UAA-001

U.S. Army Intelligence Center and School
Software Analysis and Management System

Intelligence/Electronic Warfare (IEW) Direction-Finding and Fix Estimation Analysis Report

Volume 2 TRAILBLAZER

December 20, 1985

National Aeronautics and
Space Administration

JPL

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California

JPL D-180, Vol. 2

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER UAA-001	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Intelligence/Electronic Warfare (IEW) Direction-Finding and Fix Estimation Analysis Report, Vol 2, TRAILBLAZER		5. TYPE OF REPORT & PERIOD COVERED Final
		6. PERFORMING ORG. REPORT NUMBER D-180
7. AUTHOR(s) R. Gardner, J. Gillis, Dr. A. Griesel, B. Pardo, SFC N. Sizemore		8. CONTRACT OR GRANT NUMBER(s) NAS7-918
9. PERFORMING ORGANIZATION NAME AND ADDRESS Jet Propulsion Laboratory, ATTN: 171-209 California Institute of Technology 4800 Oak Grove, Pasadena, CA 91109		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS RE182 AMEND #187
11. CONTROLLING OFFICE NAME AND ADDRESS Commander, USAICS ATTN: ATSI-CD-SF Ft. Huachuca, AZ 85613-7000		12. REPORT DATE 20 Dec 85
		13. NUMBER OF PAGES 146
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Commander USAICS ATTN: ATSI-CD-SF Ft. Huachuca, AZ 85613-7000		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE None

16. DISTRIBUTION STATEMENT (of this Report)

Approved for Public Disemination

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Prepared by Jet Propulsion Laboratory for the U.S. Army Intelligence Center and School's Combat Developer's Support Facility.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Fix Estimation, Statistical Assumptions, Error Budget, Unmodeled Errors, Coding Errors, Peak Searching, Convergence

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report contains an analysis of the direction finding (DF) and fix estimation algorithms in TRAILBLAZER. The TRAILBLAZER software analyzed is old and not currently used in the field. However, the algorithms here analyzed are used in other current IEW systems. The report examines the underlying algorithm assumptions (including unmodeled errors) and their appropriateness for TRAIL-BLAZER. Coding and documentation problems are then discussed. A detailed error budget is presented in Appendix B. An extensive bibliography on fix estimation procuedres is contained in Appendix A.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

U.S. ARMY INTELLIGENCE CENTER AND SCHOOL
Software Analysis and Management System

INTELLIGENCE/ELECTRONIC WARFARE (IEW) DIRECTION-FINDING AND FIX ESTIMATION ANALYSIS REPORT VOLUME 2 TRAILBLAZER

December 20, 1985

Authors:

Robert Gardner
Algorithm Analysis Subgroup

James W. Gillis, Subgroup Leader
Algorithm Analysis Subgroup

Ann Griesel, Supervisor
Algorithm Analysis Subgroup

Bruce Pardo
Algorithm Analysis Subgroup

SFC Nicky L. Sizemore, Consultant
USAICS

Concur:

John R. Radbill, Manager
USAMS Task

J. P. McClure, Manager
Ground Data Systems Section

Peter Gray, Acting Manager
Communications and Data
Systems Development Subprogram

QUALITY INSPECTED 4

Accesion For	
NTIS CRA&I	☑
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution /	
Availability Codes	
Dist	Avail and / or Special
A-1	

JET PROPULSION LABORATORY
California Institute of Technology
Pasadena, California

ACKNOWLEDGMENT

The following people have contributed to this document:

N. Covella

E. Drell

C. Lowry

V. Nibler

N. Palmer

P. South

A. Walker

Page intentionally left blank

ABBREVIATIONS AND ACRONYMS

ACM	Association for Computing Machinery
AGTELIS	Automated Ground Transportable Emitter Location and Identification System
BETA	Battlefield Exploitation and Target Acquisition
DF	Direction Finding
DOD	Department of Defense
EEP	Elliptical Error Probability
ELINT	Electronic Intelligence
EOB	Enemy Order of Battle
GR	GUARDRAIL
IPF	Information Processing Facility
ITEP	Interim Tactical ELINT Processor
JPL	Jet Propulsion Laboratory
LOB	Line-of-Bearing
LOP	Line-of-Position
MCS	Master Control Station
MGRS	Military Grid Reference System
QUICKLOOK	An Airborne Non-Communication Emitter Location and Identification System
RMS	Root-Mean-Square
RSS	Remote Slave Station
SNR	Signal-to-Noise-Ratio
TCATA	TRADOC Combined Arms Training Activity
TR	TRAILBLAZER
USAICS	U. S. Army Intelligence Center and School
USAMS	USAICS Software Analysis and Management System
UTM	Universal Transverse Mercator

Page intentionally left blank

CONTENTS

APPENDIXES

CONTENTS (Continued)

Figures

SECTION 1
INTRODUCTION

1.1 PURPOSE

This analysis focuses on illuminating the logical and mathematical structure of the location estimating algorithms found in the TRAILBLAZER system, and identifying the assumptions that must hold for these algorithms to give valid results. TRAILBLAZER is one of several current U. S. Army direction-finding systems. These systems use several lines-of-bearing to estimate the location of an enemy emitter. Such a location estimate is often called a "fix." Several general methods for direction finding and fix estimations, some with more mathematically rigorous foundations, some frankly empirical, are discussed in Intelligence/Electronic Warfare (IEW) Direction Finding and Fix Estimation Analysis Report, Volume 1, Overview. The TRAILBLAZER algorithms analyzed belong to that most interesting hybrid class of empirical algorithms with a strong mathematical flavor. Although the designer of such an algorithm often has a specific mathematical structure in mind, the empirical nature of the algorithm often leaves the analyst several possible mathematical interpretations. This richness of interpretation increases the understanding of just how well the algorithms function in various environments and how compatible they are with algorithms found in other systems.

