5.

To carry out this optimization, it is necessary to determine the probability density function (PDF) of the log-likelihood ratio, which is covered in Chapter 4. Special techniques are needed to compute the log-likelihood ratio and the PDF of the log-likelihood ratio due to the discrete nature of the feature tests. These are discussed in Appendices B and D. In addition, the relative frequency of occurrence of the data sequences addresses the missing measurement/feature test problem and is necessary for the computation of the optimal log-likelihood ratio thresholds, which is discussed in Chapter 5.

The above three-step formulation allows the method to operate with the speed of the k-out-of-N techniques discussed in Section 1.1, while adding the ability to deal with features for which the PDF is incompletely defined, and to deal with dissimilar sensors. Indeed, if all the feature tests were identical across the sensors, then the best solution to the threshold optimization would be some k-out-of-N criterion.

2.2 Maximum Sequence Length and Gate Dimensions

As was discussed in step 1 of the MSTI process detailed in the previous section, the maximum sequence length or the maximum size of any set of multisensor data group is s. Processing and memory constraints fix a

maximum value for s. Given s, we wish to determine the extent of the spatiotemporal window. However, a true optimization of s would be extremely lengthy. This is not practical; therefore the following scheme is proposed:

1. Fix the window time length so that the sensor operating with the slowest update rate can provide s pieces of data, assuming no missing measurements. This value is based on the maximum scan time of the sensors in question (in this case the radar, which has a scan time of 2 seconds). Therefore, if s=3, then the extent of the sliding time window would be 6 seconds. Most data fusion algorithms are based on some k-out-of-N criterion, and this should allow the algorithm to attain robust performance in cases for which there may be missing measurements.

2. Once the value of the time window is fixed, it can be used to determine the spatial extent of the window using the 4σ points obtained from the target maneuver model[13]. For the sensors in question, only an azimuth window was used (because of the small field of view (FOV) in elevation), and its total extent was determined to be 2.56°. Note that in some cases it may not be possible to use the 4σ criterion (due to clutter densities or target maneuver parameters). For example, if the window size calculated using the 4σ points is extremely large, then multiple targets may not be declared. In that case, the extent of the

[13] For a gaussian model, this allows the gate to pass nearly 100% of the target data.

time window should be modified, resulting in a smaller spatial window. This, however, could decrease the performance of the method, because valid detections may not be passed by the window.

Note that in some cases, the maximum sequence length will also be driven by the desired false alarm rate. As the sequence length increases, so does the number of feature tests. Consequently, a lower false alarm rate can be attained.

Figure 2.2 provides an illustration of the considerations involved in determining the window parameters.

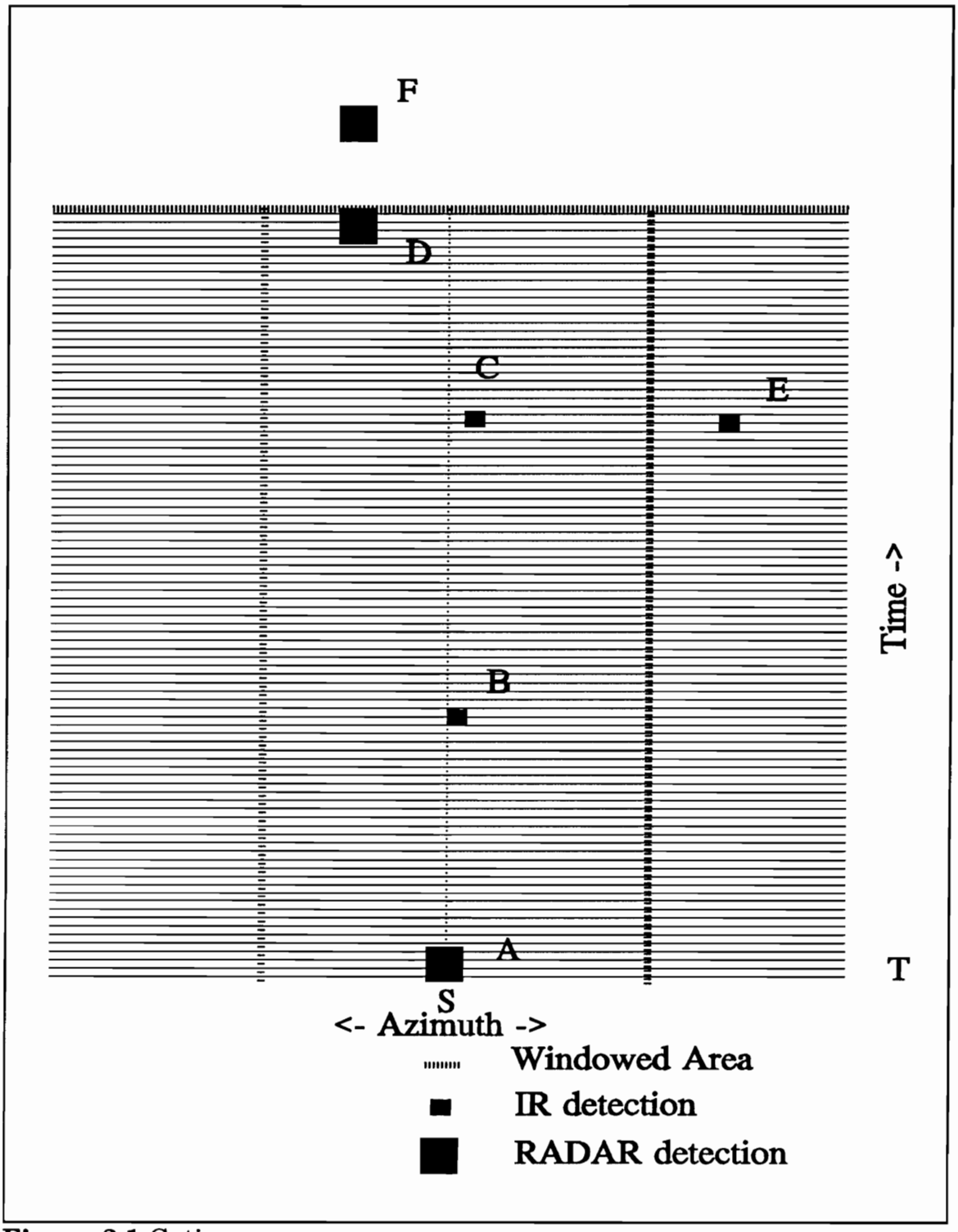


Figure 2.1 Gating process

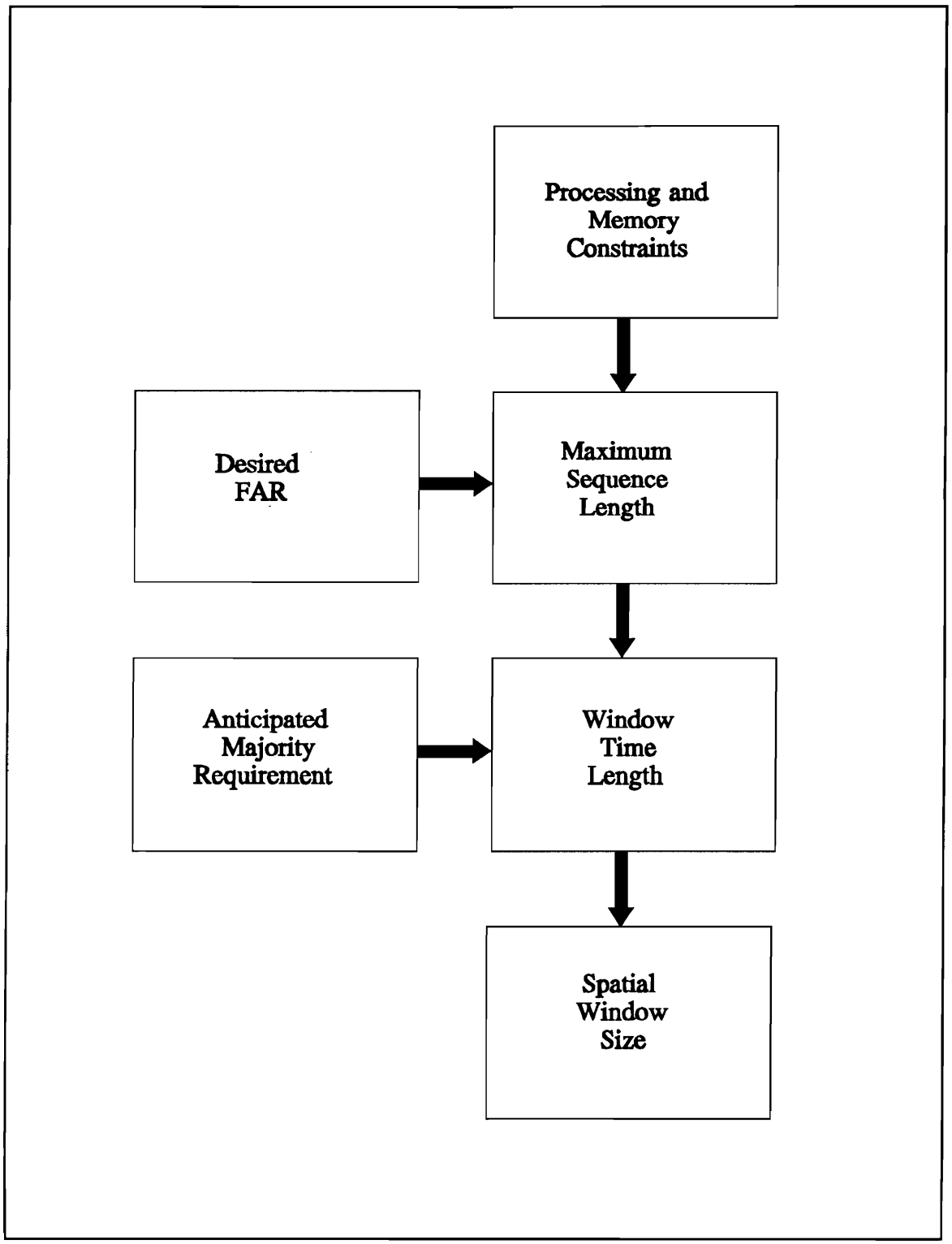


Figure 2.2 Diagram of threshold determination methodology

Chapter 3: Feature Tests for the Track Initiation Algorithm

3.1 Assumptions for Clutter and Target

The following sections address the derivation of the probability density functions (PDFs) of the feature tests used for discrimination purposes. Some derivations are performed using dummy random variables; substitutions are then made at the end of the derivations. This was done because of the general nature of the results (*i.e.*, the general form of the azimuth and elevation feature tests is the same, with the only variation being the specific parameters used).

Each feature test is based on a likelihood-ratio test that uses a minimum probability of error (MPE) criterion. The likelihood ratio test approach was proven optimal [21] for the distributed detection problem.

The radar, and to a certain extent the infrared[14], sensors will be clutter limited in their operation. This means that the performance of the sensors will not necessarily be limited by electronic noise but instead will be limited by spurious detections (or false alarms) that arise from other objects in the field

[14] For high-speed targets that appear just above the horizon, the infrared background will be benign (in most instances), and sensor noise will be the limiting factor in detection of the target. For low-speed targets, the infrared background will not necessarily be benign, and clutter returns may result. In either case, some form of constant false alarm rate and/or non-uniformity correction can be employed, so that approximately uniform spatial noise results.

of view, such as sea birds, and waves. Typically, clutter and, at times, sensor noise for the two sensors operating in the region of interest are spatially non-uniform and non-stationary. Constant false alarm rate devices have been developed to address this problem [27] [28] [29]. Barnett [28] has suggested median subtraction filtering to control spatial noise in an infrared sensor; Hansen [29] has addressed the use of a constant false alarm rate device to reduce radar clutter false alarms in non-uniform noise. Figure 3.1 illustrates the performance of a typical constant false alarm rate device when confronted with clutter edges. Note that this type of constant false alarm rate processing allows the probability of false alarm to remain relatively constant across range cells. Given that constant false alarm rate devices fulfill their purpose, the assumption that the process noise is approximately uniform over the surveillance volume is not unreasonable. Section 6.7.1 assesses the performance of the algorithm when the uniformity assumption is not valid, by changing the true probability of detection (P_D) and probability of false alarm (P_F) of the feature tests from their assumed values.

Note that there is no 'measurement noise' associated with the clutter; therefore the model for the spatial clutter accounts for the process noise only.

The process model describes the dynamic behavior of the target. In the specific case being considered, it is assumed that because of the extended ranges at which target declaration is to take place, the angular movement (in

azimuth and elevation) of the target will be minimal and is modelled as being Gaussian over a small time window. This is due to the nature of the ship self-defense problem, where the target is aimed at the sensor platform, and therefore will pursue a flight path that is almost radially inbound. A first order Markov model is used to describe target motion in range coordinates. This assumption allows easy, high speed computation of the range/range rate feature tests. Higher order models (such as a three state (range, range rate and acceleration) Kalman filter) could potentially improve the performance of the various feature tests. However, this usually comes at the expense of increased sensitivity to modelling errors, see Bar-Shalom [3]. In addition, the system should be able to declare on as few data points as is possible. As the number of states in the filter rises, so do latencies in detection time (since we require more data for a higher order filter).

The measurement model assumes that measurement errors have a Gaussian distribution. The Gaussian assumption is a standard one made in many target tracking problems, where it is assumed that the measurement x at iteration n, denoted x_n, is an observation of the true constant or random parameter m that is perturbed by zero-mean Gaussian random noise with variance σ_ϵ^2:

$$x_n = m + \epsilon_n, \quad \epsilon_n \sim N(0, \sigma_\epsilon) \tag{3-1}$$

3.2 Derivation of the Feature Tests

Lattice distributions are used to model the PDFs of the target and clutter azimuth and elevation features. This is done because sensors have finite resolutions; therefore continuous models are not appropriate. However, the range rate distribution is practically continuous, because the radar being investigated for multisensor track initiation (MSTI) purposes uses a multiple hypothesis algorithm to obtain a continuous estimate of the range rate for each radar data point.

The target dynamic model assumes that target movement is a Markov process, that is, if x_n is a measurement taken at time t_n then $P\{x_n \mid x_{n-1},H_i\}= P\{x_n \mid x_{n-1},x_{n-2},...,x_1,H_i\}$ [30], and in the interest of reducing the dimensionality of the feature space, the feature test PDFs are computed on the basis of the difference between two random variables, which can be either like measurements or calculated values. The pair-wise difference of two consecutive measurements is used for the azimuth and elevation feature tests (*i.e.* a_{i+1}-a_i, and e_{i+1}-e_i, for example); the difference between the calculated straight line prediction of range and the actual range measurement is used for range/range rate correlation.

3.2.1 Clutter Feature Probability Density Functions

Consider a measurement pair in either azimuth or elevation. Let the random variable X represent the first measurement, and Y the second. Let Z=f(X,Y)=X-Y. Thus, we seek to define the statistics of Z for two cases: that in which both measurements x and y originate from the same sensor, and that in which they originate from different sensors.

As argued in Section 3.1, for the clutter case, X and Y are uniformly distributed in space and are of lattice type. Therefore, the PDFs of the measurements for the clutter case can be expressed as follows:

$$p_X(x)=\frac{1}{r_x}\sum_{i=0}^{r_x-1}\delta(x-i\frac{F}{r_x}) \tag{3-2}$$

$$p_Y(y)=\frac{1}{r_y}\sum_{j=0}^{r_y-1}\delta(y-j\frac{F}{r_y}) \tag{3-3}$$

where F is the field of view, r_x is the number of sensor resolution elements in F for the sensor that produced x, and r_y is the number of sensor resolution elements in F for the sensor that produced y. From Papoulis [30], if Z=X-Y, X and Y independent, then $p(x,y)=p_X(x)p_Y(y)$ and

$$p_Z(z)=\int_{-\infty}^{\infty} p_X(y+z)p_Y(y)dy \tag{3-4}$$

this integral is recognized as the correlation of $p_X(z)$ and $p_Y(-z)$. Thus, the PDF of the feature z can be expressed as:

$$\begin{aligned} p_Z(z)&=\frac{1}{r_x r_y}\int_{-\infty}^{\infty}\sum_{i=0}^{r_x-1}\sum_{j=0}^{r_y-1}\delta(y+z-i\frac{F}{r_x})\delta(y-j\frac{F}{r_y})dy \\ &=\frac{1}{r_x r_y}\sum_{i=0}^{r_x-1}\sum_{j=0}^{r_y-1}\delta(z+F(\frac{j}{r_y}-\frac{i}{r_x})) \end{aligned} \tag{3-5}$$

For identical sensors, $r_x=r_y$, and the expression for the PDF of the feature z becomes:

$$p_Z(z)=\frac{1}{r_x^2}\sum_{i=0}^{r_x-1}\sum_{j=0}^{r_x-1}\delta(z+(j-i)\frac{F}{r_x}) \tag{3-6}$$

Denote the first azimuth measurement as x_{a1}, the second as x_{a2}, and the difference as $x_{ad}=x_{a1}-x_{a2}$, then let $x=x_{a1}$, $y=x_{a2}$, and $z=x_{ad}$; then the PDF of the azimuth difference feature for the clutter case is given by equation (3-6). The expression for the elevation difference feature x_{ed} is defined similarly in terms of the first and second elevation measurements x_{e1} and x_{e2}, respectively.

It was assumed that the azimuth and elevation of a target stayed relatively constant over a time window at extended ranges (*i.e.*, m in equation

(3-1) was assumed to be constant). However, this is not true for the range rate. It is therefore necessary to use a feature that accounts for change when performing range/range-rate correlation. For this reason, the maximum a posteriori (MAP) estimator was used to arrive at an estimated value for the target range rate, v_m. The latter value was then used to compute the range/range-rate correlation feature. The maximum a posteriori estimator is typically used to estimate a random parameter using a number of observations corrupted by noise. The application at hand can be viewed in a similar fashion. For example, the first range-rate measurement v_1 is made with some finite accuracy indicated by the associated variance $\sigma_{v_1}^2$. This measurement can then be treated as the random parameter to be estimated using the second range-rate measurement v_2 made with associated variance $\sigma_{v_2}^2$. From Sorenson [31], the expression for the maximum a posteriori estimator of the range rate v_m is given by:

$$v_m = \frac{\sigma_{v_2}^2 v_1 + \sigma_{v_1}^2 v_2}{\sigma_{v_1}^2 + \sigma_{v_2}^2} \qquad \textbf{(3-7)}$$

with its associated variance $\sigma_{v_m}^2$:

$$\sigma_{v_m}^2 = \frac{\sigma_{v_1}^2 \sigma_{v_2}^2}{\sigma_{v_1}^2 + \sigma_{v_2}^2} \tag{3-8}$$

The expression for the first order predicted range of the second measurement x_{r2}' using x_{r1} (the range of the first measurement), v_1 and v_2 is given by:

$$x_{r2}' = v_m \cdot \Delta t + x_{r1} \tag{3-9}$$

The feature to be tested is x_{r2}-x_{r2}' (actual-predicted range).

Note that the values of σ_{v_1} and σ_{v_2} are in general time dependent even though that is not explicitly indicated here (see Appendix C for the error model). Note also that the measurements of v_1 and v_2 are continuous, but the range measurements are not. This complicates the derivation of the feature for the range/range-rate correlation PDF. If the number of range cells is sufficiently large, the lattice distribution of the range measurements can be approximated by a continuous one. We desire to determine the feature PDF for the difference between the actual and the predicted range:

$$z = x'_{r2} - x_{r2}$$
$$= \frac{\sigma_{v_2}^2 \cdot \Delta t}{\sigma_{v_1}^2 + \sigma_{v_2}^2} v_1 + \frac{\sigma_{v_1}^2 \cdot \Delta t}{\sigma_{v_1}^2 + \sigma_{v_2}^2} v_2 + x_{rI} - x_{r2} \quad \textbf{(3-10)}$$

with v_1, v_2, x_{r1}, x_{r2} uniformly distributed[15]. This requires the convolution of 4 uniform PDFs. It is first necessary to compute the intermediate results for convolving 2 and 3 distributions. We can proceed as follows: If Z=X+Y, $p_X(x)=U(a,b)$, and $p_Y(y)=U(c,d)$, where U(a,b) denotes the uniform PDF nonzero on the interval (a,b), then:

$$\begin{aligned} p_Z(z) &= \int_{-\infty}^{\infty} p_X(x) p_Y(z-x)\,dx \\ &= p_X(z) * p_Y(z) \\ &= \frac{1}{(b-a)(d-c)} [(z-d-b)I(z-d-b) - (z-c-b)I(z-c-b) - (z-d-a)I(z-d-a) + \\ &\qquad (z-c-a)I(z-c-a)] \end{aligned} \quad \textbf{(3-11)}$$

Note that the result is a piecewise polynomial of order 1 and that $p_Z(z)$ is trapezoidal in shape with height 1/((b-a)(d-c)) and vertices at x={b+d,a+c,b+c,a+d}. The distribution of Z=W+X+Y, $p_W(w)=U(a,b)$, $p_X(x)=U(c,d)$,

---

[15] The use of sensitivity time control should allow most of the radar returns to be approximately uniformly distributed (since the sensitivity time control will eliminate most, but not all of the bird returns), and thus the multiple hypothesis range-rate estimate should be approximately continuous. It is furthermore assumed that these measurements are statistically independent, although there may be some dependence between range-rate measurements in the multiple hypothesis range-rate estimate. The effect of deviations from these assumptions is addressed in Section 6.7.1.

$p_Y(y)=U(e,f)$ can be similarly determined. The result is a piecewise polynomial of order 2:

$$p_Z(z)=\frac{1}{(b-a)(d-c)(f-e)}[-Qr(z,f+d+b)+Qr(z,f+c+b)+Qr(z,f+d+a)-Qr(z,f+c+a)$$
$$+Qr(z,e+d+b)-Qr(e+c+b)-Qr(z,e+d+a)+Qr(z,e+c+a)] \quad \textbf{(3-12)}$$

where

$$Qr(z,r)=\frac{1}{2}(z-r)^2 I(z-r) \quad \textbf{(3-13)}$$

The distribution of $Z=V+W+X+Y$ where $p_V(v)=U(a,b)$, $p_W(w)=U(c,d)$, $p_X(x)=U(e,f)$, $p_Y(y)=U(g,h)$ can be described by a piecewise polynomial of order 3:

$$p_Z(z)=\frac{1}{(b-a)(d-c)(f-e)(h-g)}[-Cb(z,h+f+d+b)-Cb(z,h+f+c+b)-Cb(z,h+f+d+a)$$
$$+Cb(z,h+f+c+a)+Cb(z,h+e+d+b)+Cb(z,h+e+c+b)$$
$$+Cb(z,h+e+d+a)-Cb(z,h+e+c+a)+Cb(z,g+f+d+b)$$
$$+Cb(z,g+f+c+b)+Cb(z,g+f+d+a)-Cb(z,g+f+c+a)$$
$$-Cb(z,g+e+d+b)-Cb(z,g+e+c+b)-Cb(z,g+e+d+a)$$
$$+Cb(z,g+e+c+a)] \quad \textbf{(3-14)}$$

where:

$$Cb(z,r)=\frac{1}{6}(z-r)^3 I(z-r) \tag{3-15}$$

A plot of $p_Z(z)$ in equation (3-14) with a=c=e=g=-1, b=d=f=h=1, is shown in Figure 3.2.