1.2 BACKGROUND

This algorithm analysis effort is being performed by the Jet Propulsion Laboratory for the U. S. Army Intelligence Center and School as a research-type effort to increase the understanding of the hybrid mathematical/empirical algorithms found in intelligence processing systems. Algorithm results from one system are frequently used as input data for another system. Understanding both the assumptions under which the algorithms work, and the assumptions their results satisfy, is crucial to understanding the overall system. This view of a metasystem of intelligence processing systems (see Figure 1-1) is central to this algorithm analysis effort.

For purposes of these studies, "algorithm" means a set of rules for carrying out a single conceptual operation on a set of data. There are many types of algorithms necessary to the operation of the metasystem shown in Figure 1-1. Analyses reported on so far, listed in Appendix E, have focused on four of these: geographical transformation algorithms, self and cross-correlation algorithms, and aggregation algorithms. Geographical transformation algorithms translate locations from one grid reference system to another. These algorithms appear in almost all systems, often as incoming data or report preparation functions. Self-correlation algorithms test if the entity referred to in a new report has already been recorded in the database that reflects the estimated enemy situation. Cross-correlation algorithms test if a sighted piece of equipment belongs to an already identified unit, or a lower echelon unit to a higher echelon one. Aggregation algorithms try to identify an artillery battery in a cluster of equipment, a division in a group of regiments, or like groupings. Several statistical issues arising particularly in the correlation algorithms, are analyzed in a companion set of technical memoranda.

Looking once more at Figure 1-1, note that the same intelligence function, hence algorithms performing that function, is often embedded in several intelligence processing systems. Some generic algorithms, such as the geographical transformation algorithms mentioned above, appear in almost all systems. Comparing these algorithms that perform the same function in different systems increases the understanding not only of what these algorithms actually do and how well they perform, but also increases the understanding of how a "good" algorithm would work and what it would look like. Such comparisons should lead to developing criteria for selecting algorithms for embedding in new or upgraded systems, and finally in the creation of a library of "good" algorithms from which the choice can be made. The development of these criteria and building such a library are two major goals of this algorithm analysis effort to which each analysis of an algorithm in an existing system contributes.

ALL SOURCE PROCESSING OVERVIEW
FROM ALL SOURCES THRU THE CROSS CORRELATION PROCESS

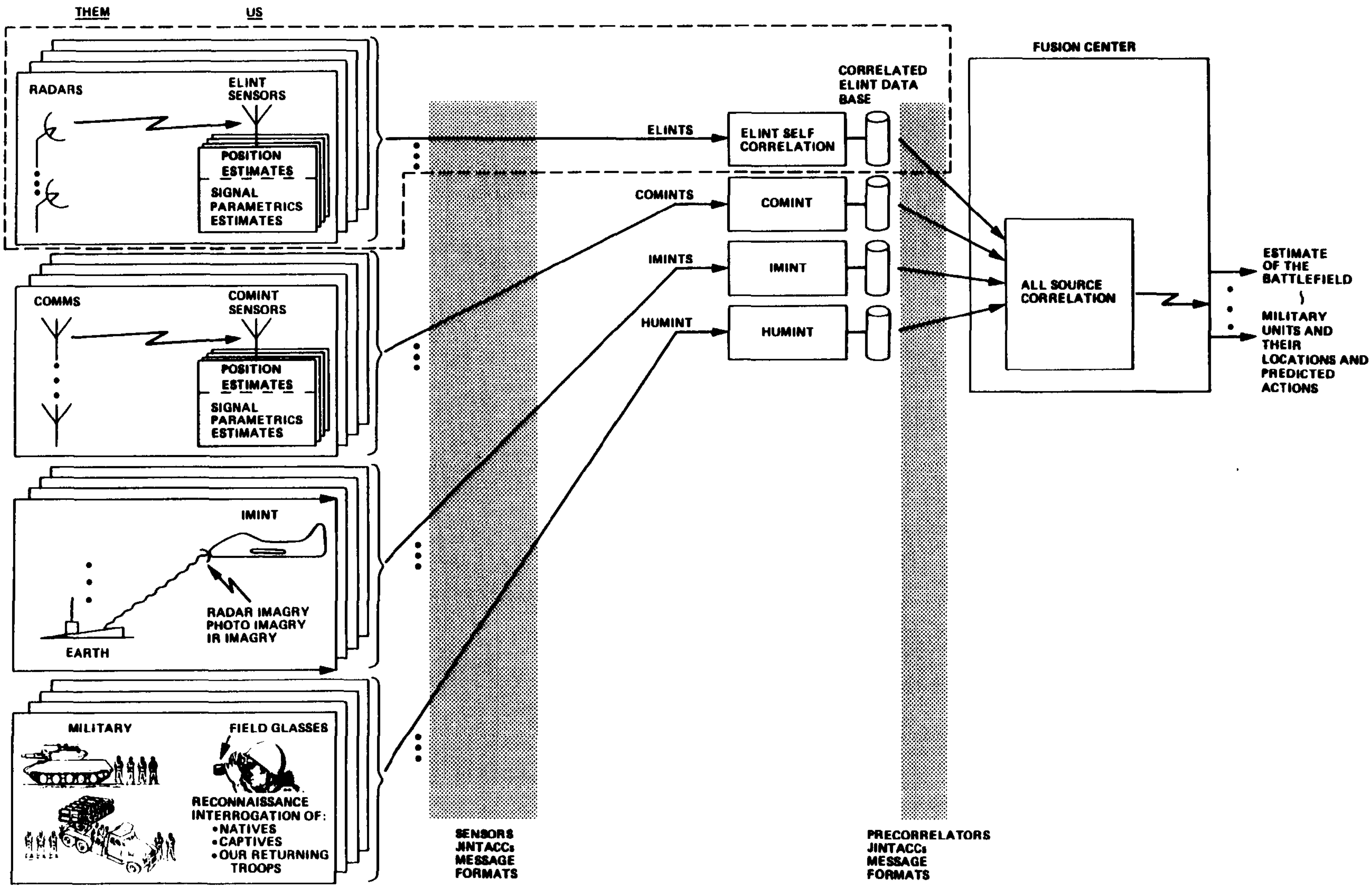

Figure 1-1. Overview - From All Sources Through the Cross-Correlation Process

SECTION 2

ASSUMPTIONS, RESTRICTIONS, SCOPE

2.1 BRIEF DESCRIPTION OF RADIO DIRECTION-FINDING AND POSITION FIXING

The purpose of radio direction-finding is to estimate or fix the position of selected emitters. Usually, the position estimate is accompanied by a confidence region reflecting measurement errors, propagation errors, and modeling errors.

Radio direction-finding (DF) requires that an emitter be viewed from at least two DF stations spaced far enough apart that their look angles intersect as close to 90 degrees as possible. However, 90° is usually impossible under battlefield conditions. Figure 2-1 illustrates a simple situation of two DF stations.

The fix estimate is at the point of intersection of the two lines-of-bearing (LOBs) (Figure 2-1). Since there is only one point of intersection, we have insufficient information to estimate the fix uncertainty due to measurement, propagation, and modeling errors.

In a multiple DF station configuration, there are many intersections (Figure 2-2). A more accurate fix estimate may be obtained by evaluating the clustering of these intersections. Since each intersection is a simple fix estimate, the uncertainty can then be expressed as a confidence region surrounding this fix estimate. This uncertainty reflects:

(1) Random measurement errors in measuring the lines-of-bearing.

(2) Errors because of different radio propagation effects along the lines-of-bearing.

(3) Errors because of spherical or flat-Earth assumptions.

(4) Phantom or ghost intersections because of the presence of multiple emitters or hidden emitter reflectors.