If v_1 and v_2 are $U(v_{min}, v_{max})$; x_{r1} and x_{r2} are $U(r_{min},r_{max})$, we can then make the following substitutions:

$$\begin{aligned}
V &\sim U\left(\frac{\Delta t\sigma_{v_1}^2}{\sigma_{v_1}^2+\sigma_{v_2}^2}v_{min}, \frac{\Delta t\sigma_{v_1}^2}{\sigma_{v_1}^2+\sigma_{v_2}^2}v_{max}\right) \\
W &\sim U\left(\frac{\Delta t\sigma_{v_2}^2}{\sigma_{v_1}^2+\sigma_{v_2}^2}v_{min}, \frac{\Delta t\sigma_{v_2}^2}{\sigma_{v_1}^2+\sigma_{v_2}^2}v_{max}\right) \\
X &\sim U(r_{min},r_{max}) \\
Y &\sim U(-r_{max},-r_{min})
\end{aligned} \tag{3-16}$$

so that the PDF of the difference between the actual and the predicted range is now given by equation (3-14).

3.2.2 Target Feature Probability Density Functions

The following development closely parallels that of the clutter case in Section 3.2.1. Here, however, the underlying distributions of X and Y will be Gaussian, with the variance of the estimates of m in equation (3-1) typically changing with time, and the type of sensor. Furthermore, the distributions will

be of lattice type due to the finite resolution of the sensors. This means that the probability mass must be calculated over one resolution cell for each impulse function of the PDF. Given this fact, the PDFs of the target azimuth/elevation measurements are as follows:

$$p_X(x)=\sum_{i=0}^{r_x-1}\int_{(i-\frac{1}{2})\frac{F}{r_x}}^{(i+\frac{1}{2})\frac{F}{r_x}}\frac{1}{\sqrt{2\pi}\sigma_x}e^{-(\frac{(q-m)^2}{2\sigma_x^2})}\,dq\;\delta(x-i\frac{F}{r_x}) \qquad \textbf{(3-17)}$$

$$\triangleq\sum_{i=0}^{r_x-1}p_{xi}\,\delta(x-i\frac{F}{r_x})$$

$$p_Y(y)=\sum_{j=0}^{r_y-1}\int_{(j-\frac{1}{2})\frac{F}{r_y}}^{(j+\frac{1}{2})\frac{F}{r_y}}\frac{1}{\sqrt{2\pi}\sigma_y}e^{-(\frac{(q-m)^2}{2\sigma_y^2})}\,dq\;\delta(y-j\frac{F}{r_y}) \qquad \textbf{(3-18)}$$

$$\triangleq\sum_{j=0}^{r_y-1}p_{yj}\,\delta(y-j\frac{F}{r_y})$$

where m is the true target position. It may be necessary to scale $p_X(x)$ and $p_Y(y)$ in equations (3-17) and (3-18) to maintain the law of total probability, because of the finite summations present in both equations; this is especially true if m falls near the boundaries of the field-of-view.

Performing the correlation of PDFs necessary for evaluating the PDF of Z=X-Y, with the assumption that the measurement noises are independent, results in:

$$p_Z(z)=\sum_{i=0}^{r_x-1}\sum_{j=0}^{r_y-1} p_{xi}p_{yj}\delta(z+j\frac{F}{r_y}-i\frac{F}{r_x}) \qquad \textbf{(3-19)}$$

For identical sensors, this expression reduces to:

$$p_Z(z)=\sum_{i=0}^{r_x-1}\sum_{j=0}^{r_x-1} p_{xi}p_{xj}\delta(z+(j-i)\frac{F}{r_x}) \qquad \textbf{(3-20)}$$

If we set x_{a1}=x and x_{a2}=y then x_{ad}=z and the PDF of the azimuth difference feature for the target case is given by the above equation. The expression for x_{ed} is defined similarly. A plot of $p_Z(z)$ according to equation (3-19) for the infrared-radar azimuth difference feature is shown in Figure 3.3, with a gate size of F=2.56°, a mean m=1.28°, that encompasses r_x=89 infrared resolution cells, and r_y=25 radar resolution cells (Appendix C gives additional detail on sensor parameters). The expressions for σ_x and σ_y are in general time-dependent and can be obtained from a second order model (see Appendix A).

In Section 3.2.1 we saw that the expressions for the range/range-rate correlation feature were given by equations (3-7)-(3-10). This required the convolution of four PDFs. For the target case the associated random variables are mutually independent , and the PDFs are Gaussian and defined as follows:

$$p_{V_m}(v_m)=N(\bar{v}_m,\sigma_{v_m}) \tag{3-21}$$

where the over bar operator ($\bar{}$) denotes the expected value, and

$$p_{X_{r1}}(x_{r1})=N(\bar{x}_{r1},\sigma_{x_{r1}}) \tag{3-22}$$

then the scaled random variable $V_m \cdot \Delta t$ has a PDF as follows:

$$p_{V_m\Delta t}(v_m\Delta t)=N(\bar{v}_m\Delta t,\sigma_{v_m}\Delta t) \tag{3-23}$$

The first order prediction $X_{r2}'=V_m \cdot \Delta t + X_{r1}$ then has PDF

$$\begin{aligned} p_{X'_{r2}}(x'_{r2}) &= N(\bar{v}_m\Delta t,\sigma_{v_m}\Delta t)*N(\bar{x}_{r1},\sigma_{x_{r1}}) \\ &= N(\bar{x}'_{r2},\sigma_{x'_{r2}}) \end{aligned} \tag{3-24}$$

where

$$\bar{x}'_{r2}=\bar{v}_m\Delta t+\bar{x}_{r1} \tag{3-25}$$

$$\sigma_{x'_{r2}}=\sqrt{(\sigma_{v_m}\Delta t)^2+\sigma_{x_{r1}}{}^2} \tag{3-26}$$

Let $X_{rd}=X_{r2}'-X_{r2}$, then:

$$p_{X_{rd}}(x_{rd}) = N(\bar{x}'_{r2}, \sigma_{x'_{r2}}) * N(-\bar{x}_{r2}, \sigma_{x_{r2}}) = N(\bar{x}_{rd}, \sigma_{rd}) \tag{3-27}$$

where

$$\bar{x}_{rd} = \bar{x}'_{r2} - \bar{x}_{r2} \tag{3-28}$$

$$\sigma_{x_{rd}} = \sqrt{\sigma^2_{x'_{r2}} + \sigma^2_{x_{r2}}} = \sqrt{(\Delta t)^2 \frac{\sigma^2_{v_1}\sigma^2_{v_2}}{\sigma^2_{v_1} + \sigma^2_{v_2}} + \sigma^2_{x_{r1}} + \sigma^2_{x_{r2}}} \tag{3-29}$$

If we assume that the first order approximation of x_{r2}' is unbiased, then $E[x_{rd}']=0$ and

$$p_{X_{rd}}(x_{rd}) = N(0, \sigma_{x_{rd}}) \tag{3-30}$$

3.3 Determining the Minimum Probability of Error Thresholds

The minimum probability of error threshold for each feature test can be determined by a one-dimensional search for the two sided hypothesis test threshold that minimizes $P_{ERROR}=1+(P_F-P_D)$. Where P_F and P_D are defined for an arbitrary feature test as follows:

$$P_F = \int_{-z_T}^{z_T} p_Z(z|H_0)dz$$
$$P_D = \int_{-z_T}^{z_T} p_Z(z|H_1)dz \qquad \textbf{(3-31)}$$

Note that $p_Z(z|H_0)$ is given by equations (3-5) and (3-14), and that $p_Z(z|H_1)$ is given by equations (3-19), and (3-30). The use of a two-sided test means that the features tests are actually dependent only on the absolute values of the features in equations (3-5), (3-14), (3-19), and (3-30). The results of these one-dimensional searches are detailed in Section 6.4.

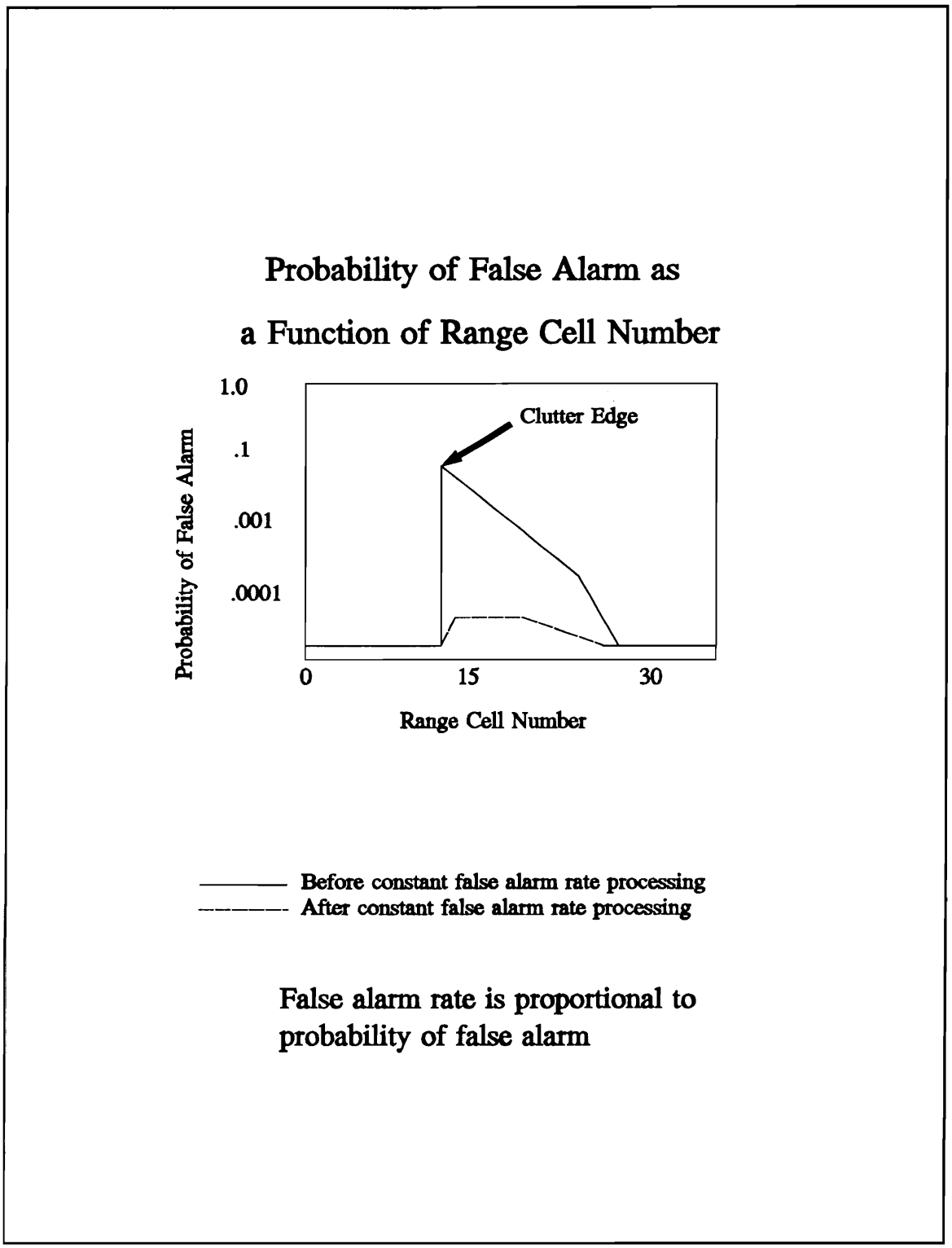


Figure 3.1 Operation of a radar constant false alarm rate processor

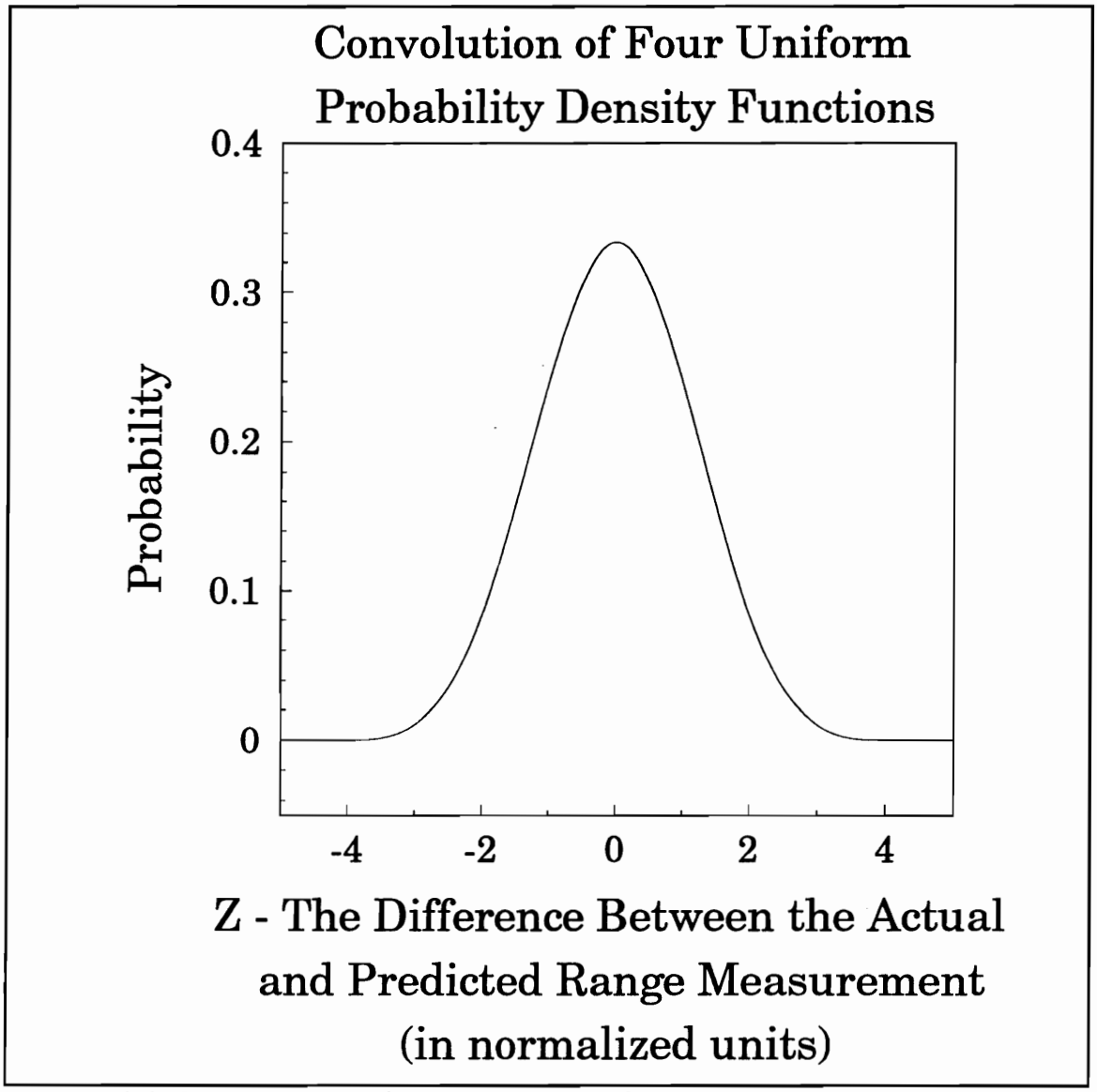


Figure 3.2 Plot of range/range-rate correlation noise feature probability density function

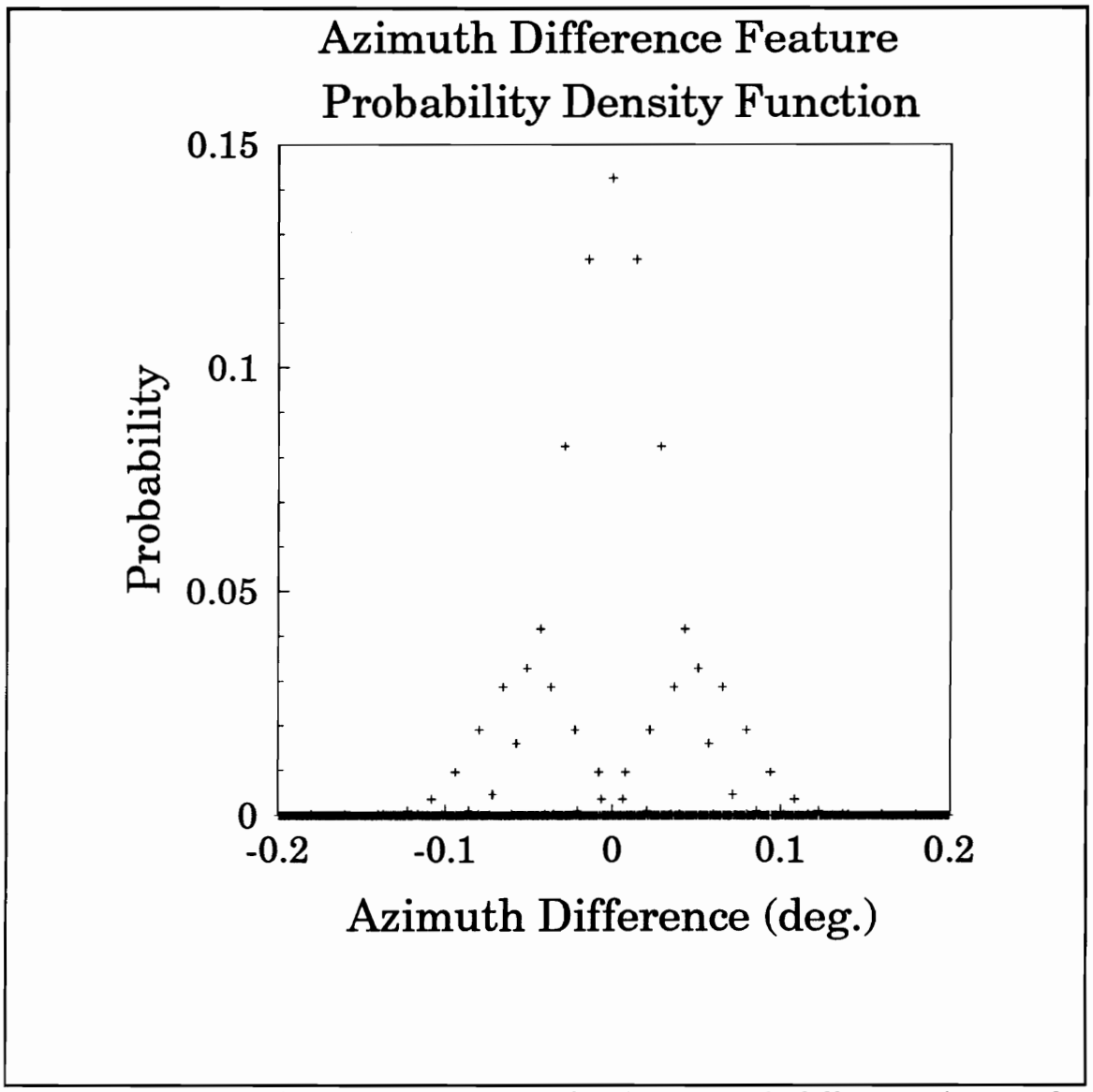


Figure 3.3 Probability density function for the azimuth difference feature for radar and infrared sensor data

Chapter 4: Data/Feature Test Fusion Method

4.1 Use of a Bi-level Threshold in the Log-Likelihood Ratio Test

As discussed in Section 1.3, the log-likelihood ratio is computed using a weighted sum of the individual sensor or feature test results. The log-likelihood ratio is used as a test statistic in a threshold test. Typically, only one threshold is used in the test [14] [15]. Chair, Hoballah and Varshney [26] have proposed the addition of a second threshold test, the results of which are then used to determine if additional data are required from the sensors. Their method closely parallels that of Wald [13], except for modifications to the log-likelihood ratio calculations to account for the quantization of sensor decisions. The solutions to the coupled equations that result are mathematically and computationally intractable [26].

The present work is different in that it assumes that requests for additional sensor data can be made but not necessarily satisfied. The determination of the additional threshold is dictated by the rate at which the sensors can respond to requests for additional data (*i.e.* Neyman-Pearson constraints). The present work also provides a method for determining the solution to the simplified optimization problem, under the assumption that some feature tests/measurements may be missing.

The threshold test then takes on the following form:

$\Lambda \geq \lambda_1$	Declare H_1 (target present) and initiate a track
$\lambda_1 > \Lambda \geq \lambda_0$	Request another measurement (or 'cue' sensor)
$\lambda_0 > \Lambda$	Declare H_0 (No target present)

where Λ is the value of the log-likelihood ratio calculated for **u** (the vector of feature test results), as in equation (2-1). We define λ_1 to be the declaration threshold, and λ_0 the cueing threshold.

4.2 Missing Measurements

All contemporary data fusion methods to date have implicitly assumed that each of the sensors is capable of providing data all of the time. This does not account for cases where the feature spaces of the sensor data overlap only partially[16] or for times when sensor detection fails to occur.

The set of feature tests required for a particular sequence of multisensor data may be dependent not only on the composition of the sequence (*i.e.* the sensors contributing to the data sequence) but on the order as well. This can occur whenever order dependent features are used, or when a Markov assumption is utilized. Therefore, a good representation for a particular group of multisensor data would be the base M representation of the sequence, where

[16] An example of this is the use of radar and infrared sensors. In this case, both the radar and the infrared sensors can provide azimuth measurements. However, the infrared sensor is incapable of providing range or range-rate information. Thus, the feature spaces of the two sensors overlap only partially.

M is the number of sensors. If the maximum sequence length for a data sequence is s, then the number of different types of data sequences of length greater than 1 but less than or equal to s will be:

$$NA = \sum_{i=2}^{s} M^i \tag{4-1}$$

Let W represent the set of all possible time ordered sensor data sequences obtained from M sensors of length greater than 1 but less than or equal to s[17]. Note that NA=$\mathfrak{C}$(W), where $\mathfrak{C}(\cdot)$ denotes the cardinality operator. In turn, let w denote a particular ordered sequence of multisensor data, where $w \in W$.

Let $z_w \subset Z$ denote the subset of features for the data sequence of type w, with Z the set of all possible z_w. This means that the specification of w completely defines the probability density function (PDF) of the conditional log-likelihood ratio, because it completely specifies the feature tests to be computed. A mapping can then be constructed:

$$\exists\ \nu\colon W \rightarrow Z \tag{4-2}$$

[17] For example, in the two sensor case, W would be the set of all possible binary vectors of length greater than two, but less than s.

Note that ν is not one-to-one (*i.e.*, the same set of features may arise from different sequences of data). Furthermore, if the cardinality of z_w is given by $\mathfrak{C}(z_w)$, $N=\mathfrak{C}(z_w)$ would be used to compute the log-likelihood ratio in equation (2-1).

Define $P(w|H_i)$ as being equal to the relative frequency of occurrence of a particular sequence of data w, conditional on hypothesis H_i,

$$\Pr(w|H_i) \Delta \frac{O(w|H_i)}{O(W|H_i)} \tag{4-3}$$

where $O(w|H_i)$ is the number of times the data group type w will occur out of a total of $O(W|H_i)$ samples. Note the implicit assumption that $P(w|H_i)$ is stationary. The effect of a deviation of $P(w|H_i)$ from expected values is considered in Section 6.7.1.

It should be noted that in some cases $O(w|H_i)$ may be zero. This may arise when the number of samples is so small that w does not occur. However, it still may be possible that the event w occurs. For this reason, $P(w|H_i)$ is always assigned a small positive value, and the remaining $P(w|H_i)$ terms are renormalized to account for the change. This serves to make the multisensor track initiation (MSTI) algorithm more robust.

The concept of the relative frequency of occurrence can be used to account for missing measurements. $P(w|H_i)$ weights the contribution that each data sequence makes to the overall system P_D, P_C (probability of cue) and P_F.

Because the different data sequences will have different features, and hence have different feature test statistics, the declaration and cue thresholds discussed in Section 4.1 are conditional on w. The following section will note this dependence.

4.3 The Probability Density Function of the Conditional Log-likelihood Ratio

The use of Neyman-Pearson constraints to determine cue and declaration thresholds dictates that expressions for the cumulative probability density function (CPDF) and PDF of the log-likelihood ratio be developed. This section will illustrate that these expressions are functions of P_{Di} and P_{Fi} (the probabilities of detection and false alarm for the feature test i, respectively).

Thomopoulos *et al.* [21] discuss the case in which P_{Di} and P_{Fi} are the same across all feature tests. In this case, the PDF of K tests with $u_i=1$ occurring is given by:

$$p_K(k|H_1)=B(N,a) \quad , \quad k=0,...,N \tag{4-4}$$

$$p_K(k|H_0)=B(N,f) \quad , \quad k=0,...,N \tag{4-5}$$

with

$$\begin{aligned} a&=P_{Di},\ P_{Di} \text{ a constant} \\ f&=P_{Fi},\ P_{Fi} \text{ a constant} \end{aligned} \tag{4-6}$$

and where B(N,a) denotes the binomial distribution with parameters N and a. The determination of the proper threshold for the log-likelihood ratio Λ is then obtained via a one-dimensional search. Note that a value of k, k^* can be determined, such that if k^* or more feature tests have result $u_i=1$, the fusion center will declare $u_f=1$ also. This represents the so-called "k-out-of-N" decision rule.

However, if the values of P_{Di} and P_{Fi} are different across feature tests - as they are in many cases - and if a proper substitution of variables is made, the following expressions for the PDFs of the conditional log-likelihood ratios for each non-zero weighted feature test result[18]:

$$p_{\Lambda_i}(\lambda \mid H_0)=g_i\delta(\lambda-c_i)+f_i\delta(\lambda-d_i) \tag{4-7}$$

$$p_{\Lambda_i}(\lambda \mid H_1)=b_i\delta(\lambda-c_i)+a_i\delta(\lambda-d_i) \tag{4-8}$$

where δ is the Dirac delta function, Λ_i is the log-likelihood ratio for the i^{th} weighted feature test and,

[18] A value of $u_i=0$ does not change the log-likelihood ratio, however, it does add an additional point to the PDF of the log-likelihood ratio. We can reduce the complexity of the overall fusion center log-likelihood ratio PDF by ignoring the $u_i=0$ contribution in the composite log-likelihood ratio, and using the concept of data sequence type to account for it. This will be illustrated later in this section.

$$
\begin{aligned}
a_i &= P_{Di} \\
b_i &= 1 - P_{Di} \\
c_i &= \log\left(\frac{1-P_{Di}}{1-P_{Fi}}\right) \\
d_i &= \log\left(\frac{P_{Di}}{P_{Fi}}\right) \\
f_i &= P_{Fi} \\
g_i &= 1 - P_{Fi}
\end{aligned}
\qquad \textbf{(4-9)}
$$

The log-likelihood ratio at the fusion center is the sum of the log-likelihood ratios for the weighted feature tests. Therefore, the PDF of the log-likelihood ratio at the fusion center will be the convolution of the PDFs of the Λ_is:

$$
p_{\Lambda}(\lambda \mid H_i) = p_{\Lambda_1}(\lambda \mid H_i) * \ \ldots \ * p_{\Lambda_N}(\lambda \mid H_i) \qquad \textbf{(4-10)}
$$

Taking the dual-sided Laplace transform results in:

$$
p_{\Lambda}(\lambda \mid H_1) = \mathcal{L}^{-1}\left\{\prod_{i=1}^{N} \left(b_i e^{-c_i s} + a_i e^{-d_i s}\right)\right\} \qquad \textbf{(4-11)}
$$

$$
p_{\Lambda}(\lambda \mid H_0) = \mathcal{L}^{-1}\left\{\prod_{i=1}^{N} \left(g_i e^{-c_i s} + f_i e^{-d_i s}\right)\right\} \qquad \textbf{(4-12)}
$$

It is important to note that the PDFs of the conditional log-likelihood ratios are discrete and members of the general family of multinomial PDFs. Note that because equations (4-11) and (4-12) are PDFs of a discrete random variable, it may not be possible to attain the exact system probability of false

alarm (P_F) that is desired. This may require a relaxation of constraints and/or the introduction of penalty functions into the optimization problem, which is discussed in Chapter 5.

More than one type of data sequence may occur, in which case the variables N (the number of feature tests), a_i, b_i, c_i, d_i, f_i and g_i (equation (4-9)) will be dependent on w. This dependence is noted explicitly in subsequent equations. After optimizing the values for the declaration threshold ($\lambda_{1,w}$) for each data sequence type w, the probability of detection, false alarm, cue and detection for a cue for a particular type of data sequence (Appendix B) will be:

$$P_{D,w}=\sum_{q_w} \psi_{1,w}(q_w) I(\phi_w(q_w)-\lambda_{1,w}) \qquad \textbf{(4-13)}$$

$$P_{F,w}=\sum_{q_w} \psi_{0,w}(q_w) I(\phi_w(q_w)-\lambda_{1,w}) \qquad \textbf{(4-14)}$$

$$P_{C,w}=\sum_{q_w} \psi_{0,w}(q_w)(I(\phi_w(q_w)-\lambda_{0,w})-I(\phi_w(q_w)-\lambda_{1,w})) \qquad \textbf{(4-15)}$$

$$P_{DC,w}=\sum_{q_w} \psi_{1,w}(q_w)(I(\phi_w(q_w)-\lambda_{0,w})-I(\phi_w(q_w)-\lambda_{1,w})) \qquad \textbf{(4-16)}$$

where q_w is a member of the index set Q_w that references the terms of the expansion shown in equation (B-2), $\phi(q_w)$ is the q_w^{th} term in the sum of sums of $c_{i,w}$ and $d_{i,w}$. $\Psi_{1,w}(q_w)$ and $\Psi_{0,w}(q_w)$ are the q_w^{th} monomials of the multinomials

in $a_{i,w}$, $b_{i,w}$ and $f_{i,w}$, $g_{i,w}$, respectively. It should be noted that as N increases, the number of points in the CPDF rises exponentially (for unequal P_{Di} and P_{Fi}). Therefore, it may be advantageous to approximate the log-likelihood ratio. Two methods of approximating the log-likelihood ratio using formulas found in Abramowitz and Stegun [32] are discussed in Appendix D.

The form of the expressions in equations (4-13)-(4-16) facilitate the computation of the upper-tail probabilities on a computer. Appendix B discusses an efficient method for the computation of the PDF of the conditional log-likelihood ratio.

Using the relative frequency of occurrence, we can arrive at expressions for the system probabilities of detection, false alarm, cue and detection for a cue:

$$P_D=\sum_w P(w|H_1)\sum_{q_w} \psi_{1,w}(q_w)\,I(\phi_w(q_w)-\lambda_{1,w}) \qquad \textbf{(4-17)}$$

$$P_F=\sum_w P(w|H_0)\sum_{q_w} \psi_{0,w}(q_w)\,I(\phi_w(q_w)-\lambda_{1,w}) \qquad \textbf{(4-18)}$$

$$P_C=\sum_w P(w|H_0)\sum_{q_w} \psi_{0,w}(q_w)(I(q_w)-\lambda_{0,w})-I(\phi_w(q_w)-\lambda_{1,w})) \qquad \textbf{(4-19)}$$

$$P_{DC}=\sum_{w} P(w|H_1)\sum_{q_w} \psi_{1,w}(q_w)(I(q_w)-\lambda_{0,w})-I(\phi_w(q_w)-\lambda_{1,w})) \qquad \textbf{(4-20)}$$

The Neyman-Pearson constrained optimization uses these equations to find λ_1 and λ_0 such that P_F and P_C attain desired values, while P_D and P_{DC} are maximized. The Neyman-Pearson constrained optimization is detailed in the following chapter.

Chapter 5: Optimization of Log-likelihood Ratio Thresholds

5.1 Problem Description

The objective of the Neyman-Pearson constrained problem is to maximize the system probabilities of detection (P_D) and detection on a cue (P_{DC}) at the fusion center while keeping a desired system of probability of false alarm (P_F) and probability of cue (P_C). Attaining a desired P_F or P_C is not always possible because of two factors: one is the discrete nature of the log-likelihood ratio probability density functions (PDFs)[19], and the other is the probabilities of detection (P_{Di}s) and false alarm (P_{Fi}s) of the feature tests.

This makes the addition of the following necessary:

1. Thresholds that effectively exclude certain data sets (*i.e.* thresholds that exceed the domain of $\phi_{i,w}(q_w)$, which defines all the values that the log-likelihood ratio can attain) - this promotes feasibility of the threshold solutions.

2. Some manner of discouraging the exclusion. This penalizes the exclusion of data sets that may have relatively poor decision statistics but which may have some other redeeming value; an example of this would be data sequences that contain both radar and infrared data [20].

---

[19] This is detailed in Appendix B.

[20] The problem being addressed is one in which both the target declaration and the data contributing to the declaration are important from a track-initiation standpoint. This being the case, a declaration resulting from a data

5.2 Scale of the Optimization Problem

Because of the discrete nature of the PDF of the log-likelihood ratio, the number of possible solutions to the optimization problem for various values of the maximum sequence length (s) can be determined. This insight will enable the selection of effective techniques for dealing with the problem.

An upper limit on the number of possible solutions to the threshold optimization is given as:

$$NS_s = \prod_{w \in W} (2^{\mathbf{C}(z_w)} + 1) \qquad \textbf{(5-1)}$$

Where W is the index set of data sequences, w refers to a particular data sequence, $\mathbf{C}(\cdot)$ is the cardinality operator, and z_w is the subset of features for the data sequence w. Note that the 1 is present in the product term to account for the possibility of a threshold $\lambda_{i,w}=\infty,(\infty \rightarrow \infty)$ (*i.e.*, that a particular sequence in W is ignored). Evaluating the above expression for s={2,3,4} gives:

$$NS_2 = (4+1)(2+1)(2+1)(4+1) = 225$$

$$NS_3 = (2^4+1)(2^3+1)(2^2+1)(2^3+1)(2^3+1)(2^2+1)(2^3+1)(2^4+1)NS_2$$

$$= 1.06657 \times 10^{10}$$

---

sequence that contains data from different sensors is inherently more valuable to the track initiation process in the ship self-defense problem than is a declaration that results from a data sequence containing data from only one sensor; this, however, must be balanced against tradeoffs in declaration time.

$NS_4=1.06681x10^{22}NS_3=1.13783x10^{32}$

for the two sensor case with two feature calculations for each radar-radar and infrared-infrared detection pair, and one feature measurement for each infrared-radar detection pair. It can easily be seen that the number of solutions to the threshold optimization climbs exponentially as the maximum sequence length is increased. This phenomenon also occurs when the number of feature tests per detection pair increases. Given this fact, it may be advisable under suitable conditions (*i.e.*, an extremely large number of feature tests and/or a long sequence length) to approximate the discrete cumulative probability density functions (CPDFs) of the conditional log-likelihood ratios (Appendix D).

5.3 Existing Methods for Performing the Optimization

Many alternatives exist for solving the Neyman-Pearson-constrained optimization problem. These alternatives may be grouped into two categories: those that rely on solving relaxations (approximations) of the problem, and those that solve an exact version of the problem.

In using any continuous approximation to the problem, infeasible solutions may result. However, use of the approximation will in some cases make the problem tractable when previously it was not. Unfortunately, the fact that these algorithms do not guarantee a feasible solution renders them

unsuitable for the current problem. Current data fusion literature (for example, see [22],[26]) has focused on the method of Lagrange as a means of solving the nonlinear optimization problem[21]. Application of the Lagrange method would require a continuous approximation of the discontinuous PDFs of the conditional log-likelihood ratio; thus, this method falls into the continuous approximation category.

Formulating the problem at hand in terms of an integer-programming problem [33] is possibly the only way in which the feasibility of the final solution can be guaranteed. There are different ways in which the integer programming problem can be approached -- as a 0/1 binary integer program or as a non-linear integer program The problem with the binary integer program formulation is the extremely large number of binary variables generated. For the current problem formulation, the number of variables is given by:

$$N_{VAR}=\sum_{w\in W}(2^{\Phi(z_w)}+1) \qquad \textbf{(5-2)}$$

For example, for the case in which the maximum sequence length s=2 the resulting binary integer program has (5+3+3+5)=16 binary variables. For a maximum sequence length of s=3, the number of binary variables would be (80+16)=96. The number of binary variables in the problem rises exponentially

[21] The Lagrange method is often useful from an analytical standpoint for small problems, but is not suitable for the actual solution of large ones, especially those that are discrete.

with increasing sequence length. In contrast, if the problem were formulated as a nonlinear integer program, the number of variables would be given by the following expression:

$$N_{VAR}=\mathbb{C}(W) \tag{5-3}$$

We see that for the nonlinear integer program, we would have 4 and 12 variables in the optimization, respectively. Thus there is a reduction in the number of variables in the nonlinear integer program problem. However, the binary integer program has some advantages in many problems; it is linear, and reliable techniques can be used to solve it, whereas the nonlinear integer program is not. But, as the sequence length rises so does the number of variables (and constraints), and problems arise with storage of the binary integer program - along with the computation of solutions to the intermediate linear program relaxations -such as occurs in the Land-Doig branch and bound techniques or when cutting-plane methods are used [33]. Note also that the binary integer program can be formulated as a knapsack problem, for which the only solutions are non-polynomial time [34].

Large-scale nonlinear integer program problems of formulations similar to this one are typically difficult to solve, and sometimes the best approach is to use one of the so-called "direct" methods. Unfortunately, execution of many direct-method algorithms is usually painfully slow due to the oft-mentioned

"curse of dimensionality" [35]. Another family of direct techniques based on statistical perturbations, loosely termed "adaptive heuristics" by Nahar *et al.* [36] are also available for solution of the nonlinear integer program. This family includes general Monte-Carlo methods, simulated annealing, probabilistic hill climbing, and sequence heuristics. Of these four techniques, the first two are the most completely researched, and require the least *a priori* knowledge of the structure of the optimal solution. Of these two techniques, simulated annealing typically provides better performance.

The simulated annealing algorithm works approximately as follows: First, an initial solution to the optimization problem is selected. This solution is then perturbed. Perturbations of the solution (called candidate solutions) are accepted if the objective function of the optimization is improved. Perturbations which do not improve the solution are accepted on a probabilistic basis. Accepted perturbed/candidate solutions become the solution for the next iteration. The perturbations continue in an iterative fashion until no new perturbed solutions are accepted.

5.4 Application of Simulated Annealing to the Optimization

Simulated annealing is probably the best known and most widely used adaptive heuristic. The algorithm itself was originally proposed by Kirkpatrick [37]. Convergence of the algorithm to a global optimum is guaranteed when

a proper "cooling sequence"[22] is used [38]. However, due to the time required, this is generally not practical (or else one could just explicitly enumerate the entire solution). In simulated annealing, the "acceptance criterion"[23] is based on the Boltzmann distribution. For a review of the basic concepts of simulated annealing and sequence heuristics, Nahar *et al.* [36] provide a good introduction.

5.4.1 Procedure

The simulated annealing algorithm used for the current problem represents an adaptation of the classic Metropolis algorithm [38], with the following notation:

$l_{i,k}$ is the set of $\lambda_{i,w}$ (the value of the log-likelihood ratio threshold i for the data sequence w) at iteration k, with $w \in W$, and $l_{i,k} \subset L$, with L the set of all $l_{i,k}$, $i \in \{0,1\}$

c_k is the "temperature parameter" which regulates the number of candidate solutions examined

The steps in the algorithm are then as follows:

---

[22] The cooling sequence determines how many candidate solutions to the optimization are considered.

[23] The acceptance criteria determines probabilistically when an iterative candidate solution to the optimization problem is accepted.

1. Generate an initial solution $l_{i,0}$. If the objective is to find $l_{0,opt}$, then choose $l_{0,0}=l_{1,opt}$. If the objective is to find $l_{1,opt}$, use a random initial solution. Additionally, calculate an initial value for the temperature parameter (c_0). Ideally c_0 should be such that the acceptance probability is approximately 1:

$$e^{\left(\frac{J(l_{i,k})-J(l_{i,p})}{c_k}\right)} \approx 1 \qquad \textbf{(5-4)}$$

where $J(\cdot)$ is the cost or objective function to be minimized and $l_{i,p}$ is a set of proposed thresholds (obtained from the generating mechanism). The actual criterion was to use a value of 0.9 for the initial acceptance probability, instead of 1, *i.e.*, select c_0, such that:

$$c_0 = \frac{\min_{l_{i,k} \subset L}\{J(l_{i,k})\} - \max_{l_{i,k} \subset L}\{J(l_{i,k})\}}{0.1054} \qquad \textbf{(5-5)}$$

Note that the values of the cost functions are sensitive to changes in any penalty parameters utilized. Section 5.4.2 discusses the cost functions used in the optimization. Also select a value of R, the size of the neighborhood about $l_{i,k}$, defined as:

$$R=\frac{\sum_{w\in W}(2^{\Phi(z_w)}+1)}{8} \qquad \textbf{(5-6)}$$

This value was selected on the basis of experimental results and produced good convergence for s={3,4,5} to a relatively optimal solution. In addition, select a value for the temperature multiplier $\kappa\in[0.8,1.0)$.

2. R perturbations of $l_{i,k}$ are generated by use of the generating function. The following acceptance criterion is then applied:

if $J(l_{i,k})>J(l_{i,p})$ or $c_k^{-1}(J(l_{i,p})-J(l_{i,k}))\leq \ln(U[0,1])$ then set $l_{i,k}=l_{i,p}$

else $l_{i,k}$ remains unchanged.

3. If $l_{i,k}$ has not changed for any of the R perturbations, then the algorithm terminates with $l_{i,opt}=l_{i,k}$. If it has changed in any of the perturbations, the algorithm continues with step 4.

4. Set $c_{k+1}=\kappa c_k$, and let k=k+1; go to step 2.

5.4.2 Cost Function Definition

Penalty functions [35] [39] were used to account for equality and inequality constraints in the optimization problem (*i.e.*, for both the Neyman-Pearson constraints and system related constraints) so that only a single cost function had to be used in the minimization at any one time. The threshold optimizations for the P_F and P_C and Neyman-Pearson constrained problems

were carried out separately, with the P_F Neyman-Pearson constrained problem solved first. The step-wise system cost is then defined as:

$$J(l_{i,k})=\wp_D(l_{i,k})+\rho_F\wp_F(l_{i,k})+\rho_I\wp_I(l_{i,k})+\rho_C\wp_C(l_{i,k})+\rho_{C<D}\wp_{C<D}(l_{i,k}) \qquad \textbf{(5-7)}$$

where $\wp(\ l_{i,k})$ denotes a penalty function for the threshold set $l_{i,k}$. The P_D penalty is:

$$\wp_D(l_{i,k})=\begin{cases}-P_{DC}(l_{i,k}), & \text{when optimizing for } \lambda_{0,w}\\ -P_D(l_{i,k}), & \text{when optimizing for } \lambda_{1,w}\end{cases} \qquad \textbf{(5-8)}$$

where $P_D(l_{i,k})$ is the probability of detection and $P_{DC}(l_{i,k})$ is the probability of detection for a cue at the fusion center for the threshold set $l_{i,k}$. The P_F penalty is:

$$\wp_F(l_{i,k})=\begin{cases}\dfrac{P_F(l_{i,k})}{P_{Fdesired}}, & P_F(l_{i,k})\geq P_{Fdesired}\\ \dfrac{P_{Fdesired}}{P_F(l_{i,k})}, & P_F(l_{i,k})<P_{Fdesired}\end{cases} \qquad \textbf{(5-9)}$$

where $P_F(l_{i,k})$ is the P_F for the threshold set $l_{i,k}$, and $P_{Fdesired}$ is the desired probability of false alarm. Note that the P_F penalty function has a minimum when the actual P_F is closest to the desired P_F, so the optimization is not strictly Neyman-Pearson constrained. However, the form of equations (5-8), (5-9) will generally ensure that fewer data sequences are ignored. Additionally, it reduces the sensitivity of the optimization to changes in the

penalty weights (i.e. it prevents convergence to a global minimum that is far from $P_{Fdesired}$, but minimizes the objective function). The same is true for the cue penalty function. Note that in some applications, it may be desirable to use a hard Neyman-Pearson constraint. The costs for the inequality constraints are defined as:

$$\rho_I(l_{i,k}) = \mathbb{C}(\{\lambda_{i,w} : \lambda_{i,w} \in l_{i,k},\ w \in W,\ \lambda_{i,w} = \beta\}) \tag{5-10}$$

$$\rho_{C<D}(l_{i,k}) = \begin{cases} \mathbb{C}(\{\lambda_{0,w} : \lambda_{0,w} > \lambda_{1,w},\ w \in W,\ \lambda_{i,w} \in l_{i,k}\}), & \text{when optimizing for } \lambda_{0,w} \\ 0, & \text{when optimizing for } \lambda_{1,w} \end{cases} \tag{5-11}$$

The cue penalty is defined as:

$$\rho_C(l_{i,k}) = \begin{cases} \dfrac{P_C(l_{i,k})}{P_{Cdesired}}, & P_C(l_{i,k}) \geq P_{Cdesired} \\ \dfrac{P_{Cdesired}}{P_C(l_{i,k})}, & P_C(l_{i,k}) < P_{Cdesired} \end{cases} \tag{5-12}$$

with $P_{Cdesired}$ being the desired probability of cue at the fusion center.