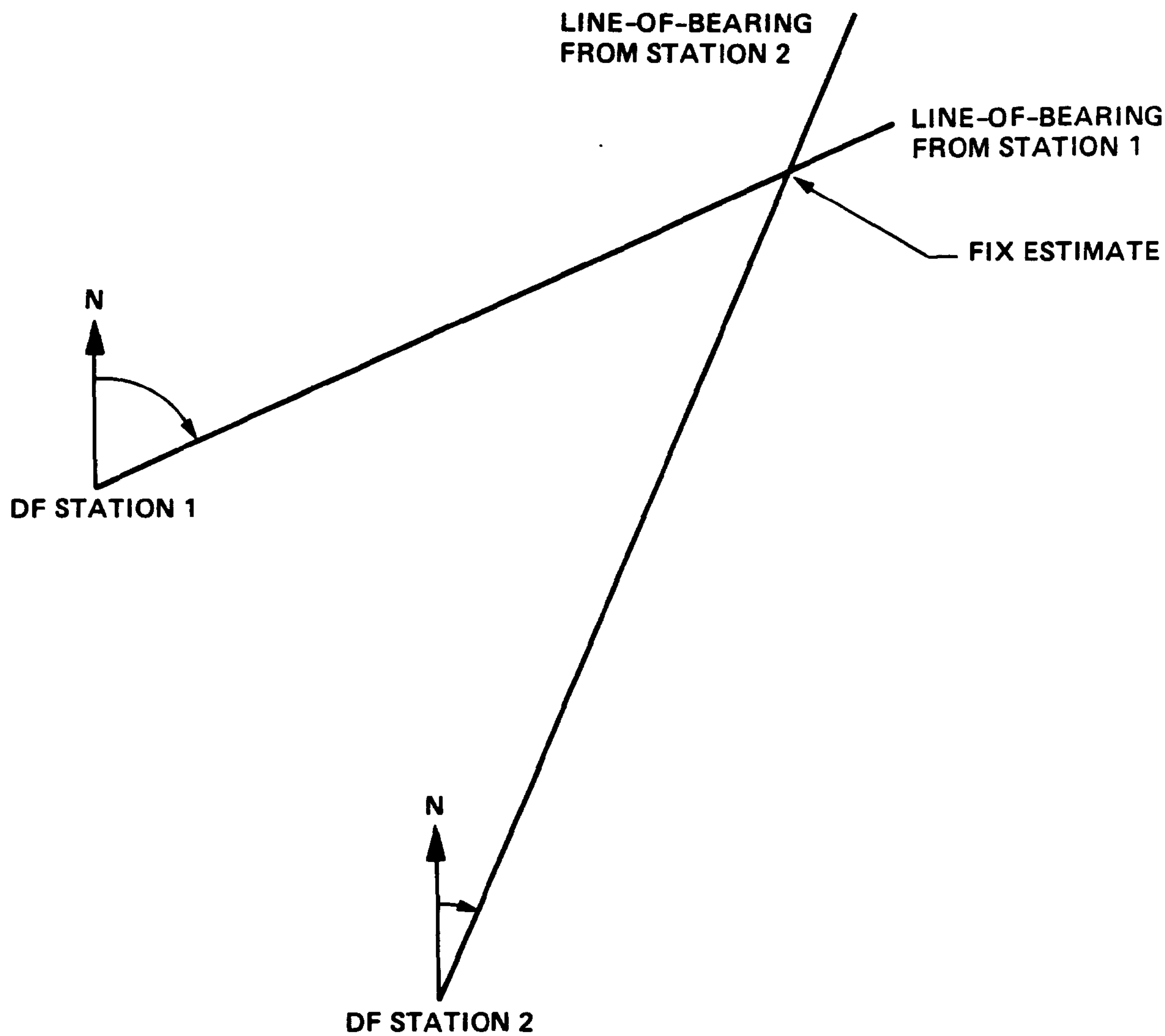

Figure 2-1. Relative Location of Two Direction-Finding Stations

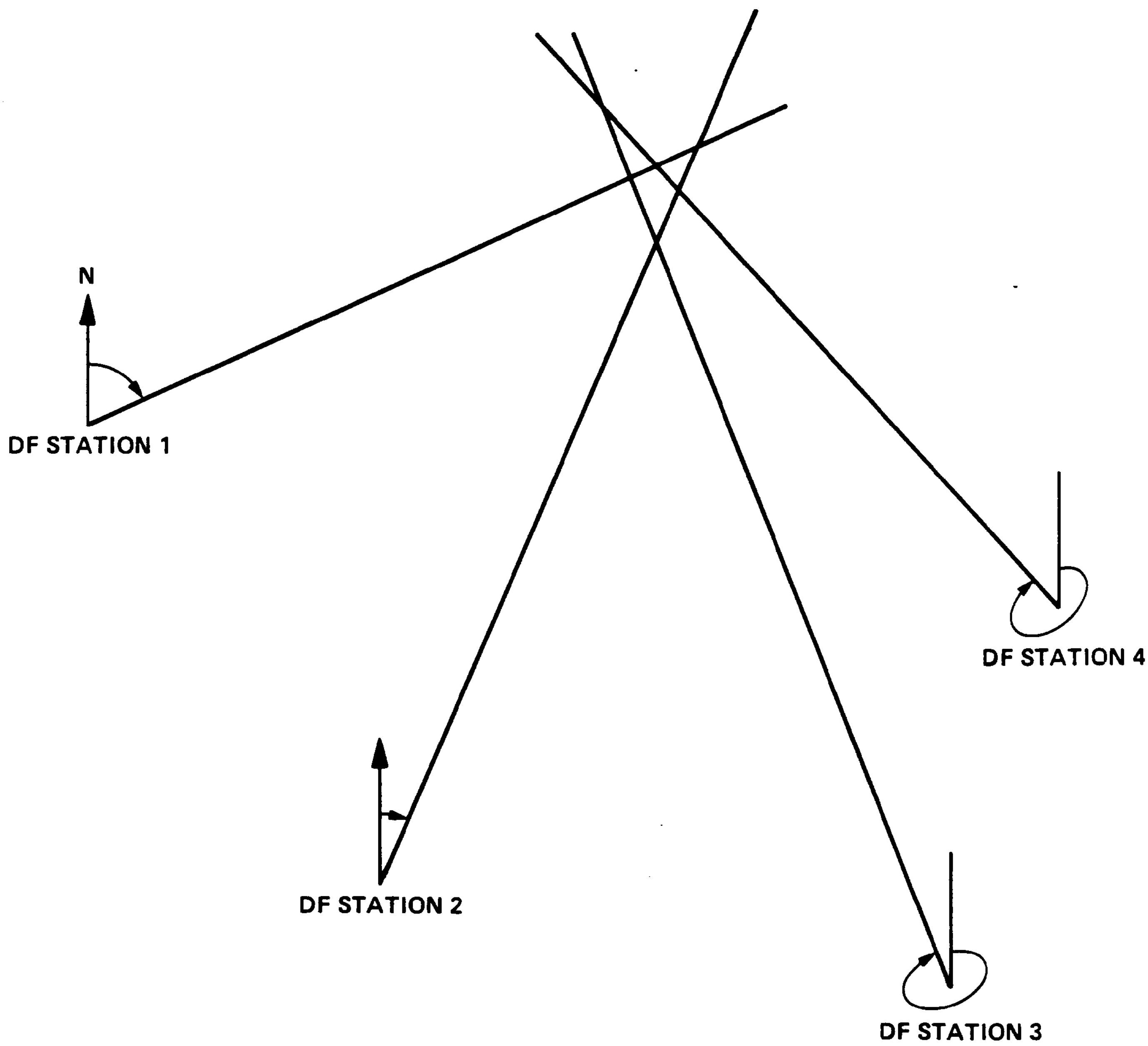

Figure 2-2. Relative Location of Several Direction-Finding Stations

2.2 ASSUMPTIONS

Some standard assumptions are made in the following analyses:

(1) The lines-of-bearing are straight.

(2) The errors in the separate lines-of-bearing are independent.

(3) The errors in the lines-of-bearing are Normally (Gaussian) distributed with zero mean and fixed estimable variance.

(4) The emitter location estimate error is distributed as a bivariate Normal distribution.

(5) The sensor positions are known exactly.

(6) The transmitter location is fixed during the period of DF fixing.

(7) The sensors are properly sited, calibrated, and operated.

Assumption #1 is reasonable for the systems considered in this report when the sensors are properly sited. However, this assumption is weak at frequencies below approximately 30 MHz because of the effects of atmospheric tilt.

Assumption #2 is reasonable based on the systematic errors being accounted for in calibrations. This assumption is weak at frequencies below approximately 30 MHz when some stations are close enough to each other to be subjected to the same propagation effects.

Assumption #3 is usual when considering measurements which are subject to random measurement error. There are biases in the measurements from navigation errors, errors in the calibration tables, interference, depression angle effects, etc; these biases may be removed. In the absence of specific knowledge about these errors the normal assumption is reasonable. Distorting effects such as plinthing to account for wild bearings, skewedness because of low receiver signal-to-noise ratios, and distortions resulting from the sensors not uniformly surrounding the emitter can weaken or invalidate this assumption.

Assumption #4 is necessary to allow confidence levels about the estimated emitter position to be computed. The qualifications on assumption #3 also apply to #4.

Assumption #5 is reasonable based on the fact that any such position errors can be added to the emitter estimate uncertainty, if they are significant.

Assumption #6 is necessary to the analyses of the systems considered in this report, and it is reasonable over the period required to obtain a single fix.

Assumption #7 is reasonable in the absence of contradictory information.

2.3 RESTRICTIONS

In addition to the assumptions discussed in section 2.2, this report does not consider the following effects:

(1) Geographic transformation, map projection effects, and grid reference system conversions (see UAA002 Analysis of Geographic Transformation Algorithms July 9, 1982 of this series of algorithm analysis reports).

(2) Propagation effects.

(3) Centroid effects and susceptibility to deception (meaconing, gated signal parameter techniques, etc.).

(4) Special problems associated with low-probability-of-intercept emitters (low SNR, spread-spectrum, time-frequency diversity, frequency agility, etc).

(5) Numerical computation and normal truncation effects.

(6) Combination of lines-of-bearing, or emitter location estimates and their confidence ellipses from different systems (these problems will be the subject of a future report in this series of algorithm analysis reports).

(7) Elimination of wild bearings and ghost intersections using hardware/software processing of target message internals.

2.4 SCOPE

This report covers the TRAILBLAZER system as documented in ROLM 1602 Extended Assembly Language listings, marked DSO:TBSYS.SV, generated on 2/25/82.

SECTION 3

TRAILBLAZER DF FIX ESTIMATION

3.1 A GENERAL DESCRIPTION OF TRAILBLAZER DF FIX ESTIMATION

This analysis of the TRAILBLAZER DF Fixing System (a communication intelligence collection system (COMINT)) is based on the TRAILBLAZER AN/TSQ-114, Operator's Manual TM 32-5811-022-10-1, and assembly language listings dated 2/25/82. There are some questions as to whether the manual and the listing correspond to the same version of TRAILBLAZER, of which there are several.

TRAILBLAZER is a ground-based, computer-assisted COMINT DF Fixing System consisting of five sensors: two master control stations (MCC) and three remote slave stations (RSS). TRAILBLAZER can obtain relatively accurate fixes in the "normal fix mode" with as few as three operational sensors. Less reliable fixes (cuts) can be obtained in the "degraded fix mode" using multiple lines-of-bearing from each of only two sensors.

Figure 3-1 depicts the most desirable siting of the five stations of a TRAILBLAZER system. This layout allows for a maximum DF base line consistent with maintaining the required data-links between the sensors. Over flat terrain the penetration of the system is about 15 to 20 km.

Figure 3-2 indicates extended penetration ranges possible when the system is operated from elevated vantage points.

TRAILBLAZER's five sensors can obtain up to five lines-of-bearing (LOBs) simultaneously on a desired emitter(s) and place this set of LOBs in one of up to five available bins (arrays). Each bin may contain up to five sets of LOBs on same or different emitters. Wild LOBs may be edited (rejected) by one of the system operators based on actual content.

The system operates in a multifix (automatic) mode and in a single fix (manual) mode. Since the single fix mode amounts to the first pass in the multifix mode it will not be discussed separately.