The penalty weights must be selected with a certain amount of care, and they must reflect the relative importance of the various constraints placed on the system. With this in mind, the following weights were selected for the instances in which s={3,4,5}:

$$\rho_F=5 \quad \rho_{C<D}=8000$$
$$\rho_I=3 \quad \rho_C=5 \tag{5-13}$$

Note that $\rho_{C<D}$ is a large number, which heavily reinforces the constraint that cue thresholds must be less than or equal to detection thresholds. Note also that the lowest penalty multiplier is placed on the constraint that the threshold for any data sequence be finite; *i.e.,* no data sequence is ignored. This is necessary because the Neyman-Pearson constraints placed on the system may force the exclusion of certain combinations of tests (their $P_{F,w}$ may be too high), which is especially true for groups of feature measurements with small $\mathfrak{C}(z_w)$.

Given the previously discussed penalty function formulation, the optimization problem can be expressed as follows:

Minimize:

$$J(l_{i,k})=\wp_D(l_{i,k})+\rho_F\wp_F(l_{i,k})+\rho_I\wp_I(l_{i,k})+\rho_C\wp_C(l_{i,k})+\rho_{C<D}\wp_{C<D}(l_{i,k}) \tag{5-14}$$

Subject to:

$$\lambda_{i,w}\in l_{i,k},\ l_{i,k}\subset L,\ w\in W \tag{5-15}$$

The simulated annealing algorithm can now be applied to the problem to determine the set of optimal thresholds $l_{i,opt}$.

5.4.3 Generating Functions

The selection of a generation method usually receives little attention in papers that address the simulated annealing technique, and thus the selection is determined partly by intuition. The only requirement is that any neighborhood generation function must be symmetric; that is, it must allow the algorithm to move from one solution to another and then back again. In this way, the algorithm avoids getting caught in local minima. Therefore, the following generating method is proposed:

Generate the number of perturbations specified in equation (5-6) of $l_{i,k}$ for each c_k, with each perturbation consisting of a change of up to $\mathfrak{S}(W)$ thresholds $\lambda_{i,w}$, with each $\lambda_{i,w}$ taking on any value in its domain (including $\mathfrak{G}$) for which the value taken is determined by an appropriately scaled U[0,1] random variable.

Chapter 6: Results

6.1 Simulation for the Track Initiation Method

Simulated or real data are necessary for the effective evaluation of the track initiation method. The unavailability of actual low elevation concurrent radar and infrared data necessitates the use of simulated data. Section 6.7.1 discusses a sensitivity analysis that is used to determine the effect of deviations from assumptions incorporated in the simulation.

Figure 6.1 is a block diagram of the simulation, which is comprised of approximately seven different modules, only one of which actually implements the algorithm [40]. One module is responsible for initializing the others (*i.e.*, for defining global variables and initializing sensor parameters). Two simulated sensors are incorporated as well: one infrared sensor with a fixed scan rate, and one radar system with the ability to vary it's azimuth scan rate. A target generator calculates signatures and trajectories for the insertion of a target into the field of view. A sensor resource manager is also incorporated that generates a dwell schedule for the radar and scan extent commands for the infrared sensor. Also note that an interface for a tracker correlator has been incorporated, although the tracker correlator itself will not be implemented for the purposes of this research. Note that the structure of the

simulation facilitates the use of real sensor data when it becomes available (with the limitation that sensor cueing cannot be evaluated).

The sensor models do not operate at the signal processing level. Instead, data are generated using the expected target characteristics (*e.g.* trajectory, radar cross section and speed), propagation conditions, sensor false alarm rate and operating characteristics. These factors are used to generate a probability of detection for the target at every point in its trajectory. Uniform random deviates are used to determine at which points the target will be detected. These results are combined with simulated clutter returns based on a spatially uniform model. The result is a combination of false alarms and valid target detections. Appendix C contains the sensor parameters used in the models.

6.2 Simulation Data Sets and Baseline Method (AND)

Five possible detection scenarios were used for the purpose of evaluating the performance of the multisensor track initiation (MSTI) technique:

1. High speed target (mach 2) with the radar detecting first and infrared sensor detecting second.
2. High speed target (mach 2) with the infrared sensor detecting first, and radar detecting second.
3. High speed target (mach 2) with the radar and infrared sensor detecting at approximately the same time.

4. Low speed target (mach 0.8) with the radar detecting first, and infrared sensor detecting second.

5. Low speed target (mach 0.8) with the infrared sensor detecting first, and radar detecting second.

Because of the detection characteristics of the sensors, it was extremely difficult to guarantee a situation where both sensors detected at approximately the same time for the low speed target. Therefore, a low speed target simultaneous detection scenario was not examined. Scenario 3 should provide a good indication of the effectiveness of the multisensor track initiation (MSTI) method in cases where there is simultaneous detection.

The AND algorithm was selected as the baseline for evaluation of the performance of the MSTI method. This method was selected over others for the following reasons: First, any candidate baseline method must be capable of accepting high false alarm rate data in, with an output that has a relatively low false alarm rate. If this were not the case, the track processor would be confronted with an extremely high number of false track initiations, thus resulting in an overload of the track processor and a possible suppression of target tracks. Second, the selected baseline method must be relatively easy to implement. Third, the method must be suitable for the task at hand: track initiation on multi-sensor data. The AND algorithm satisfies all the aforementioned criteria. Another alternative that may come to mind is the 'OR'

algorithm. However, the 'OR' method is not suitable for this application because the false alarm rate out of the algorithm is greater than the false alarm rate in.

Additionally, it should be noted that once the false alarm performance of the baseline and MSTI methods are shown to be approximately the same, it is then possible to compare detection performance.

6.3 Gate Size

Azimuth gate sizes were determined for two cases, when the maximum sequence length was 3 (s=3) and when s=5. For the s=3 case, the time window will be 6 seconds (from Appendix C we see that the radar scan rate of 2 seconds. drives this; (3 measurements) * (worst case 2 s./measurement) = 6 s. time window), and the spatial window will be 1.28°, from Appendix A.

We see that for the s=5 case, the time length of the window should be 10 seconds. However, this results in a spatial window size in excess of 8°, which is large enough so that there is little difference in the relative frequencies of occurrence $P(w|H_0)$ and $P(w|H_1)$, which diminishes the attainable system probability of detection (P_D). We can then apply the 3/5 majority rule (Section 2.2), and use a 6 second time window with the corresponding azimuth window size (1.28°).

6.4 Feature Tests

Using the feature probability density functions (PDFs) discussed in Chapter 3, the values shown in Table 6-i were calculated using the minimum probability of error criterion and one-dimensional exhaustive searches of all possible solutions for the spatial thresholds. Median data pair separation times were used to determine time dependent target PDF parameters.

6.5 Relative Frequency of Occurrence

Values of $P(w|H_i)$ were obtained by averaging the results for one run each of data runs 1-3; *i.e.*, the average was not taken over the entire data set. These results are shown in Figure 6.2 for s=3 and in Figure 6.3 for s=5, respectively. These plots show the relative separation of the two conditional relative frequency of occurrence distributions and are indexed to the decimal representation of the data sequence. The optimization algorithm tends to lower the log-likelihood ratio threshold for those data sets that have $P(w|H_1)>>P(w|H_0)$, and raise it for those log-likelihood ratio thresholds where $P(w|H_0)>>P(w|H_1)$.

6.6 Optimization Results

The results of the simulated annealing optimization for s=3 are shown in Figure 6.4, with Figure 6.5 showing the results for s=5.

Absolute interpretations of the optimization results are difficult to formulate. However, general observations about the results are possible: First, we see that the optimization for s=3 resulted in data sequences 1,2,3,4, 7 and 10 being ignored from a declaration standpoint. The reason for this becomes clear when Figure 6.2 is examined; $P(w \mid H_0) > P(w \mid H_1)$ for these data sequences, and the maximum value of the log-likelihood ratio is somewhat low. Second, we see that similar results are evident for the s=5 optimization. In Figure 6.5 we see that data sequence 28 has the lowest declaration threshold, and that $P(w=28 \mid H_1) > P(w=28 \mid H_0)$, with data sequence 28 contributing the most to the probability mass in $P(w \mid H_1)$.

Note how this threshold assignment differs from the k-out-of-N criteria where, in general, a declaration decision is based solely on the number of data points present in the data sequence. For example, for the s=5 case, the k-out-of-N criteria would have 6 possible data sequences (0-out-of-5, 1-out-of-5, ... , 5-out-of-5 data points in the gated region), and the probabilities of each event occurring would be based on the binomial distribution, with the probability of a data point occurring anywhere in the gated region conditional on H_0 and H_1 being constants.

6.7 Track Initiation Results

Eleven runs each of five different sensor/target configurations were performed.

The baseline with which the algorithm was compared was a simple AND algorithm, which declares when both sensors detect the target within a small spatiotemporal time window (azimuth size of 0.2312°, over 0.5 seconds[24]). The dimensions of the window were consistent with the model proposed in Appendix A. The AND algorithm has been proposed by many as a method of fusing data from multiple sensors [22], [41]. The AND algorithm had a median false alarm rate of 2 over 120 seconds and 360°. The MSTI algorithm had a median false alarm rate of 1 over 120 seconds and 360°. The median false cue rate was 1 per second over 360°. Table 6-ii shows the median and S_3 time to first cue - that is, the time to the first request for a sensor cue issued by the algorithm as a result of target data. The S_3 norm is a robust measure of the scale of a distribution and is sometimes referred to as the normalized median absolute deviation from the median, *i.e.*,

$$S_3(z) = 1.4826 * \underset{i}{\text{med}} \left| z_i - \underset{j}{\text{med}}(z_j) \right| \qquad \textbf{(6-1)}$$

where z is the data set. Note that there is no provision in the AND algorithm for a bilevel threshold, which would enable cueing. Table 6-iii shows the

[24] A maximum of 0.5 seconds separates data from consecutive scans of the two sensors.

median and S_3 time to first declare, which is the time to the first valid declaration of a target being present. Table 6-iv shows the median and S_3 time to the first 3-D declare; that is, the time at which an algorithm declares the target, and does so on the basis of data from both sensors. All times are measured from the start of the simulation. Robust statistics were used because of the occasionally wide dispersion of data points, and the relatively small data sets involved. This is illustrated by Figure 6.6, where frequency distributions of the median and S_3 times to first 3-D declare and first cue are shown for data set 1. The median was used to estimate the parameter of location while the S_3 norm was used as an estimate of the scale of the distribution.

Figure 6.7 shows an example of the data being operated on by the MSTI and AND algorithms, data set 2 (high-speed target) is being processed in this example. Figure 6.8 illustrates the output of the AND algorithm for this data. The output shown in this and subsequent figures consists of all those data which give rise to a target declaration. Here we see that the first valid target declaration takes place at approximately 17 seconds, with consistent declaration at about 30 seconds. Figure 6.9 shows the cue requests of the MSTI algorithm for s=3. We see that the first cue arising from valid target data is at approximately 9 seconds. Figure 6.10 shows the declaration performance for s=3, where consistent declaration begins at approximately 14 seconds. Note that for the MSTI and AND methods, all data giving rise to a declaration or

cue are plotted. Thus, because of the windowing operation that takes place, it may appear that a declaration or cue appears sooner than is shown from the plots.

Figure 6.11 shows an example of the data set 4 (low-speed target) being operated on by the MSTI and AND algorithms. Figure 6.12 illustrates the output of the AND algorithm for this data. Here we see that the first target declaration takes place at approximately 60 seconds, with consistent declaration at about 90 seconds. Figure 6.13 shows the cue requests of the MSTI algorithm for s=3. We see that the first cue arising from valid target data is at approximately 41 seconds. Figure 6.14 shows the declaration performance for s=3, where consistent declaration begins at approximately 60 seconds.

It is evident from Table 6-iii and Table 6-iv that both the MSTI and AND algorithms are effective against the high speed threat (data sets 1-3) and the low speed threat (data sets 4 and 5). Table 6-iii shows that the median declaration performance of the MSTI algorithm was equal to or better than that of the AND algorithm for all the data sets. For data sets where one sensor is working better than the other (data sets 1,2,4 and 5), we see that the improvement in declaration time can be large, up to 28 seconds for data set 4. Table 6-iv shows that the 3-D declaration performance of the MSTI and AND algorithms is roughly equivalent for all data sets. The latter is to be expected;

the AND algorithm represents the best performance we can expect when both sensors declare at the same time, if a declaration is based on two measurements only[25].

It becomes apparent from examining Table 6-ii, Table 6-iii and Table 6-iv that the performance of the MSTI algorithm with s=3 is slightly better than with s=5. This is because of the different data requirements for the two cases. First we note that the spatiotemporal window size for both cases is the same. This means that the same data are being processed by both algorithms. Given that this is the case, the s=3 case will require fewer data to declare than the s=5 case (*i.e.*, a maximum of 3 data points for s=3, and a maximum of 5 data points for s=5). Thus, because fewer data are required, the detection time is shorter for the s=3 case. A severely clutter-limited environment will necessitate the use of a longer sequence length (due to additional clutter passing the gating criteria), and thus increase the declaration time.

Table 6-ii, Table 6-iii and Table 6-iv indicate a possibly significant performance advantage to using the MSTI algorithm. We saw previously in this section that while both the MSTI and AND algorithms provide

---

[25] The AND algorithm only requires confirming detections on two consecutive scans, and thus a declaration will result when the second detection is received. Conversely, the MSTI method requires a minimum of two measurements - and possibly more - depending on which data sequences are ignored as a result of the threshold optimization.

approximately equivalent clutter suppression (*i.e.*, the false alarm rates are approximately equal out of both algorithms, see section), the detection performance of the MSTI method can be better[26] than the AND algorithm for cases in which one sensor works better than the other (which may constitute most scenarios). This is because while 3-D declaration performance is important from the standpoint of a fire control system, so is 2-D declaration performance, *i.e.*, even though the tracker may not be able to hand-off to a fire control system immediately, it can still maintain a track history on the target using a 2-D track. When a 3-D track is initiated, the tracker can use previous filter state estimates to arrive at a more accurate track.

6.7.1 Sensitivity Analysis

A sensitivity analysis was conducted to measure the effects of deviations from assumptions made in the MSTI method. The following deviations were tested:

1. Incorrect values for the feature test statistics (P_{Di} and P_{Fi}) - Deviation from assumed values may occur when there is correlated clutter or nonuniform spatial noise, or when assumed values for feature PDFs are not accurate.

---

[26] From a detection time standpoint, the median performance of the MSTI algorithm is better than that of the AND; however, the performance of the AND algorithm may have at times been better than that of the MSTI algorithm on a per-data run basis.

2. Incorrect values for $P(w \mid H_i)$ - Deviation from assumed values may occur when there is correlated clutter.

The 'most used feature' test was used to test the sensitivity of the method to a type 1 deviation. We see from equations (4-13)-(4-16) that if the relative frequency of occurrence of all data sequences were equal, then a change in the statistics of any feature test would have the same effect on the probability mass. However, for the problem being considered, some feature tests do occur more often than others. We see then that the specific test that constitutes the majority of the tests used in each w will have the greatest effect on the probability mass.

For the infrared-radar low-elevation problem, this test is the infrared-radar azimuth feature test, which we see from Table 6-v. Here a 0 in the data vector represents an infrared data point, and a 1 represents a radar data point. A similar analysis for s=5 produces the same result. We see from Table 6-i that the original statistics of the test were: P_{Dtest}=0.9992 and P_{Ftest}=0.1542. There are many ways in which we can perturb these values. Two types of perturbations to the P_{Di} and P_{Fi} were considered. The first was small scale (values for P_{Di} and P_{Fi} change less than 10%), and the second was large scale (values for P_{Di} change by up to a factor of 2 and values for P_{Fi} change by up to a factor of 10). A factor of 10 was used for the P_{Fi} because we expect that the P_F will change more drastically in response to nonstationary clutter

fluctuations and is usually set to a fairly low value to begin with. We can then define a number of possible variations in the P_{Di}, P_{Fi}, and the relative frequency of occurrence of the data sequences. The types of P_{Di} and P_{Fi} variations tested are shown in Table 6-vi.

For deviations of type 2, the relative frequency of occurrences $P(w|H_0)$ and $P(w|H_1)$ were randomly perturbed in an iterative fashion until a χ^2 goodness-of-fit test indicated that the distributions of the assumed and perturbed relative frequency of occurrences were the same at a less than 10% level of significance. Only small scale sensitivity tests were conducted, because it will be shown that the MSTI method is moderately sensitive to small scale perturbations in the relative frequency of occurrences.

6.7.2 Sensitivity Analysis Results

Table 6-vii shows the cueing performance of the MSTI algorithm for the s=3 and s=5 cases for perturbations of type 1 and 2. From Table 6-vii and Table 6-ii, we see that the performance of the MSTI algorithm is relatively insensitive (the majority of times changed by less than 10%) to variations of type 1a and 1b, and in some cases performance improved in excess of 10% (data sets 2 and 3)[27]. The possible exception is data set 4, where performance

[27] This is a consequence of using perturbed values for the optimization. It should also be noted that the increase in detection performance was in many instances accompanied by an increase in the false alarm rate.

was adversely affected to a high degree. Both the s=3 and s=5 performance changed by over 10% in response to deviations of type 1c. In summary, we see that the cueing performance of the MSTI algorithm for both s=3 and s=5 is relatively insensitive to small scale changes of type 1, and somewhat sensitive (the majority of times changed over 10%) to large scale changes of type 1. For changes of type 2, we note cases where performance changed over 10% in both directions. For some types of perturbations, we note an increase in the S_3 norm of the cueing time, which indicates increased variability of performance.

Table 6-viii shows the declaration performance of the MSTI algorithm, with the type 1 and 2 perturbations, which we can compare to the results of Table 6-iii. Two things are apparent from the data: First, that performance generally changes less than 10% in response to deviations of type 1. Second, that in most cases, the declaration performance of the MSTI algorithm for type 1 and 2 perturbations meets or exceeds that of the AND.

Table 6-ix shows the 3-D declaration performance of the MSTI algorithm, which can be compared to Table 6-iv. We see once again, that in most cases, performance changes less than 10% in response to deviations of type 1 or 2, and that at all times, the 3-D declaration performance of the MSTI algorithm was within 10% of the AND.

Table 6-x shows the number of false cues out of the simulation for the different types of perturbations. It should be noted that data are not presented

for false declarations because virtually no increase was observed. For the unperturbed case, we expect to see approximately 1 false cue every 18 seconds. For data sets 1,2 and 3, this means approximately 2 false cues per data set. For data sets 4 and 5, this translates into approximately 6 false cues per data set. We see from Table 6-x that, overall, the MSTI algorithm with s=3 is more sensitive from a false cue standpoint than is the MSTI method with s=5. We also see that the MSTI method with s=3 is very sensitive (performance changed in excess of 100% in some cases) to variations in the relative frequency of occurrences, and that in some cases, the false cue rate was a factor of 3 or 4 times the desired value.

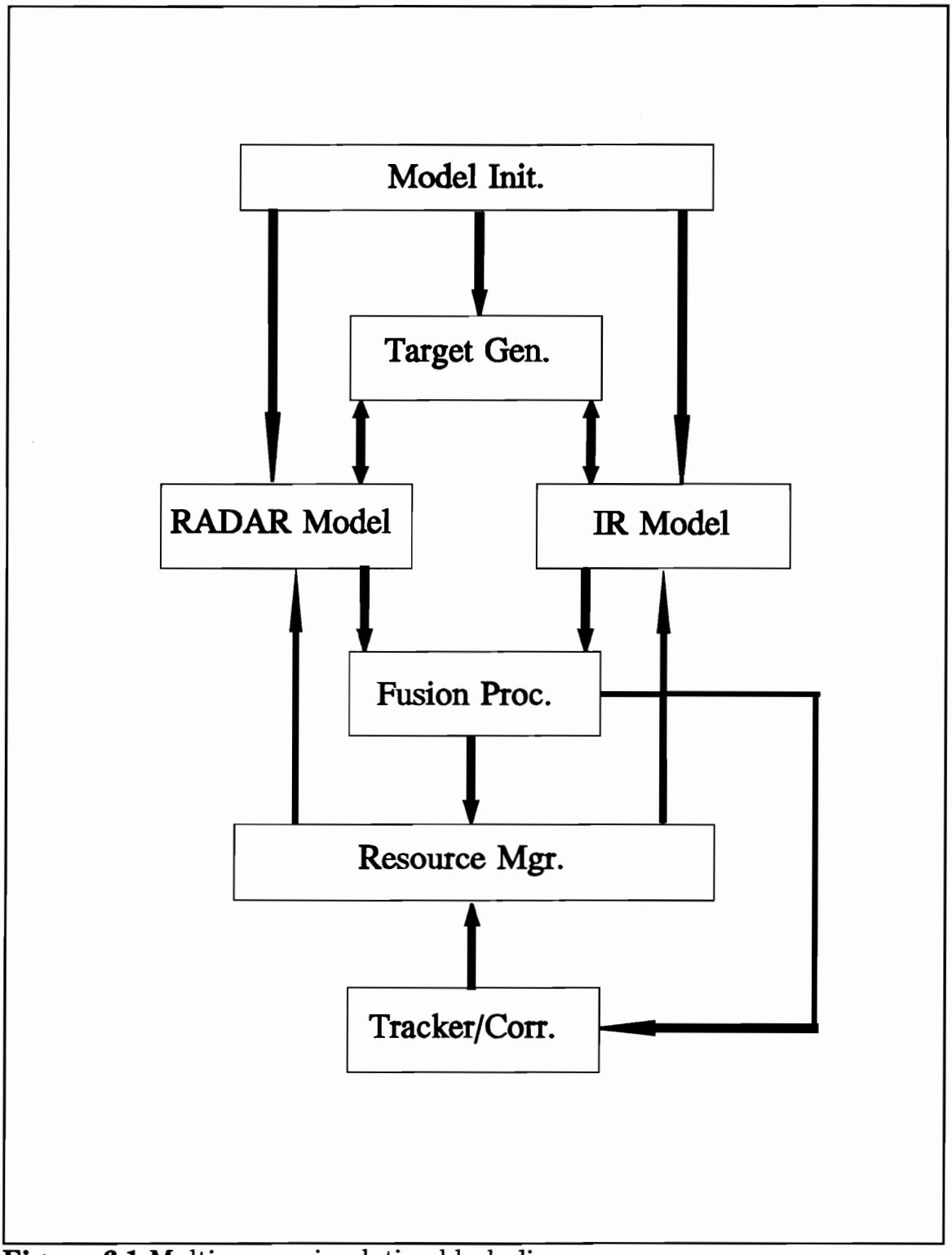


Figure 6.1 Multisensor simulation block diagram

Table 6-i Feature test probability of detection (P_{Di}) and probability of false alarm (P_{Fi})

Test Name	P_{Di}	P_{Fi}	$Threshold_i$
Azimuth (infrared - infrared)	9.9346e-01	5.5422e-02	2.8812e-02°
Azimuth (radar-radar)	9.2743e-01	2.6080e-01	1.5306e-01°
Azimuth (infrared - radar)	9.9922e-01	1.5416e-01	1.0202e-01°
Elevation (infrared - infrared)	9.9941e-01	3.8998e-02	5.7526e-03°
Range/Range Rate Correlation (radar-radar)	9.892e-01	1.6807e-02	2.0914e+02 m.