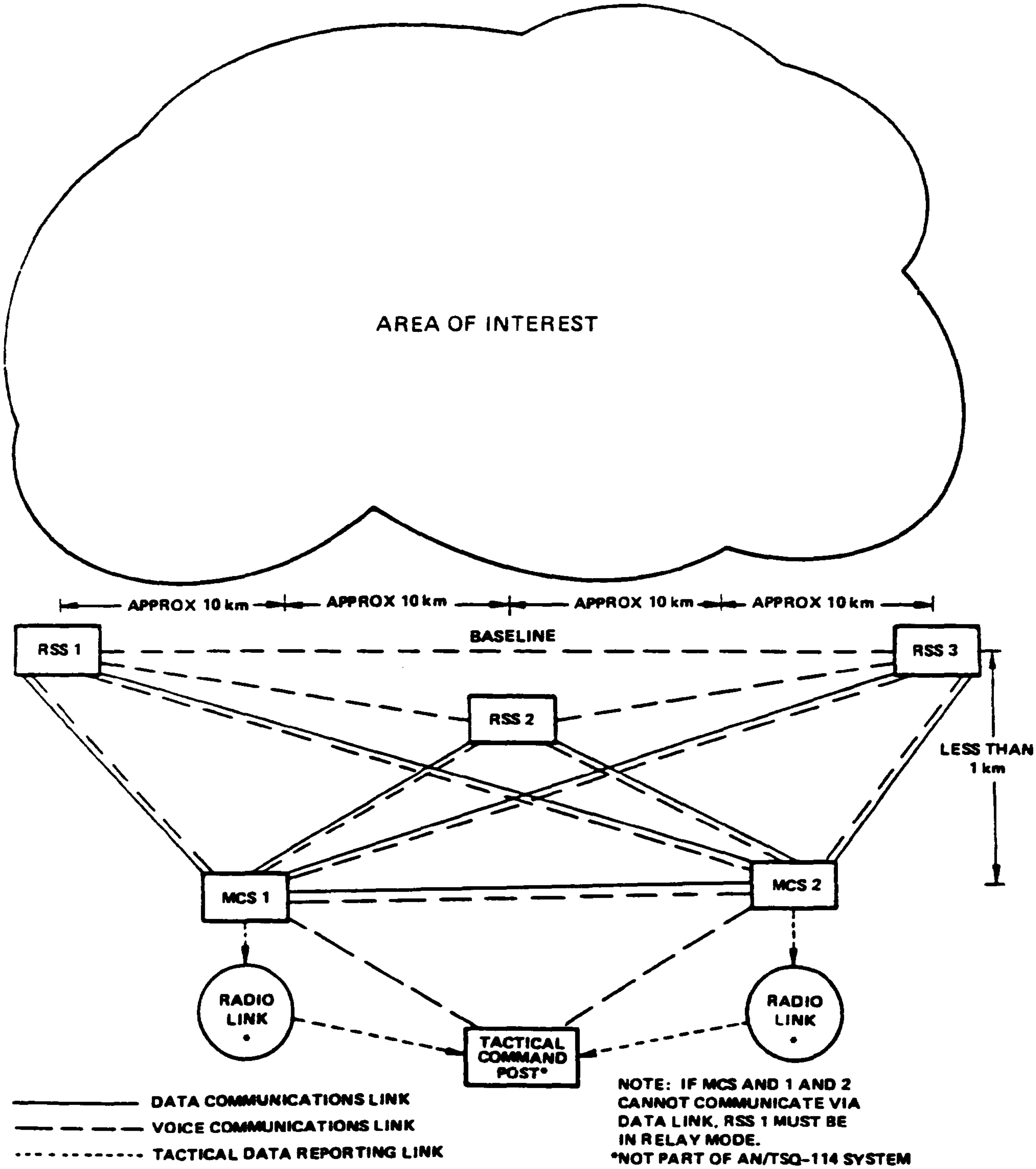

Figure 3-1. TRAILBLAZER Deployment

From: U.S. Army Field Manual (FM 30-476), Radio Direction-Finding, 8 April 1977

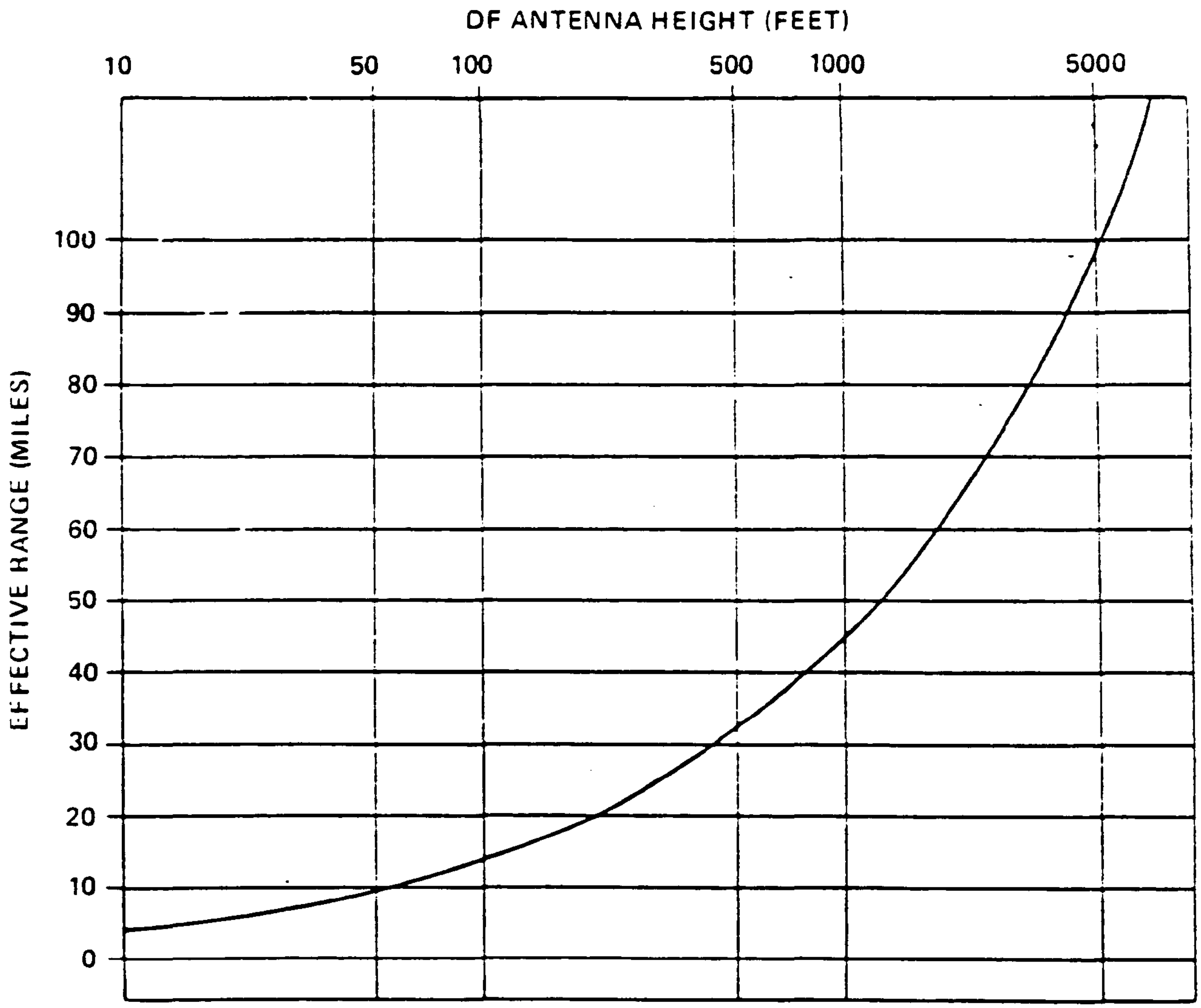

Figure 3-2. Normal Radio Range vs Antenna Height

From: U.S. Army Field Manual (FM 30-476),
Radio Direction-Finding, 8 April 1977

3.2 SIMPLIFIED DESCRIPTION OF TRAILBLAZER DF FIXING PROCESS

The following description is based on obtaining a DF fix using one set of five LOBs in one bin (array). Since there are five LOBs, there can be up to 10 possible intersections of two LOBs.

$$C_r^n = \frac{n!}{r!\,(n-r)!}$$

is the number of combinations of n things taking r at a time.

Each of these intersections represents an initial fix candidate. The "best" intersection must be selected, and any obviously "wild" (extraneous) LOBs must be discarded (edited). Figure 3-3 represents a set of five LOBs with the intersections numbered for reference.

The first step is to edit any wild bearings. Since LOB from station five does not form any intersections near the cluster of the other intersections (see Figure 3-3), it will be edited from the set of LOBs as a wild bearing (Figure 3-4). "Ghost" or "phantom" intersections from the geometry of the LOBs should also be edited (Figure 3-5). These are inadvertent crossings of LOBs and not relevent to the fix estimation process.

Next, each of the remaining intersections is evaluated to choose the "best" one. The "best" intersection is determined by considering which intersection is best supported by the other LOBs. For a LOB to support a fix estimate the "exactness of the LOB" from the station to the estimated fix is determined. If the angular difference between the two LOBs is excessive (greater than 3 standard deviation (sigma) units in statistical terms*) the station's LOB is considered to be a wild bearing and is discarded. Otherwise, it is considered a supporting LOB. The number of supporting LOBs is noted for each intersection. The intersection with the greatest number of supporting LOBs is selected as the initial fix estimate. Figures 3-6 and 3-7 show how the supporting LOBs are determined.

*± one sigma about the average value of the LOBs will usually contain about 68% of all the LOBs. ± 3 sigma corresponds to about 99.7%.