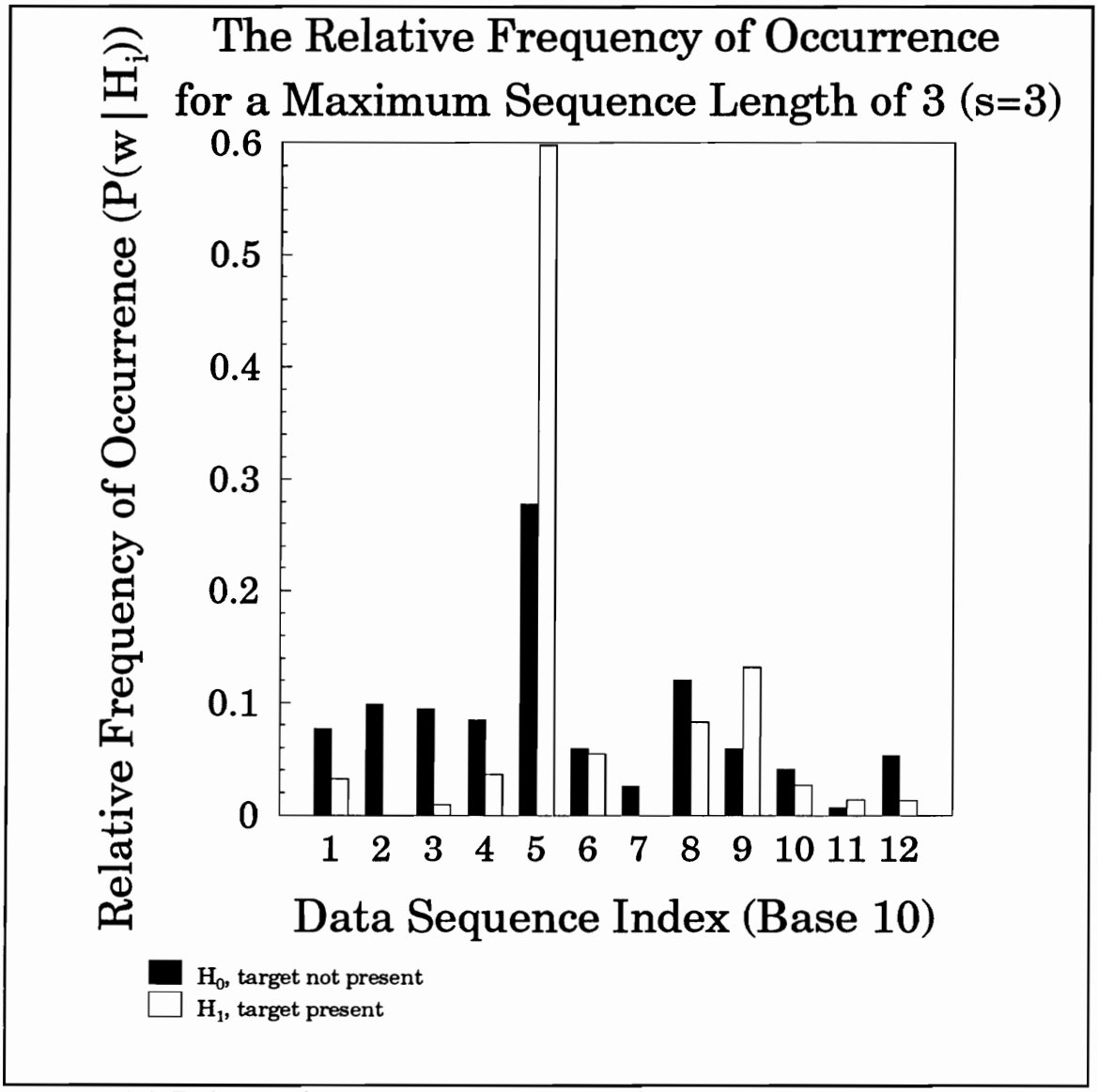


Figure 6.2 Relative frequency of occurence, for a maximum sequence length of 3

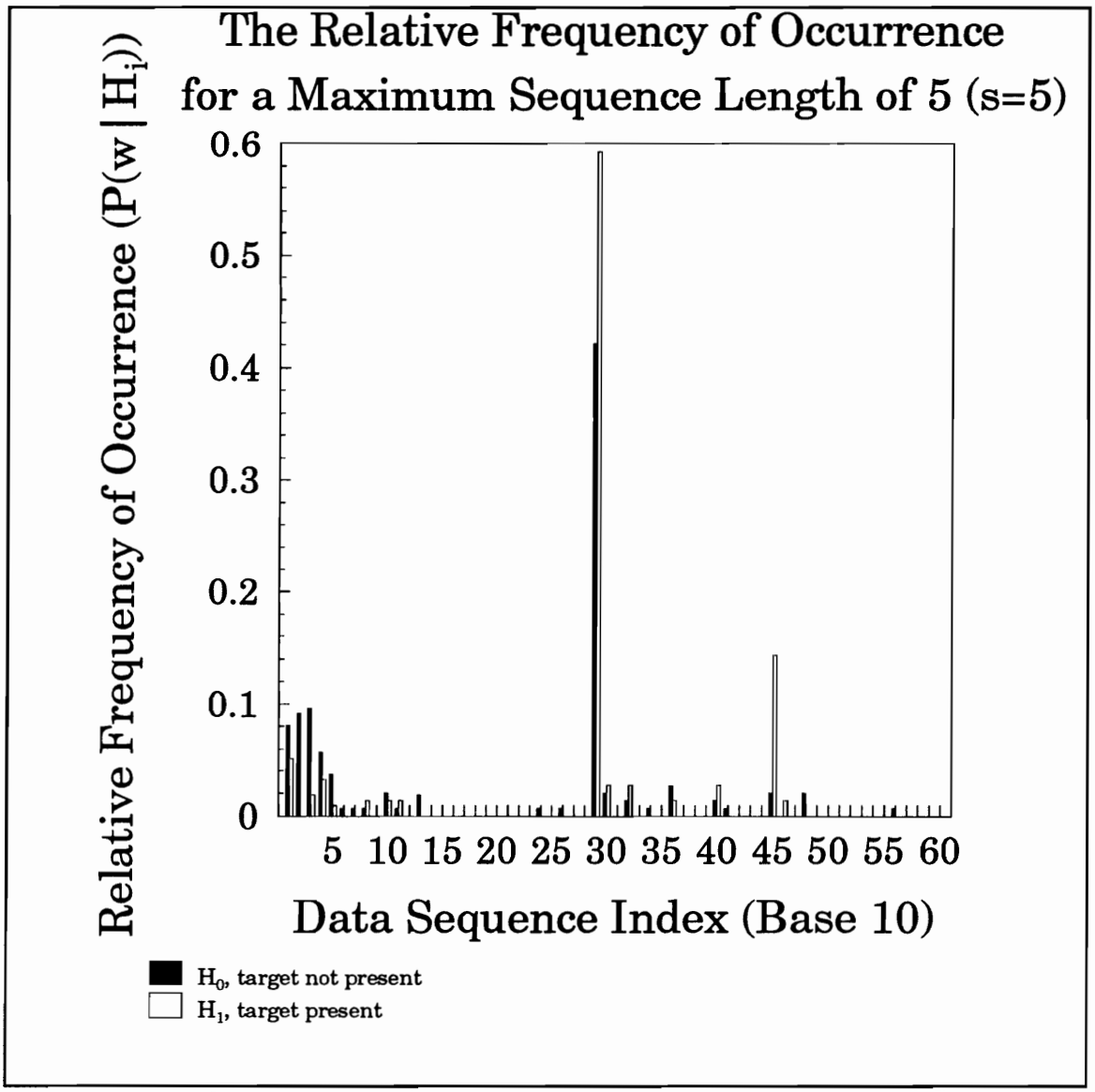


Figure 6.3 Relative frequency of occurrence, for a maximum sequence length of 5

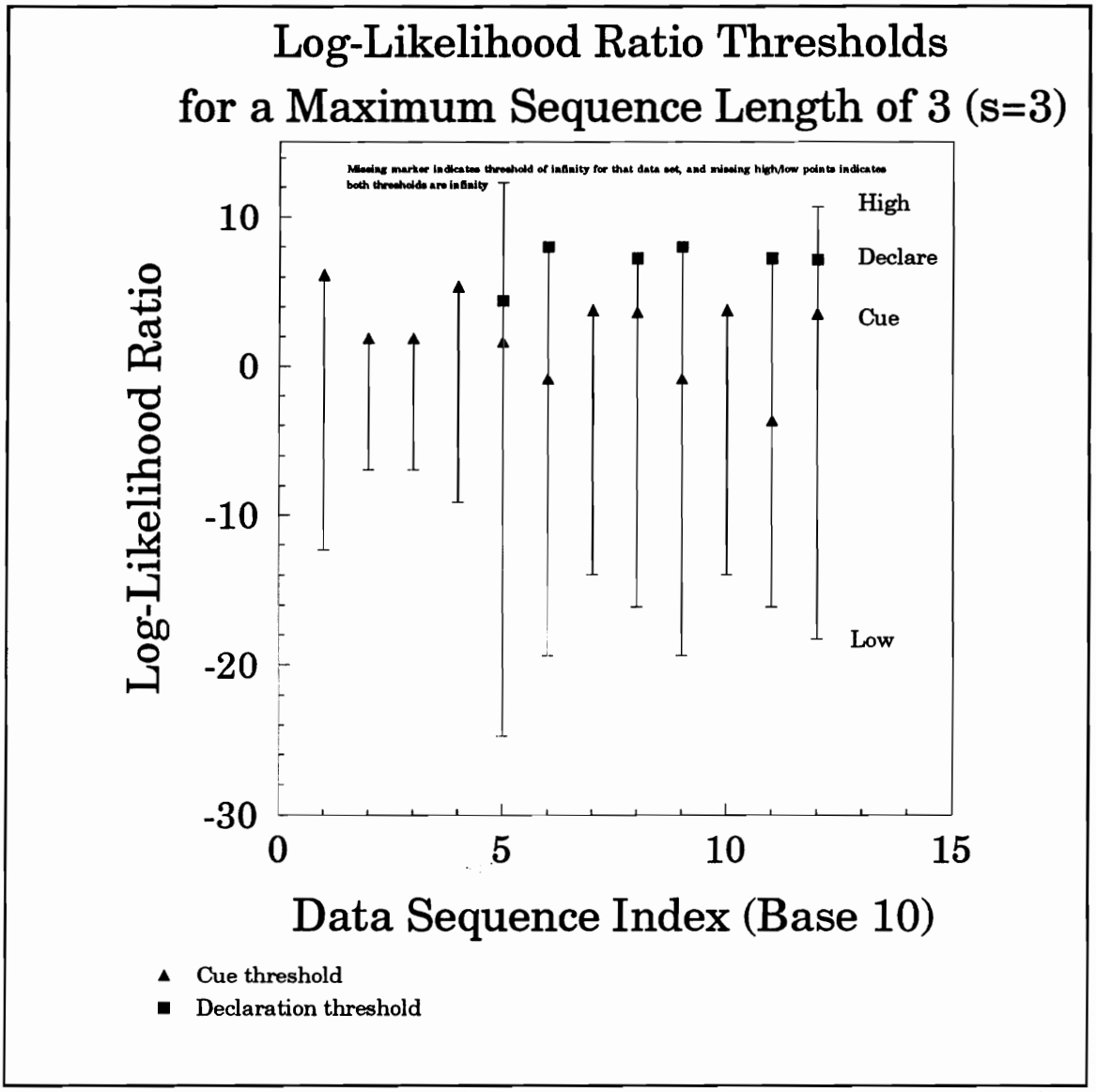


Figure 6.4 Log-likelihood ratio thresholds, for a maximum sequence length of 3

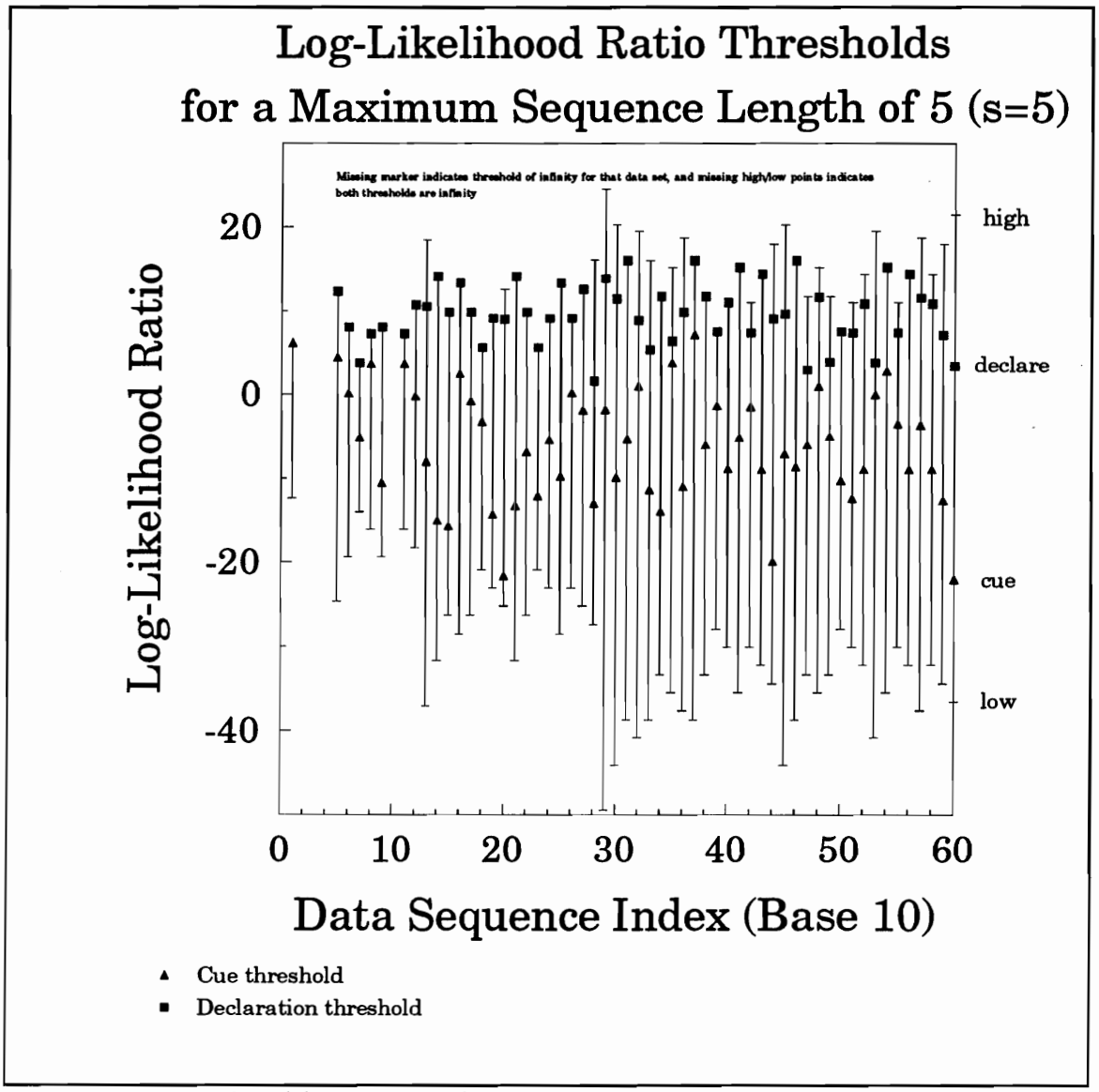


Figure 6.5 Log-likelihood ratio thresholds, for a maximum sequence length of 5

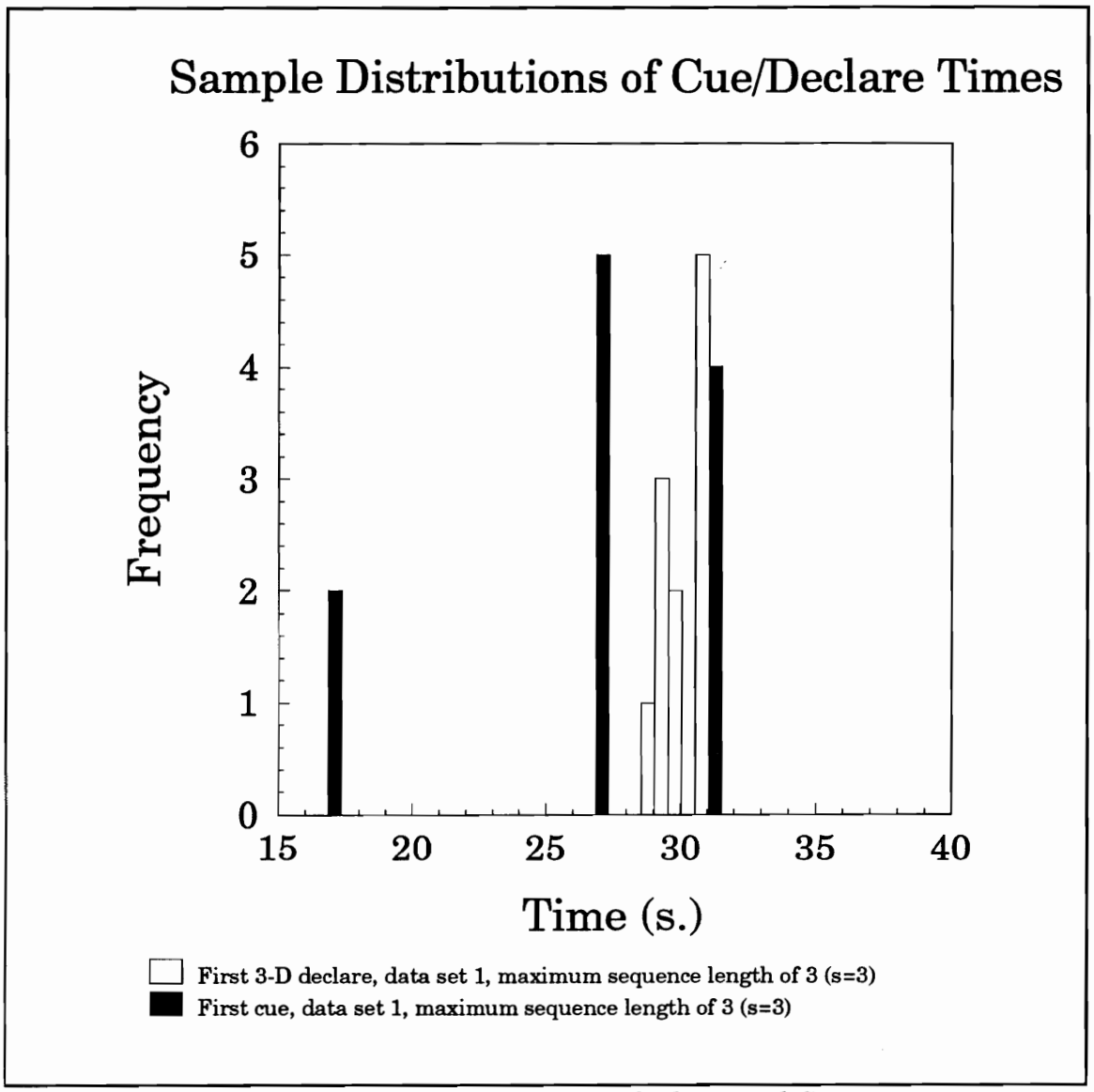


Figure 6.6 Distribution of time to first 3-D declare and first cue

Table 6-ii Median/S_3 time to first cue (s.)

Data set #	Maximum sequence length of 3 (s=3), Median/S_3 time to first cue (s.)	Maximum sequence length of 5 (s=5), Median/S_3 time to first cue (s.)	AND
1	27.1/6.2	29.8/4.9	NA
2	32.1/5.7	9.3/0.0	NA
3	27.1/0.0	33.1/3.0	NA
4	41.1/3.0	43.1/3.0	NA
5	66.3/3.7	66.3/0.7	NA

Table 6-iii Median/S_3 time to first declare (s.)