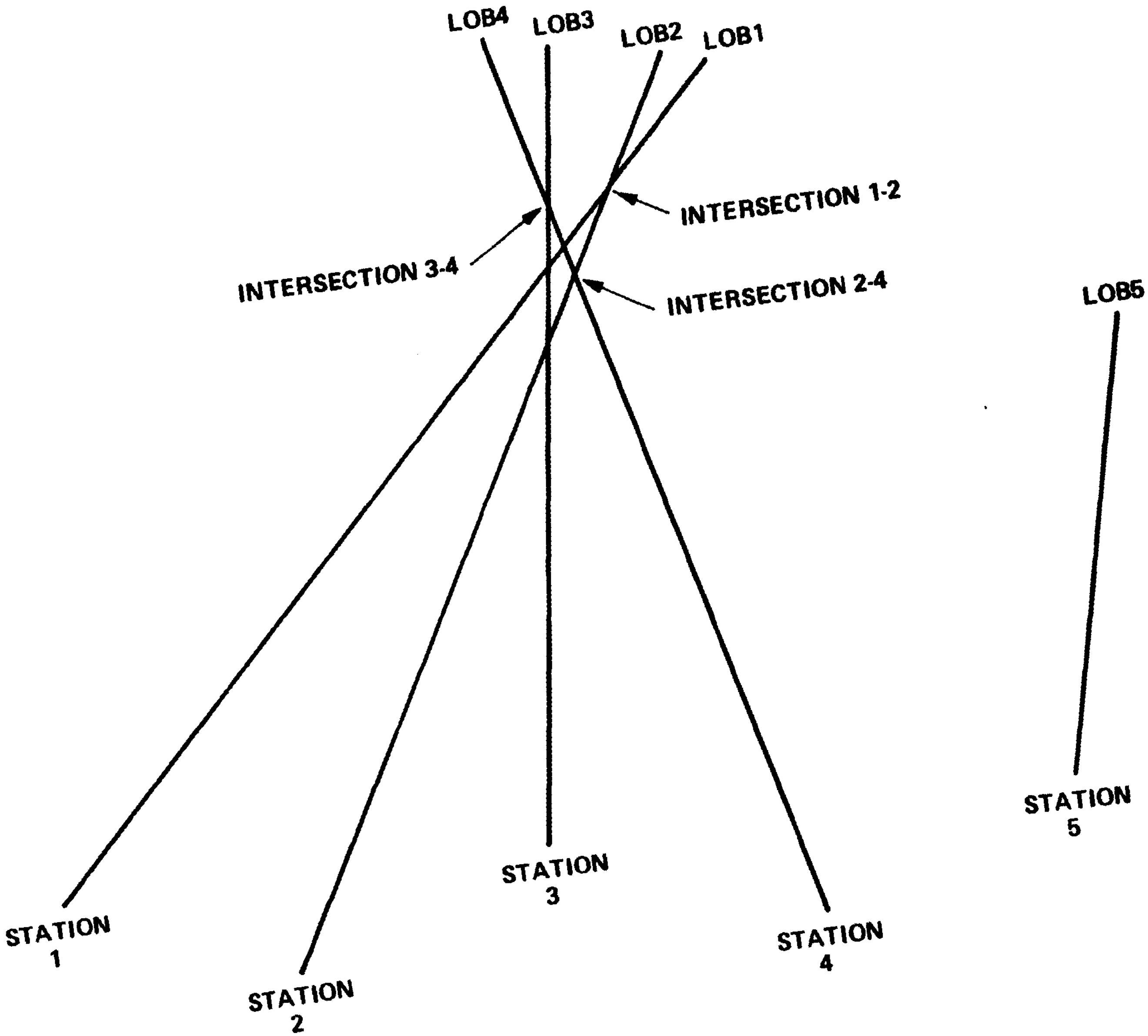

Figure 3-3. Set of Five LOBs Showing Intersections

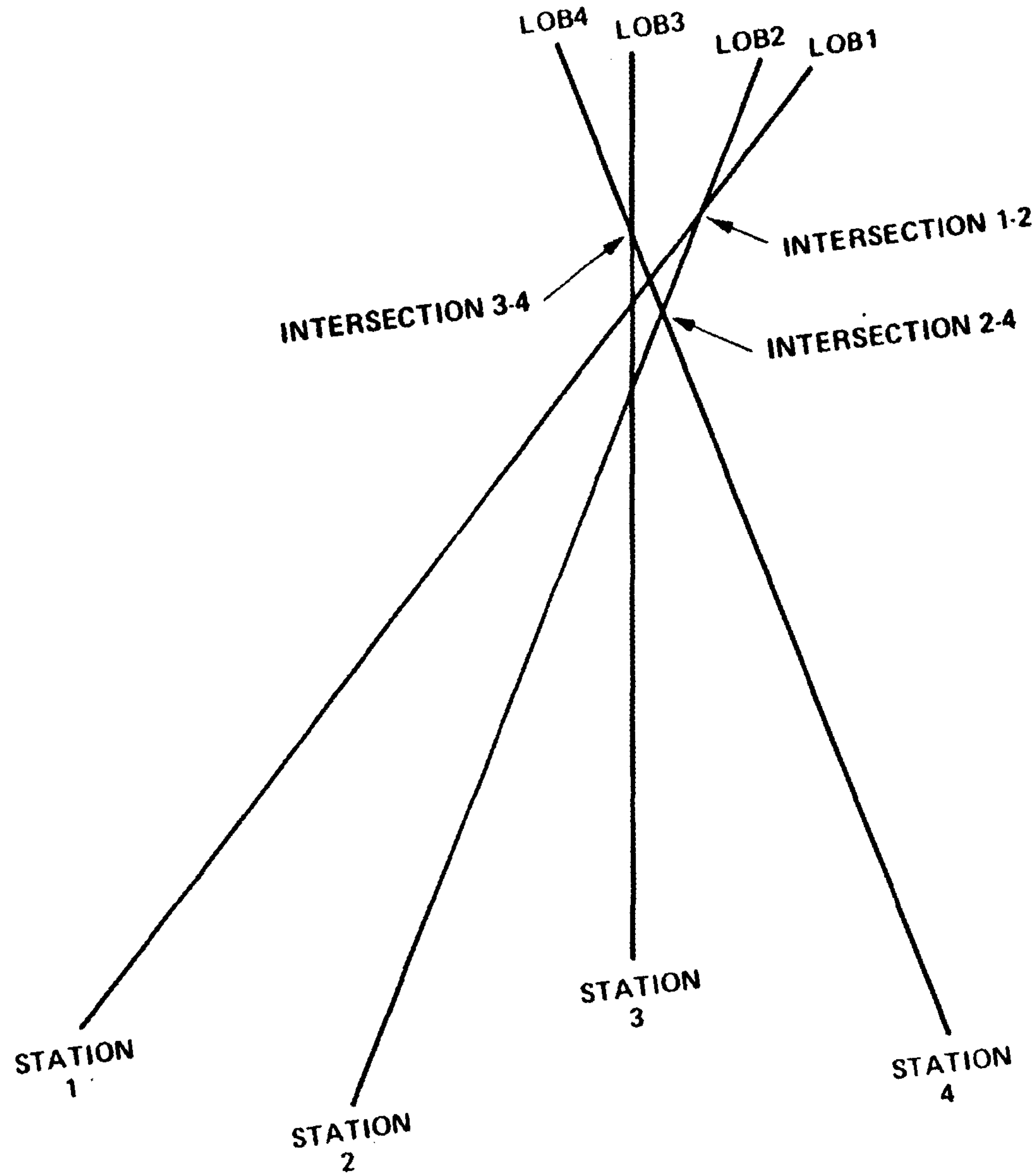

Figure 3-4. Set of LOBs Showing Intersections After LOB5 has been Removed