Data set #	Maximum sequence length of 3 (s=3), Median/S_3 time to first declare (s.)	Maximum sequence length of 5 (s=5), Median/S_3 time to first declare (s.)	AND, Median/S_3 time to first declare (s.)
1	29.1/0.0	30.8/0.5	31.1/0.0
2	9.3/0.0	13.8/0.0	17.1/3.0
3	29.1/3.0	29.1/3.0	31.3/0.0
4	59.1/0.5	59.1/0.5	87.1/0.3
5	66.8/0.7	66.8/0.7	69.1/0.0

Table 6-iv Median/S_3 time to first 3-D declare

Data set #	Maximum sequence length of 3 (s=3), Median/S_3 time to first 3-D declare (s.)	Maximum sequence length of 5 (s=5), Median/S_3 time to first 3-D declare (s.)	AND, Median/S_3 time to first 3-D declare (s.)
1	29.8/1.5	30.8/0.7	31.1/0.0
2	17.1/3.0	17.1/3.0	17.1/3.0
3	31.3/0.0	31.3/0.7	31.3/0.0
4	87.1/2.7	87.3/0.3	87.1/0.3
5	69.1/0.0	69.1/0.0	69.1/0.0

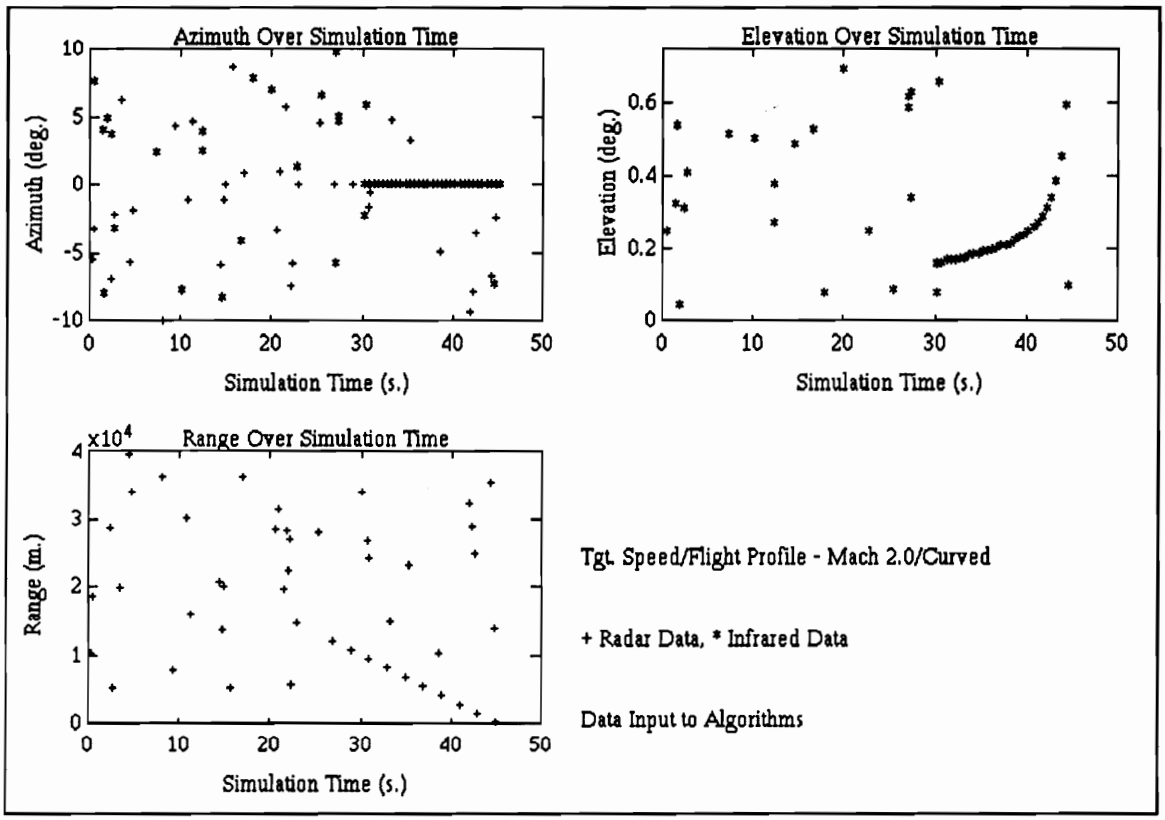


Figure 6.7 Example for data set 2, raw data input to algorithms

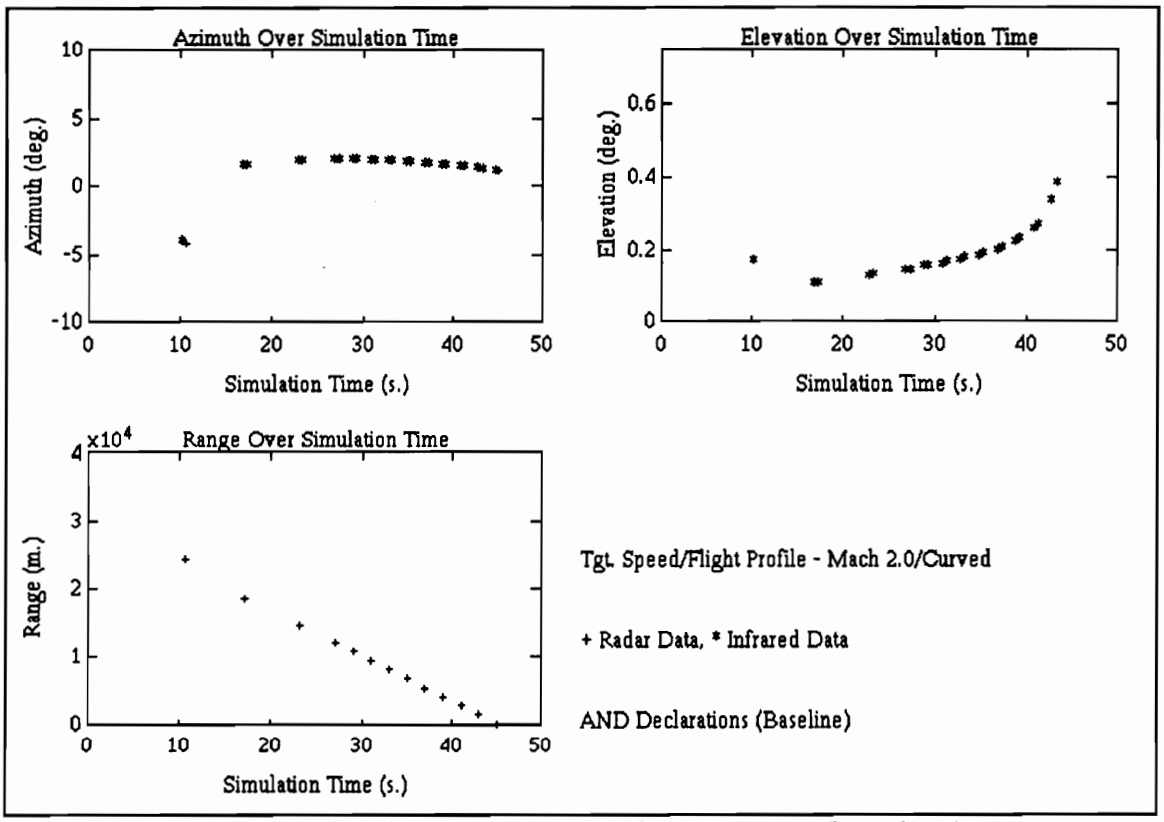


Figure 6.8 Example for data set 2, AND declarations (baseline)

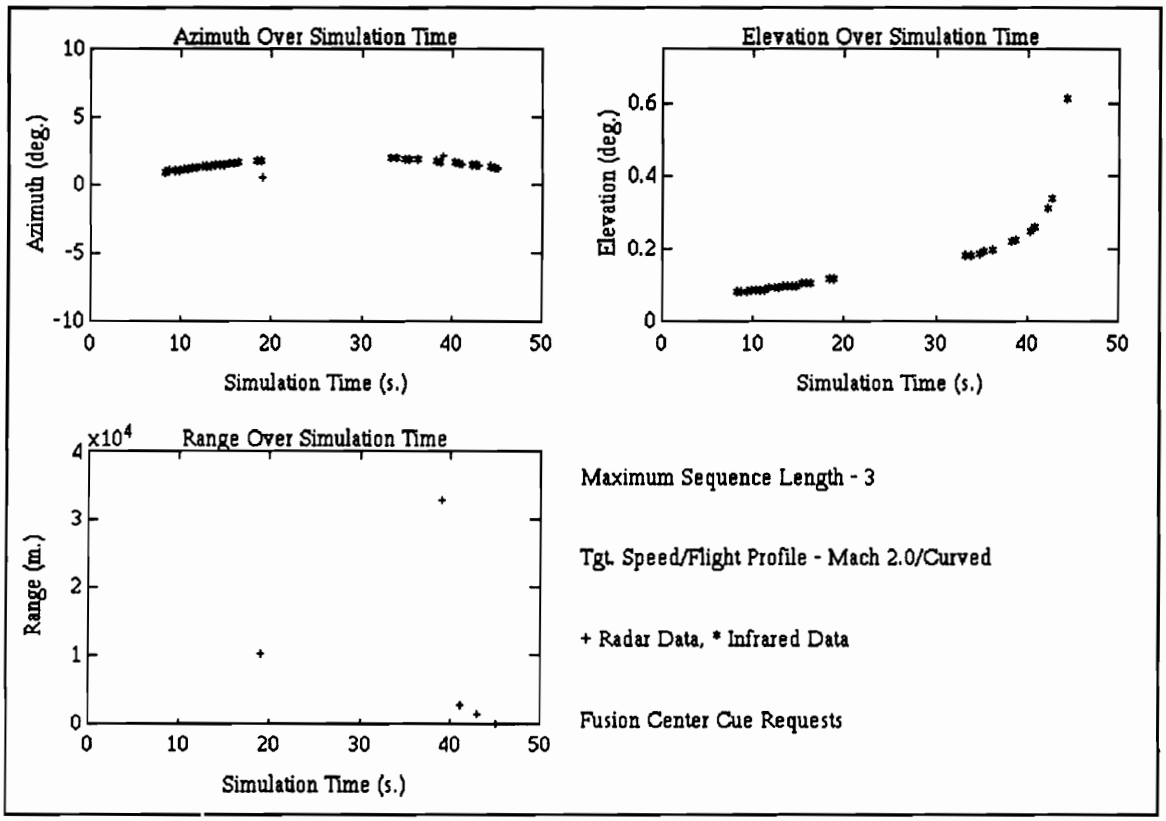


Figure 6.9 Example for data set 2, cues

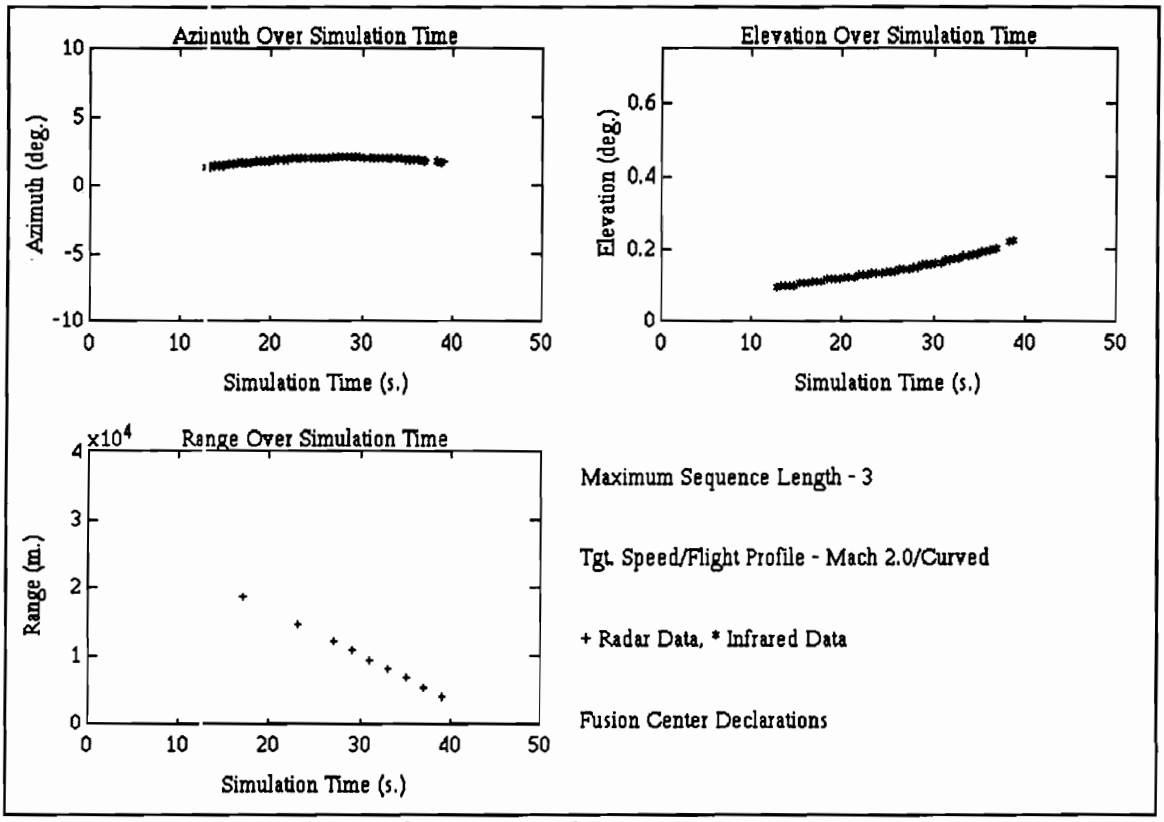


Figure 6.10 Example for data set 2, declarations

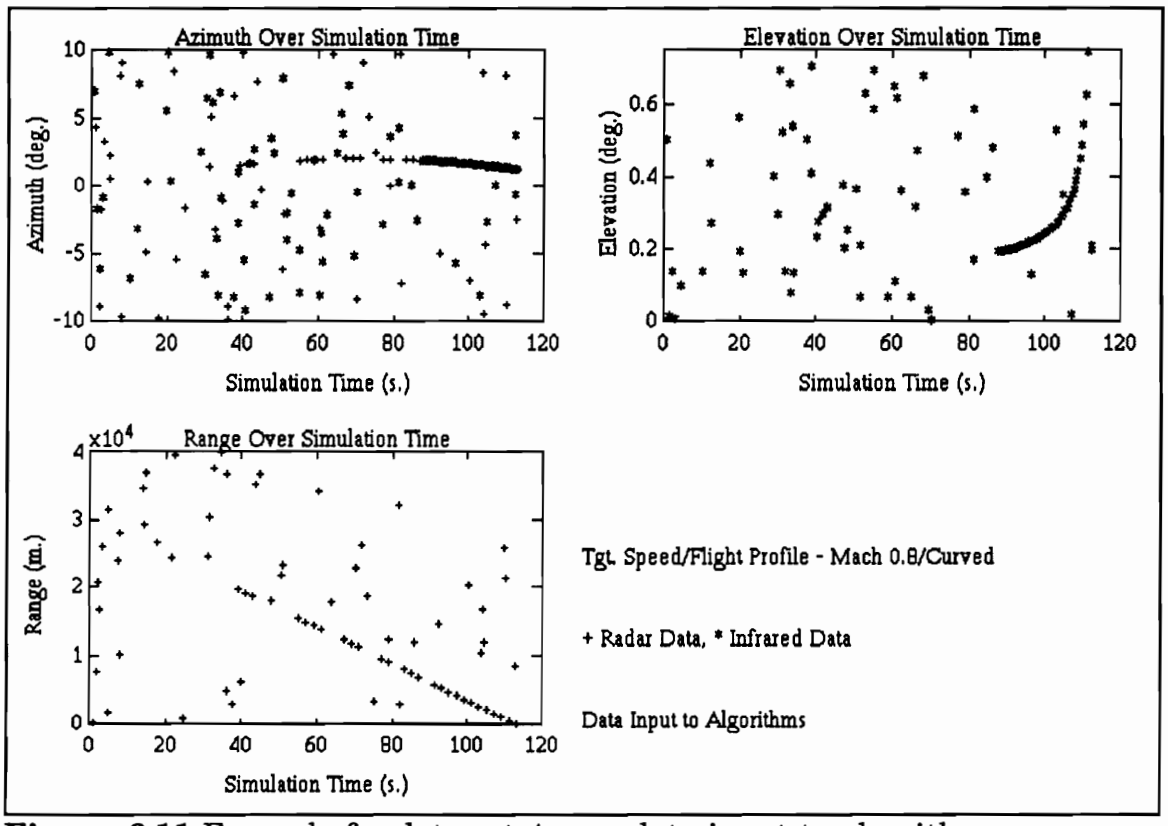


Figure 6.11 Example for data set 4, raw data input to algorithms

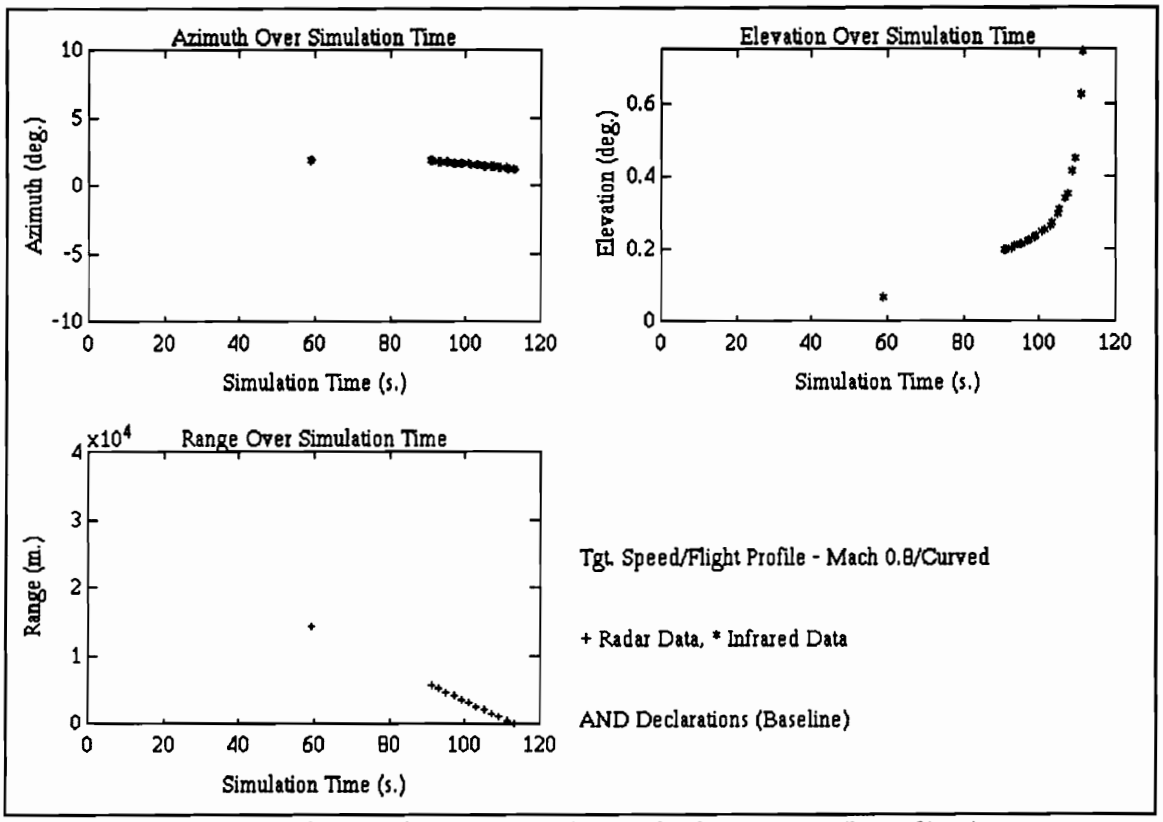


Figure 6.12 Example for data set 4, AND declarations (baseline)

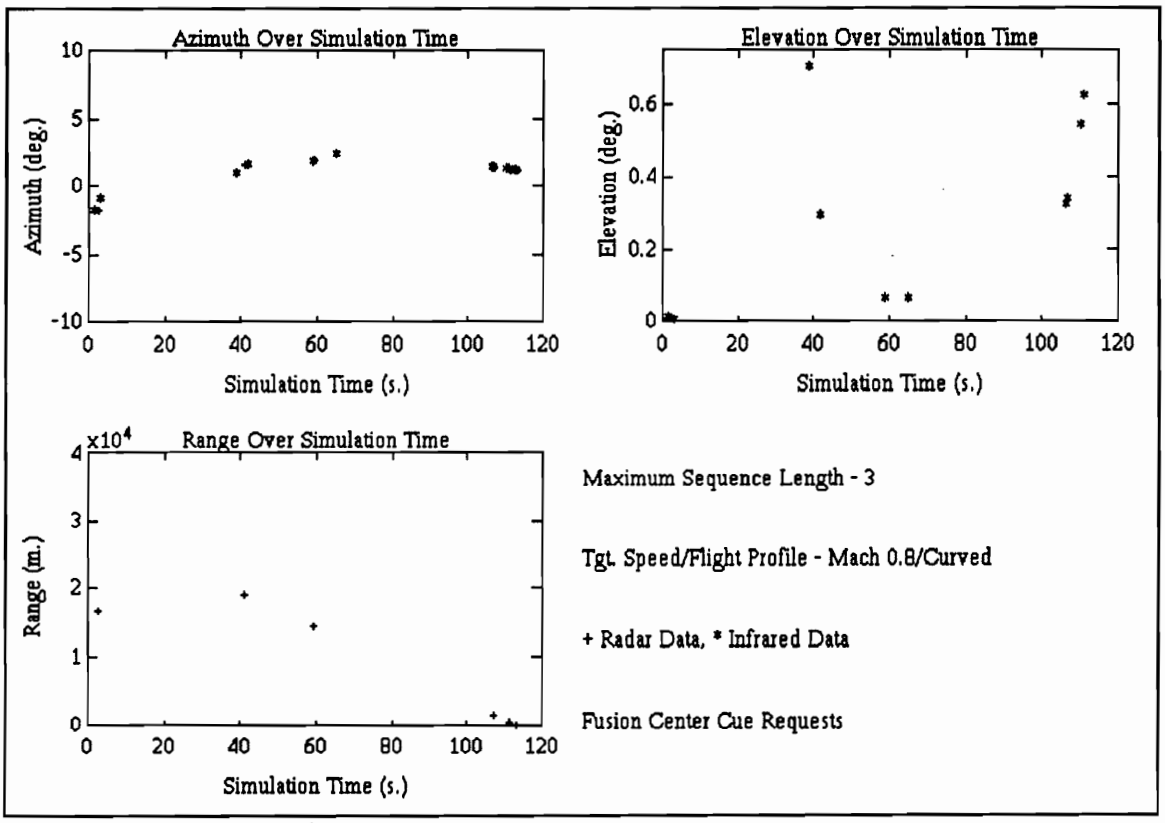


Figure 6.13 Example for data set 4, cues

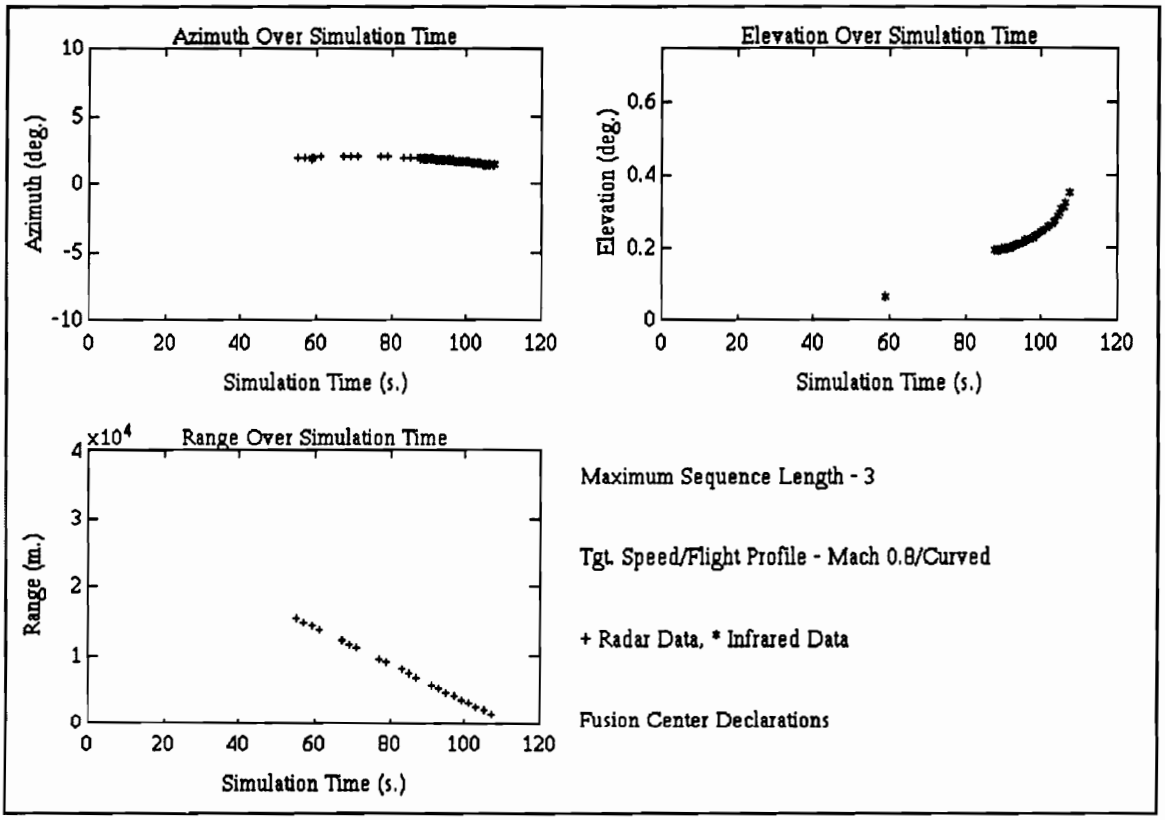


Figure 6.14 Example for data set 4, declarations

Table 6-v Data pair breakdown for maximum sequence length of 3 (s=3) case

Binary data vector (s=3)	# Consecutive infrared - infrared data pairs	# Consecutive infrared -radar data pairs	# Consecutive radar-radar data pairs
0 0	1	0	0
0 1	0	1	0
1 0	0	1	0
1 1	0	0	1
0 0 0	2	0	0
0 0 1	1	1	0
0 1 0	0	2	0
0 1 1	0	1	1
1 0 0	1	1	0
1 0 1	0	2	0
1 1 0	0	1	1
1 1 1	0	0	2
Total	5	10	5

This table illustrates the number of consecutive data pairs for every possible data sequence of maximum length 3 (s=3).

Table 6-vi Types of perturbations used for sensitivity analysis

Perturbation type	Perturbed probability of detection for the feature test i (P_{Di})	Perturbed probability of false alarm for the feature test i (P_{Fi})
1a (small scale)	1.0000	0.1387
1b (small scale)	0.8993	0.1696
1c (large scale)	1.0000	0.0154

Table 6-vii MSTI cueing times for perturbations 1 and 2

Data set #	Perturbation type	Maximum sequence length of 3 (s=3), Median/S_3 time to first cue (s.)	Maximum sequence length of 5 (s=5), Median/S_3 time to first cue (s.)
1	1a	21.3/6.2	29.3/13.3
2	1a	8.8/1.5	10.8/0.0
3	1a	17.1/3.0	23.1/11.9
4	1a	39.1/0.0	67.1/29.2
5	1a	65.8/3.7	66.8/2.6
1	1b	27.1/0.0	27.1/3.0
2	1b	8.3/0.8	9.3/0.8
3	1b	17.1/0.6	27.1/0.0
4	1b	41.1/3.0	43.1/6.0
5	1b	65.8/3.0	65.8/1.5
1	1c	27.1/0.0	16.5/1.0
2	1c	8.9/1.3	-/-
3	1c	31.1/0.3	23.1/0.0
4	1c	69.1/17.8	67.1/29.1
5	1c	74.1/13.0	105.3/0.0
1	2	27.3/1.5	36.8/0.0
2	2	7.3/2.2	36.8/0.0
3	2	23.1/10.4	36.3/0.0
4	2	41.1/3.4	71.1/4.2
5	2	65.8/1.0	66.8/2.2

Table 6-viii MSTI declaration times for perturbations 1 and 2

Data set #	Perturbation	Maximum sequence length of 3 (s=3), Median/S_3 time to first declare (s.)	Maximum sequence length of 5 (s=5), Median/S_3 time to first declare (s.)
1	1a	29.1/0.0	30.8/0.4
2	1a	9.3/0.0	13.8/0.0
3	1a	29.1/3.0	27.1/3.0
4	1a	59.1/3.0	59.1/0.4
5	1a	66.8/0.7	66.3/3.0
1	1b	29.1/1.2	30.8/0.4
2	1b	13.8/0.0	10.8/0.0
3	1b	29.1/3.0	29.1/3.0
4	1b	58.8/0.4	59.1/0.4
5	1b	65.8/1.5	67.8/0.7
1	1c	29.1/0.0	29.3/2.2
2	1c	9.3/0.0	10.8/0.0
3	1c	29.1/0.0	17.1/4.1
4	1c	61.1/0.0	43.1/2.0
5	1c	66.3/0.8	66.8/0.0
1	2	29.1/2.5	30.8/0.4
2	2	9.3/0.0	10.8/0.0
3	2	27.1/6.2	29.1/3.0
4	2	57.1/0.0	59.1/1.2
5	2	66.8/0.7	67.8/1.5

Table 6-ix MSTI 3-D declaration times for perturbations 1 and 2

Data set #	Perturbation	Maximum sequence length of 3 (s=3), Median/S_3 time to first 3-D declare (s.)	Maximum sequence length of 5 (s=5), Median/S_3 time to first 3-D declare (s.)
1	1a	29.8/1.0	30.8/1.0
2	1a	16.9/3.0	17.1/3.0
3	1a	31.1/0.0	31.1/0.7
4	1a	87.1/2.0	87.1/3.0
5	1a	69.1/0.0	69.1/1.0
1	1b	29.8/2.0	30.1/1.1
2	1b	17.3/2.0	17.3/2.0
3	1b	31.3/1.1	31.7/0.7
4	1b	87.5/3.3	87.3/0.0
5	1b	70.1/0.0	70.1/0.0
1	1c	29.8/2.0	30.5/0.6
2	1c	17.1/0.0	17.1/1.0
3	1c	31.1/1.0	31.1/1.0
4	1c	86.1/3.3	87.1/0.0
5	1c	69.1/0.0	66.8/0.0
1	2	29.8/4.0	30.8/4.0
2	2	17.7/1.2	17.7/1.2
3	2	33.1/0.0	31.7/2.3
4	2	90.3/4.0	88.7/1.0
5	2	71.1/0.0	69.1/0.0

Table 6-x MSTI false cues for perturbations 1 and 2

Data set #	Per-turbations	Maximum sequence length of 3 (s=3), Median/S_3 # of false cues	Maximum sequence length of 5 (s=5), Median/S_3 # of false cues	AND
1	1a	3/3.0	1/0.0	NA
2	1a	6/3.0	1/0.0	NA
3	1a	5/4.4	2/3.0	NA
4	1a	8/1.5	5/1.5	NA
5	1a	7/3.0	1/1.5	NA
1	1b	1/0.0	0/0.0	NA
2	1b	0/0.0	0/0.0	NA
3	1b	0/0.0	0/0.0	NA
4	1b	1/1.5	1/0.0	NA
5	1b	1/0.0	0/0.0	NA
1	1c	0/0.0	1/0.0	NA
2	1c	0/0.0	1/0.0	NA
3	1c	0/0.0	2/3.0	NA
4	1c	0/0.0	3/0.0	NA
5	1c	0/0.0	1/1.5	NA
1	2	9/3.0	0/0.0	NA
2	2	6/3.0	0/0.0	NA
3	2	7/3.0	0/0.0	NA
4	2	21/3.0	1/1.5	NA
5	2	17/4.4	0/0.0	NA

Chapter 7: Conclusions and Suggestions for Additional Research

7.1 Conclusions

A multisensor track initiation (MSTI) method has been discussed. The MSTI method formed data sequences composed of sensor detections, and computed spatial feature tests on these sequences. The weighted sum of spatial feature test results was then used for requesting additional sensor measurements (cues) and initiating tracks. The method was applied to the radar-infrared low-elevation track initiation problem. It was shown that the MSTI method accounts for the missing measurement phenomena by statistically characterizing the possible ways in which detections presented from the sensors could occur. It was also shown that the MSTI method was capable of issuing requests for cues or declaring a target and initiating a track utilizing data sequences containing single or multiple sensor detections.

Simulation results for the MSTI algorithm were presented and compared to the AND algorithm in five possible detection scenarios. It was shown for the MSTI algorithm that the median time to declaration/track initiation based on a target data sequence containing radar and infrared detections was at least as good as the AND algorithm. Additionally, it was shown that the MSTI method is capable of declaring and initiating a track more quickly than the AND algorithm when declarations made on the basis of data sequences

containing only single-sensor data were considered. For this case, gains in declaration time of from 8 seconds for the low-speed target and 27 seconds for the high-speed target were realized. This is an important factor for real-world multisensor systems where every second in the declaration/track initiation process is crucial, and where it cannot be assumed that all target data sequences will contain target detections from both sensors (due to unreliable target detection in one or both of the sensors).

Analysis was performed to ascertain the sensitivity of the MSTI algorithm to errors in the estimation of the spatial feature test and data sequence statistics. It was shown that the change in declaration time was usually less than 10% in response to approximately 10% changes in the spatial feature test or data sequence statistics. However, it was shown that in infrequent instances, the number of false cues out of the MSTI algorithm could increase by a factor of 3 or 4 in response to 10% changes in the data sequence statistics.

7.2 Future Directions

While the MSTI method was shown to be reasonably robust, additional improvements could be made. One area of interest is the use of a distributed fusion framework that addresses the use of correlated feature tests. Some work [42] has been done in this area, but neither a good general formulation of the

solution, nor an application to the track initiation problem has been discussed. The problem of correlated feature tests could also be addressed by the use of robust data fusion techniques (see Gerianiotis [43]).

Different sensor configurations could also be addressed, such as the use of electronic support measures (ESM). The MSTI method would readily support the addition of this type of data (in conjunction with the required feature tests).

Finally, the method could be modified to use on-line adaptive estimation of the relative frequency of occurrence of the data sequences or the feature test statistics, which is especially important in a non-stationary clutter environment. This would require an adaptive on-line optimization that addresses the concerns presented in Section 6.7.1. The use of neural-network and/or fuzzy-logic based approaches could be important in the implementation of any such approach.

Chapter 8: References

1. B. Greeley, JR.
"Falklands, gulf wars spur Navy to upgrade short-range ship defenses," *Aviation Week & Space Tech.*, pp. 17-18, Apr 1988

2. A. R. Reibman and L.W. Nolte
"Design and performance comparison of distributed detection networks," *IEEE Trans. Aerosp. and Elect. Syst.*, Vol. AES-23, pp.789-797, Nov. 1987

3. Y. Bar-Shalom and T.E. Fortmann
Tracking and Data Association,
San Diego: Academic Press, Inc. 1988

4. S.S. Blackman
Multiple Target Tracking with Radar Applications,
Norwood, MA: Artech House, Inc. 1986 pp.83-108

5. G.V. Trunk and J.D. Wilson
"Track initiation of occasionally unresolved radar targets," *IEEE Trans. Aerosp. and Elect. Syst.*, Vol. AES-17, pp. 122-130, Jan. 1981

6. S. Lin and D. Costello
Error Control Coding,
New Jersey: Prentice Hall, Inc., 1983, pp. 315-322

7. Y. Barniv and O. Kella
"Dynamic programming solution for detecting dim moving targets Part II Analysis," *IEEE Trans. Aerosp. and Elect. Syst.*, Vol. AES-23, pp.776-788, Nov. 1987

8. J. Arnold
"A dynamic programming approach to target detection with MTI radar," *SRI Tech. Report, Undated*

9. R. Bellman
Dynamic Programming,
Princeton, NJ: Princeton Univ. Press, 1957

10. H.L. Van Trees
Detection, Estimation, and Modulation Theory, Part I,
New York: John Wiley and Sons, 1968, pp. 19-52

11. M. Scott
"Raw data and processed data fusion performance investigation," *Air Force Armament Laboratory Technical Report*, AFATL-TR-90-51, Oct. 1990

12. J. D. Gibson and J. L. Mensa
Introduction to Nonparametric Detection with Applications,
New York, NY: Academic Press, 1975

13. A. Wald
Sequential Analysis,
New York: John Wiley and Sons, 1947

14. Z. Chair and P.K. Varshney
"Optimal data fusion in multiple sensor detection systems," *IEEE Trans. Aerosp. and Elect. Syst.* Vol. AES-22, pp. 98-101, Jan. 1986

15. S.C.A. Thomopoulos, R. Viswanathan, and D.C. Bougoulias
"Optimal decision fusion in multiple sensor systems," *IEEE Trans. Aerosp. and Elect. Syst.*, Vol. AES-23, pp.644-653, Sep. 1987

16. P. Nahin and J. Pokoski
"NCTR plus sensor fusion equals IFFN or can two plus two equal five ?," *IEEE Trans. Aerosp. and Elect. Syst.* Vol. AES-16 pp. 320-337, May 1980

17. R. Viswanathan and V. Aalo
"On counting rules in distributed detection," *IEEE Trans. Acoust. Speech and Sig. Proc.*, Vol. 37, pp.772-775, May 1989

18. V. Aalo and R. Viswanathan
"Asymptotic performance of a distributed detection system in correlated gaussian noise," *IEEE Trans. Acoust. Speech and Sig. Proc.*, Vol. 40, pp. 211-213, Jan. 1992

19. R.R. Tenney and N.R. Sandell, Jr.
"Detection with distributed sensors," *IEEE Trans. Aerosp. and Elect. Syst.*, Vol. AES-17, pp.501-510, Jul. 1981

20. M. Barkat and P.K. Varshney
"Decentralized CFAR signal detection," *IEEE Trans. Aerosp. and Elect. Syst.* Vol. AES-25, pp. 141-149, Mar. 1989

21. S.C.A. Thomopoulos, R. Viswanathan, and D.C. Bougoulias
"Optimal distributed decision fusion," *IEEE Trans. Aerosp. and Elect. Syst.*, Vol. AES-25, pp. 761-765, Sep. 1989

22. R. Srinivasan
"Distributed radar detection theory," *IEE Proc.* Vol. 133, Pt. F, pp. 55-60, Feb. 1986

23. A. Reibman and L.W. Nolte
"Optimal detection and performance of distributed sensor systems," *IEEE Trans. Aerosp. and Elect. Syst.*, Vol. AES-23, pp.24-30, Jan 1987

24. L. Izzo and L. Paura
"Comments on optimal detection and performance of distributed sensor systems," *IEEE Trans. Aerosp. and Elect. Syst.*, Vol. AES-25, pp.113, Jan. 1989

25. D. Teneketzis
"The decentralized quickest detection problem," *IEEE Trans. Auto. Cont.*, Vol. AC-29, pp.641-644, Jul. 1984

26. Z. Chair, I.Y. Hoballah, and P.K. Varshney
"The decentralized sequential probability ratio test," *1986 Asilomar Conf. on Sig. Sys. and Comp.*, pp.207-208

27. L. Peckham, J. Davis, and R. Allen
"IRST signal processing concepts," *Proceedings of the SPIE, Infrared Systems and Components*, Vol. 750, pp.92-104, 1987.

28. J. Barnett
"Statistical analysis of median subtraction filtering with application to point target detection in infrared backgrounds," *NRL Technical Report*, Undated.

29. V. Hansen
"Generalized CFAR processing and an application to the Weibull distribution," *Asilomar Symposium on Information Theory*, 1972

30. A. Papoulis
Probability, Random Variables and Stochastic Processes,
New York: McGraw-Hill, 1984

31. H. Sorenson
Parameter Estimation Principles and Problems,
New York: Marcel-Dekker, 1980, pp.205

32. M. Abramowitz and I. Stegun
Handbook of Mathematical Functions,
New York: Dover, 1970

33. H. Taha
Integer Programming Theory, Applications, and Computations,
New York, NY: Academic Press, 1975

34. G. R. Parker and R. L. Rardin
Discrete Optimization,
New York, NY: Academic Press, 1988

35. P. E. Gill, W. Murray and M. H. Wright
Practical Optimization,
San Diego, CA: Academic Press, 1981

36. S. Nahar, S. Sahni and E. Shragowitz
"Simulated annealing and combinatorial optimization," *23rd Design Automation Conference*, pp. 293-299, 1986

37. S. Kirkpatrick, C. D. Gelatt, Jr. and M. P. Vecchi
"Optimization by simulated annealing," *Science*, Vol. 220, No. 4598, pp. 671-680, May 1983

38. E. Aarts and J. Korst
Simulated Annealing and Boltzmann Machines,
New York, NY: John Wiley and Sons, 1989, pp.13-75

39. M. S. Bazaraa and C. M. Shetty
Nonlinear Programming,
New York, NY: John Wiley and Sons, 1979, pp. 331-342

40. K. Krueger, R. Pawlak, R. Stapleton and R. Wiss
"MSDSIM, a simulation for developing and evaluating multisensor detection algorithms," *NSWC TR working draft*

41. J. C. Eidson
"Correlation of detections for registered SAR, laser radar and passive IR data," 3rd National Sensor Fusion Conf., pp.47-63, 1990

42. Lauer, G.S. and N.R. Sandell, Jr.
"Distributed detection of known signals in correlated noise," *Alphatech Report* TP-131, Mar. 1982

43. E. Geraniotis and Y. A. Chau
"Robust data fusion for multisensor detection systems," *IEEE Trans. Aerosp. and Elect. Syst.*, Vol. AES-36, pp.1265-1279, Nov. 1990

Appendix A: Error Models

The standard deviation of the error is modeled by a second order approximation incorporating both measurement and process noise:

$$\sigma = \alpha + \beta \Delta t + \gamma \Delta t^2 \qquad \text{(A-1)}$$

where:

α= Measurement uncertainty, worst case

β= 1st derivative of the process uncertainty, worst case

γ= 2nd derivative of the process uncertainty, worst case

α is sensor dependent, while β and γ are dependent on target dynamics assumptions. For example, suppose two azimuth measurements are taken x_{a1} and x_{a2}. Suppose that the following parameters are then used:

α= 0.5 °

β= 1.0 °/sec.

γ= 0.5 °/sec.^{2}

Then suppose that two measurements arrive, separated by 2 seconds. Then the standard deviation for measurement x_{a1} is σ_{xa1}=0.5, and the standard deviation for measurement x_{a2} is σ_{xa2}=4.5. Note that worst case minimum detection

ranges must be hypothesized to determine α, β and γ. For the sensors in question, using a worst case data pair separation time of six seconds, $4\sigma_{xa2}=1.28°$.

Appendix B: Derivation of the Probability Density Function of the Log-likelihood Ratio

B.1 Expression for the Probability Density Function of the Conditional Log-likelihood Ratio

An expression for the PDF of Λ conditioned on H_1 is given by:

$$p_{\Lambda}(\lambda \mid H_1) = \mathcal{L}^{-1}\{\prod_{i=1}^{N} (b_i e^{-c_i s} + a_i e^{-d_i s})\} \tag{B-1}$$

Expanding the expression results in:

$$\begin{aligned} p_{\Lambda}(\lambda \mid H_1) &= \mathcal{L}^{-1}\{(a_1 e^{-d_1 s} + b_1 e^{-c_1 s})(a_2 e^{-d_2 s} + b_2 e^{-c_2 s}) \ \ldots \ (a_N e^{-d_N s} + b_N e^{-c_N s})\} \\ &= \mathcal{L}^{-1}\{a_1 a_2 \ldots a_N e^{-(d_1 + d_2 + \ldots + d_N)s} + b_1 a_2 \ldots a_N e^{-(c_1 + d_2 + \ldots + d_N)s} + \ldots \\ &\quad + b_1 b_2 \ldots b_N e^{-(c_1 + c_2 + \ldots + c_N)s}\} \\ &\triangleq \mathcal{L}^{-1}\{\sum_{q} \psi_1(q) e^{-\phi(q)s}\} \end{aligned} \tag{B-2}$$

The terms of the form e^{-ms} correspond to $\delta(\lambda\text{-m})$ in the log-likelihood ratio domain. The coefficients of the Dirac deltas are multinomials in a_i and b_i. Therefore, the previous equations can be expressed as follows:

$$p_{\Lambda}(\lambda \mid H_1) = \sum_{q} \psi_1(q) \delta(\lambda - \phi(q)) \tag{B-3}$$

Similarly, we can express the PDF of the log-likelihood ratio conditioned on H_0 as:

$$p_{\Lambda}(\lambda \mid H_0) = \sum_{q} \psi_0(q)\delta(\lambda - \phi(q)) \qquad \textbf{(B-4)}$$

where q is a member of the index set Q that references the terms of the expansion shown in equation (B-2). $\phi(q)$ is the q^{th} term in the sum of sums of c_i and d_i. $\Psi_1(q)$ and $\Psi_0(q)$ are the q^{th} monomials of the multinomials in a_i, b_i and f_i, g_i, respectively. P_D is then given by:

$$P_D = \int_{\lambda_1}^{\infty} \sum_{q} \psi_1(q)\delta(\lambda - \phi(q))d\lambda \qquad \textbf{(B-5)}$$

We can express P_F in a similar form:

$$P_F = \int_{\lambda_1}^{\infty} \sum_{q} \psi_0(q)\delta(\lambda - \phi(q))d\lambda \qquad \textbf{(B-6)}$$

Define P_C as the probability of a cue occurring. This desired probability will be determined by the following factors: the amount of time it takes to prosecute a cue, the nominal search time, and the maximum search time. Thus, P_C, the probability of a cue, can be derived as follows:

$$P_C = \int_{\lambda_0}^{\lambda_1} \sum_q [P_0 \psi_0(q) + P_1 \psi_1(q)] \delta(\lambda - \phi(q)) d\lambda \qquad \text{(B-7)}$$

In a typical search and track system, P_0 and P_1 are not known, however we can make the assumption that $P_0 >> P_1$ (which implies $P_0\Psi_0(\cdot) >> P_1\Psi_1(\cdot)$). Thus, P_0 is very nearly unity, and the expression for P_C reduces to:

$$P_C \approx \int_{\lambda_0}^{\lambda_1} \sum_q \psi_0(q) \delta(\lambda - \phi(q)) d\lambda \qquad \text{(B-8)}$$

We also see that the probability of detection for a cue is given by:

$$P_{DC} = \int_{\lambda_0}^{\lambda_1} \sum_q \psi_1(q) \delta(\lambda - \phi(q)) d\lambda \qquad \text{(B-9)}$$

Performing the integrations results in:

$$P_D = \sum_q \psi_1(q) I(\phi(q) - \lambda_1) \qquad \text{(B-10)}$$

$$P_F = \sum_q \psi_0(q) I(\phi(q) - \lambda_1) \qquad \text{(B-11)}$$

$$P_C \sim \sum_q \psi_0(q)(I(\phi(q)-\lambda_0)-I(\phi(q)-\lambda_1)) \qquad \text{(B-12)}$$

$$P_{DC} = \sum_q \psi_1(q)(I(\phi(q)-\lambda_0)-I(\phi(q)-\lambda_1)) \qquad \text{(B-13)}$$

where $I(\cdot)$ is the indicator or unit step function.

B.2 Efficient Computation of the Probability Density Function of the Log-likelihood Ratio

In examining equation (B-2), we see that the coefficients of the exponential terms are given by all the possible unique arrangements of a_i and b_i, *i.e.*, all the possible binary N-tuples. The entries in the exponential terms have a one-to-one relationship to the corresponding combination of constants in the coefficient terms (*i.e.*, if a_i appears in the coefficient term, then d_i will appear in the exponential).

We then see that an easy way to calculate the PDF of the conditional log-likelihood ratio is via the following procedure:

1. Construct a matrix Q_N containing all the unique binary words of length N. Q_N will be 2^N x N.
2. Define a new matrix R with the following elements:

$$r_{ij}=\begin{cases} a_i & if \quad q_{ij}=0 \\ b_i & if \quad q_{ij}=1 \end{cases} \tag{B-14}$$

Similarly define a matrix S with the elements:

$$s_{ij}=\begin{Bmatrix} d_i & if\ q_{ij}=0 \\ c_i & if\ q_{ij}=1 \end{Bmatrix} \tag{B-15}$$

3. Then calculate the vectors **r** and **s** using the following equations

$$r_i=\prod_{j=1}^{N} r_{ij} \tag{B-16}$$

$$s_i=\sum_{j=1}^{N} s_{ij} \tag{B-17}$$

Note that while the coefficients (r_i) and the locations (s_i) of the impulse functions have been calculated, they are no longer in the proper order.

4. Perform a sort on **s**; the resulting index set of the sort is then used to reorder **r**.

5. Combine all duplicate entries in **s** (*i.e.*, sum r_i and r_{i+1} into r_i whenever $s_i=s_{i+1}$, and delete s_{i+1}, r_{i+1}).

6. Formally, the continuous time result is then given by

$$p_{\Lambda}(\lambda \mid H_1) = \sum_{i=1}^{N_d} r_i \delta(\lambda - s_i) \qquad \text{(B-18)}$$

where $N_d \leq 2^N$. We note that $N_d = \mathfrak{C}(Q)$. The number of multiplies used to compute the r_i terms may be reduced by using values of the a_i and b_i products that have already been computed. Similarly, the number of sums used to compute s_i may be reduced.

If we reuse sum and product terms wherever possible, then the number of operations required is less than or equal to:

$$2\sum_{i=2}^{N} 2^i \text{ multiplications } + [\sum_{i=2}^{N} 2^i + 2^N - N - 1] \text{ additions } + \text{ maximum length } 2^N \text{ sort} \qquad \text{(B-19)}$$

This is due to the following: We see that to compute r we require NM multiplications

$$NM = \sum_{i=2}^{N} 2^i \qquad \text{(B-20)}$$

with 2 vectors being computed (one for each hypothesis). Because of the one-to-one correspondence of the Dirac delta locations (given by the elements of **s**) and the amplitudes of the deltas (given by the elements of **r**), we note that the number of additions needed to compute **s** is also given by (B-20). The combination of duplicate entries in the abscissa will require a maximum of 2^N-

N-1 additions. This is because the minimum number of terms in the abscissa and ordinate N_d result when all the P_Ds and P_Fs are equal across the sensors. When this arises, the conditional PDFs are binomially distributed and thus have N+1 possible values. Since the number of entries in the vector is 2^N before the sort, the total number of possible additions is 2^N-(N+1).

Appendix C: Sensor Parameters

Both Sensors:	
azimuth field of view (azimuth scanning)	20°
radar:	
scan time:	variable
frequency:	X-band
range resolution:	15 m.
maximum range:	40 km.
horiz. beamwidth:	2.0°
approximate horiz. resolution after monopulse	0.1°
vert. beamwidth:	1.0°
infrared :	
scan time:	0.5 sec.
vertical instantaneous field of view:	0.1 milliradians (5.73e-3°)
horizontal instantaneous field of view	0.5 milliradians. (2.87e-2°)
lower wavelength:	3.8μ. (micrometer)
upper wavelength:	4.1μ.
vertical field of view:	0.7334°

Appendix D: Approximating the Cumulative Probability Density Function of the Log-likelihood Ratio

D.1 Introduction

A method for computing the probability density function (PDF) and cumulative probability density function (CPDF) of the log-likelihood ratio was presented in Appendix B. This method is suitable when the number of feature tests is small. However, as N increases, the number of points in the CPDF increases as 2^N (assuming that the statistics of the feature tests are all distinct). Therefore, under the conditions of large N, it is advantageous to compute approximations to the CPDFs. There are many possible approaches to this problem. Possibly the simplest of these is to assume that the distribution of the log-likelihood ratio will become Gaussian in the limit, since Λ is the sum of independent random variables (Λ_i). However, if the feature test statistics are distinct, the assumption that Λ_i are identically distributed does not hold, and convergence is slow.

This Appendix analyzes two approaches to the problem. The first of these is to use a Gaussian approximation of the CPDF. The other is to apply a corrected Gaussian approximation utilizing a truncated Taylor series expansion that makes use of the higher order cumulants. Results are presented

for $p_\Lambda(\lambda|H_1)$, with the understanding that a proper substitution of variables can be made for H_0.

D.1.1 Gaussian Approximation

The mean and standard deviation of the multinomial distribution can be calculated as follows:

$$\begin{aligned}\mu &= E[\Lambda|H_1] \\ &= \int_{-\infty}^{\infty} \lambda \sum_q \psi_1(q)\delta(\lambda-\phi(q))d\lambda \\ &= \sum_q \psi_1(q)\phi(q)\end{aligned} \qquad \textbf{(D-1)}$$

$$\begin{aligned}\sigma^2 &= E[(\Lambda-\mu)^2|H_1] \\ &= \sum_q \psi_1(q)\phi^2(q) - \mu^2\end{aligned} \qquad \textbf{(D-2)}$$

The calculation of the CPDF can then be performed using the following series approximation (from Abramowitz and Stegun [32]):

$$CG(\Lambda_n) \Delta 1 - .5*(1+\tau_1\Lambda_n+\tau_2\Lambda_n^2+\tau_3\Lambda_n^3+\tau_4\Lambda_n^4+\tau_5\Lambda_n^5+\tau_6\Lambda_n^6)^{-16}+err(\Lambda_n)$$

$$|err(\Lambda_n)|<1.5\cdot10^{-7}$$

$$\tau_1=.0498673470$$

$$\tau_2=.0211410061$$

$$\tau_3=.0032776263$$

$$\tau_4=.0000380036$$

$$\tau_5=.0000488906$$

$$\tau_6=.0000053830$$

(D-3)

where Λ_n is the normalized log-likelihood ratio at the fusion center ($\Lambda_n=(\Lambda-\mu)/\sigma$). Figure D.1 shows the mean-squared error associated with using this fit. The mean-squared error values were calculated over 1000 samples with P_{Di}~U(0.5,1.0) and P_{Fi}~U(0.0,0.5). Results were averaged over 30 runs each of values of N=2,3,4, ... 10 (where N is the number of feature tests).

Note that although this method requires the computation of the PDF directly, not all the values need to be stored (*i.e.*, the terms in the log-likelihood ratio can be computed recursively); thus this approximation can be used when storage requirements would normally fix an upper limit on N.

D.1.2 Higher Order Fit Using Cumulants

A Gaussian fit with a Taylor series correction factor can also be used as an approximation to the CPDF of the multinomial distribution. Such a fit is detailed by Abramowitz and Stegun [32] and is given by the following equation:

$$p_{\Lambda_n}(\lambda \mid H_1) = CG(\Lambda_n) - \left(\frac{\gamma_1}{6}G^{(2)}(\Lambda_n) + \left(\frac{\gamma_2}{24}G^{(3)}(\Lambda_n) + \frac{\gamma_1^2}{72}G^{(5)}(\Lambda_n) - \left(\frac{\gamma_3}{120}G^{(4)}(\Lambda_n) + \right.\right.\right.$$
$$\gamma_1\frac{\gamma_2}{144}G^{(6)}(\Lambda_n) + \frac{\gamma_1^3}{1296}G^{(8)}(\Lambda_n)\left.\right) + \left(\frac{\gamma_4}{720}G^{(5)}(\Lambda_n) + \frac{\gamma_2^2}{1152}G^{(7)}(\Lambda_n) + \right. \quad \textbf{(D-4)}$$
$$\left.\left.\gamma_1\frac{\gamma_3}{720}G^{(7)}(\Lambda_n) + \gamma_1^2\frac{\gamma(2)}{1728}G^{(9)}(\Lambda_n) + \frac{\gamma_1^4}{31104}G^{(11)}(\Lambda_n)\right)\right.$$

where $G^{(i)}(\cdot)$ is i^{th} derivative of the Gaussian distribution function, $CG(\cdot)$ is the cumulative Gaussian distribution function, and

$$\gamma_{r-2} = \frac{1}{n^{\frac{r}{2}-1}} \frac{\left(\frac{1}{N}\sum_{i=1}^{N}\kappa_{r,i}\right)}{\left(\frac{1}{N}\sum_{i=1}^{N}\sigma_i^2\right)^{\frac{1}{2}}} \qquad \textbf{(D-5)}$$

$\kappa_{r,i}$ is the r^{th} cumulant for the i^{th} feature test log-likelihood ratio PDF. The expression for the normalized log-likelihood ratio at the fusion center is now:

$$\Lambda_n = \frac{\sum_{i=1}^{N} (\Lambda_i - \mu_i)}{(\sum_{i=1}^{N} \sigma_i^2)^{\frac{1}{2}}} \qquad \textbf{(D-6)}$$

where σ_i is the standard deviation of the i_{th} log-likelihood ratio, and μ_i is the mean of the i^{th} log-likelihood ratio. For the problem in question:

$$\mu_i = a_i d_i + b_i c_i \ , \quad \sigma_i = a_i d_i^2 + b_i c_i^2 - (a_i d_i + b_i c_i)^2 \qquad \textbf{(D-7)}$$

The moment generating function is:

$$E[e^{s\lambda}] = a_i e^{d_i s} + b_i e^{c_i s} \qquad \textbf{(D-8)}$$

and the cumulants are given by the Taylor series expansion about s=0 of the cumulant generating function:

$$\sum_{r=0}^{\infty} k_{r,i} \frac{s^r}{r!} = \ln[E[e^{s\Lambda}]] = \ln[a_i e^{d_i s} + b_i e^{c_i s}] \qquad \textbf{(D-9)}$$

and thus after some calculation, the cumulants are:

$$\kappa_{1,i} = a_i d_i + b_i c_i \qquad \textbf{(D-10)}$$

$$\kappa_{2,i} = a_i d_i^2 + b_i c_i^2 - (a_i d_i + b_i c_i)^2 \qquad \textbf{(D-11)}$$

$$\kappa_{3,i} = a_i d_i^3 + b_i c_i^3 - 3(a_i d_i^2 + b_i c_i^2)(a_i d_i + b_i c_i) + 2(a_i d_i + b_i c_i)^3 \qquad \textbf{(D-12)}$$

$$\begin{aligned}\kappa_{4,i} = & a_i d_i^4 + b_i c_i^4 - 4(a_i d_i^3 + b_i c_i^3)(a_i d_i + b_i c_i) \\ & + 12(a_i d_i^2 + b_i c_i^2)(a_i d_i + b_i c_i)^2 \\ & - 3(a_i d_i^2 + b_i c_i^2)^2 - 6(a_i d_i + b_i c_i)^4\end{aligned} \tag{D-13}$$

$$\begin{aligned}\kappa_{5,i} = & a_i d_i^5 + b_i c_i^5 - 5(a_i d_i^4 + b_i c_i^4)(a_i d_i + b_i c_i) \\ & + 20(a_i d_i^3 + b_i c_i^3)(a_i d_i + b_i c_i)^2 \\ & - 10(a_i d_i^3 + b_i c_i^3)(a_i d_i^2 + b_i c_i^2) \\ & - 60(a_i d_i^2 + b_i c_i^2)(a_i d_i + b_i c_i)^3 \\ & + 30(a_i d_i^2 + b_i c_i^2)^2 (a_i d_i + b_i c_i) \\ & + 24(a_i d_i + b_i c_i)^5\end{aligned} \tag{D-14}$$

$$\begin{aligned}\kappa_{6,i} = & a_i d_i^6 + b_i c_i^6 - 6(a_i d_i^5 + b_i c_i^5)(a_i d_i + b_i c_i) \\ & + 30(a_i d_i^4 + b_i c_i^4)(a_i d_i + b_i c_i)^2 - 15(a_i d_i^4 + b_i c_i^4)(a_i d_i^2 + b_i c_i^2) \\ & - 120(a_i d_i^3 + b_i c_i^3)(a_i d_i + b_i c_i)^3 \\ & + 120(a_i d_i^3 + b_i c_i^3)(a_i d_i + b_i c_i)(a_i d_i^2 + b_i c_i^2) \\ & - 10(a_i d_i^3 + b_i c_i^3)^2 + 360(a_i d_i^2 + b_i c_i^2)(a_i d_i + b_i c_i)^4 \\ & - 270(a_i d_i^2 + b_i c_i^2)^2 (a_i d_i + b_i c_i)^2 \\ & + 30(a_i d_i^2 + b_i c_i^2)^3 - 120(a_i d_i + b_i c_i)^6\end{aligned} \tag{D-15}$$

Figure D.1 illustrates the mean-squared error associated with using this fit.

Note that this approximation requires the computation of the cumulants and the higher order derivatives of the normal PDF. However, the relationship between the number of cumulants to be calculated and N is linear, whereas the number of terms to be calculated in equations (D-1), (D-2) rises as 2^N.

D.2 Conclusions

It is seen from Figure D.1 that for values of N=6 or greater, the corrected Gaussian approximation gives a better estimate than the standard Gaussian approximation. Therefore, the tradeoff becomes one of computation speed and required accuracy. For low values of N it is advantageous to compute the CPDF of the log-likelihood ratio directly. For higher values of N, it is clearly advantageous to use one of the approximations, with the corrected Gaussian being the better of the two (as long as the accuracy of the approximation is not a problem). Finally, note that the approximation shown in equation (D-3) will become a limiting factor on the error as N increases.

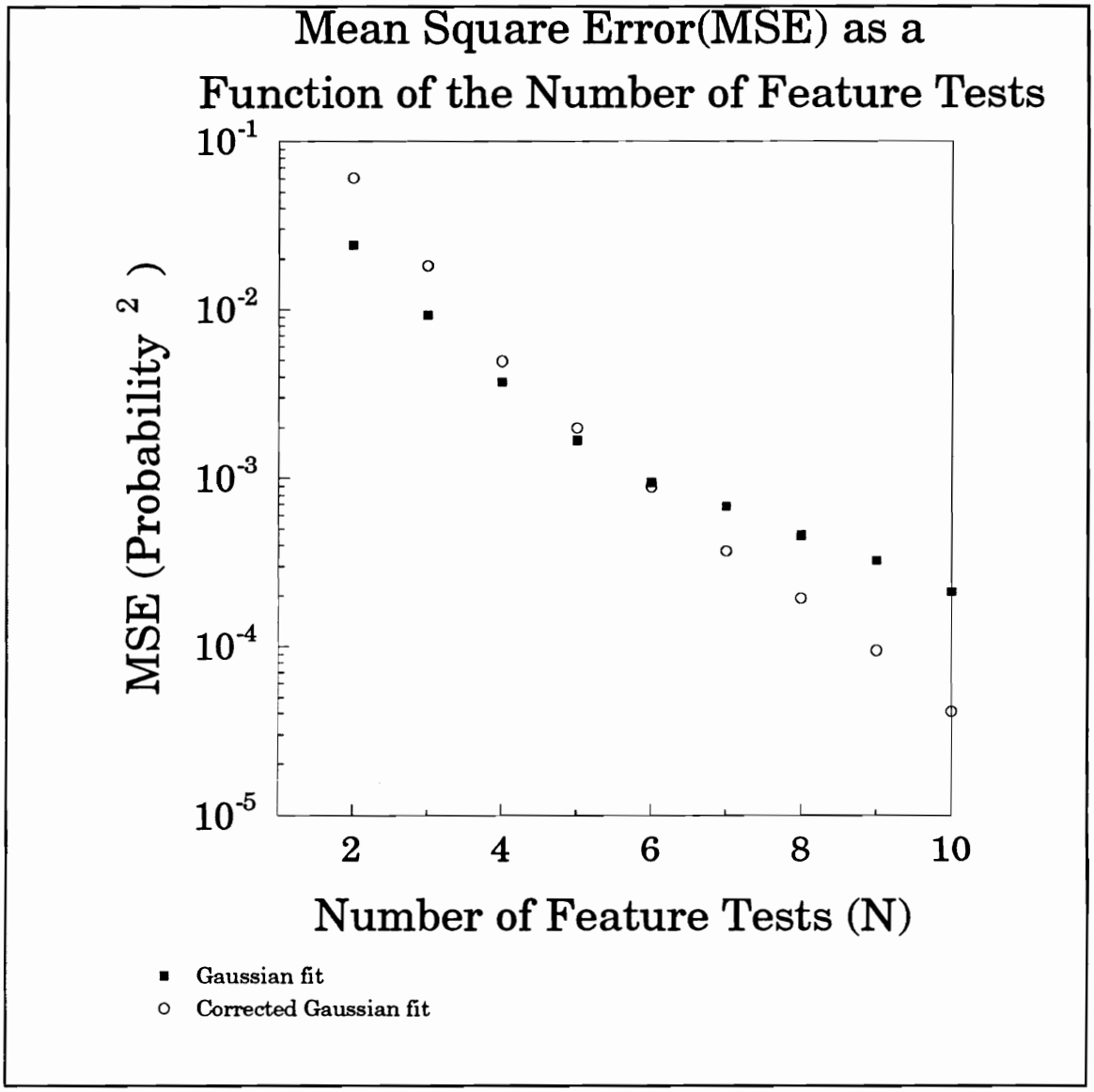


Figure D.1 Mean-squared error of Gaussian and corrected Gaussian fits

Appendix E: Variable List

α	measurement error
$\mathbf{6}$	some large number exceeding the domain of $\Phi_w(q)$ $\forall$ $q \in Q$
β	first derivative of process error
γ	second derivative of the process error
Δt	time difference between measurements taken at time t_i and t_{i+1}
κ	temperature multiplier
$\kappa_{r,i}$	r^{th} cumulant for the i^{th} feature test log-likelihood ratio PDF
$\Lambda(u)$	log-likelihood ratio across u
λ_0	cue threshold
$\lambda_{0,w}$	cue threshold for a data sequence of type w
λ_1	declaration threshold
$\lambda_{1,w}$	declaration threshold for a data sequence of type w
Λ_n	normalized log-likelihood ratio
ν	maps W to Z
ρ_C	cue penalty multiplier
$\rho_{C<D}$	declaration cue inequality parameter
ρ_F	false alarm penalty multiplier
ρ_I	infinitive inequality penalty multiplier

$\sigma_{?i}$	standard deviation of measurement i of type ?, where ?=x_{ai}, x_{ei}, x_{ri}, v_i, x_{ad}, x_{ed}, x_{rd} v_m.
$\Phi(q)$	gives the locations of the impulse functions for the PDF of the log-likelihood ratio
$\Phi_w(q)$	value of $\Phi(q)$ for data sequence type w
$\Psi_i(q)$	gives the amplitudes of the impulse functions in $\Phi(q)$ conditional on H_i
$\Psi_{i,w}(q)$	value of $\Psi_i(q)$ for data sequence type w
ω_i	decision weight factor for sensor i
$CG(\cdot)$	cumulative Gaussian PDF
c_k	temperature parameter for the simulated annealing method at iteration k
$G(\cdot)$	Gaussian PDF
H_i	hypothesis i
J	overall objective or cost function to be minimized
$L_{i,i+1}$	path length between data points i and i+1
$l_{i,k}$	subset of log-likelihood ratio thresholds at the k^{th} iteration for threshold i
$l_{i,opt}$	optimal threshold set for threshold set i
L_p	total path length
m	true target position

M	number of sensors (after Chapter 1)
N	number of sensors (Chapter 1), the number of detections (Chapter 1), or the number of feature tests (after Chapter 1)
NA	number of possible data sequences
N_d	number of terms in the PDF of the log-likelihood ratio
NM	number of multiplies used to compute **r**
$O(\cdot)$	denotes the number of occurrence of an event
P_0	probability of hypothesis H_0 occurring
P_1	probability of hypothesis H_1 occurring
P_C	system probability of cue
$P_{Cdesired}$	desired P_C
$P_{C,w}$	probability of cue for data sequence type w
P_D	system probability of detection
P_{DC}	system probability of detection for a cue
$P_{DC,w}$	probability of detection for a cue for data sequence type w
P_{Di}	probability of detection for feature test i
$P_{D,w}$	probability of detection for data sequence type w
P_F	system probability of false alarm
$P_{Fdesired}$	desired P_F
P_{Fi}	probability of false alarm for feature test i
$P_{F,w}$	probability of false alarm for data sequence type w

P_{Mi}	probability of miss for feature test i
q	$q \in Q$
Q	an index set with members pointing to the i^{th} term in the PDF of the log-likelihood ratio
Q_N	a binary matrix that has rows which correspond to the elements of Q in binary form
R	neighborhood size about $l_{i,k}$
r,s	dummy variables used in the computation of the log-likelihood ratio
s	maximum data sequence length
t_i	time at measurement i
u	vector of feature test results
v_i	i^{th} range rate measurement
v_m	maximum a posteriori/mean-squared error estimate of the range rate
w	a data sequence, where $w \in W$
W	index set of data sequences
x	denotes a dummy random variable
x_{ai}	i^{th} azimuth measurement
x_{ei}	i^{th} elevation measurement
x_{ri}	i^{th} range measurement

r_{max}	maximum instrumented range
r_{min}	minimum instrumented range
y	denotes a dummy random variable
y_i	observation or measurement for sensor i
z	a set of feature measurements or dummy random variable
z_T	a spatial threshold
Z	set of possible feature measurements
z_w	$z_w \subset Z$ the subset of feature measurements corresponding to w
$'$	denotes a predicted value
$\bar{}$	denotes the expected value
$\varrho(\cdot)$	penalty function

VITA

NAME	Robert J. Pawlak
ADDRESS	Code F44, Electro-Optical Systems Branch Naval Surface Warfare Center Dahlgren, VA 22448
DATE OF BIRTH	May 19, 1963
BIRTHPLACE	Buffalo, NY
EDUCATION	Bachelor of Science in Electrical Engineering (1985) Grove City College Grove City, PA
	Master of Science in Electrical Engineering (1987) Virginia Polytechnic Institute and State University Blacksburg, VA
PROFESSIONAL SOCIETIES	Institute of Electrical and Electronics Engineers (1982-Present)
EMPLOYMENT	Teaching Assistant (1985-1987) Virginia Polytechnic Institute and State University Blacksburg, VA
	Electrical Engineer (1987-Present) Code F44, Electro-Optical Systems Branch Naval Surface Warfare Center Dahlgren, VA 22448
RESEARCH INTERESTS	Data fusion techniques, neural networks, nonparametric decision theory

Signed: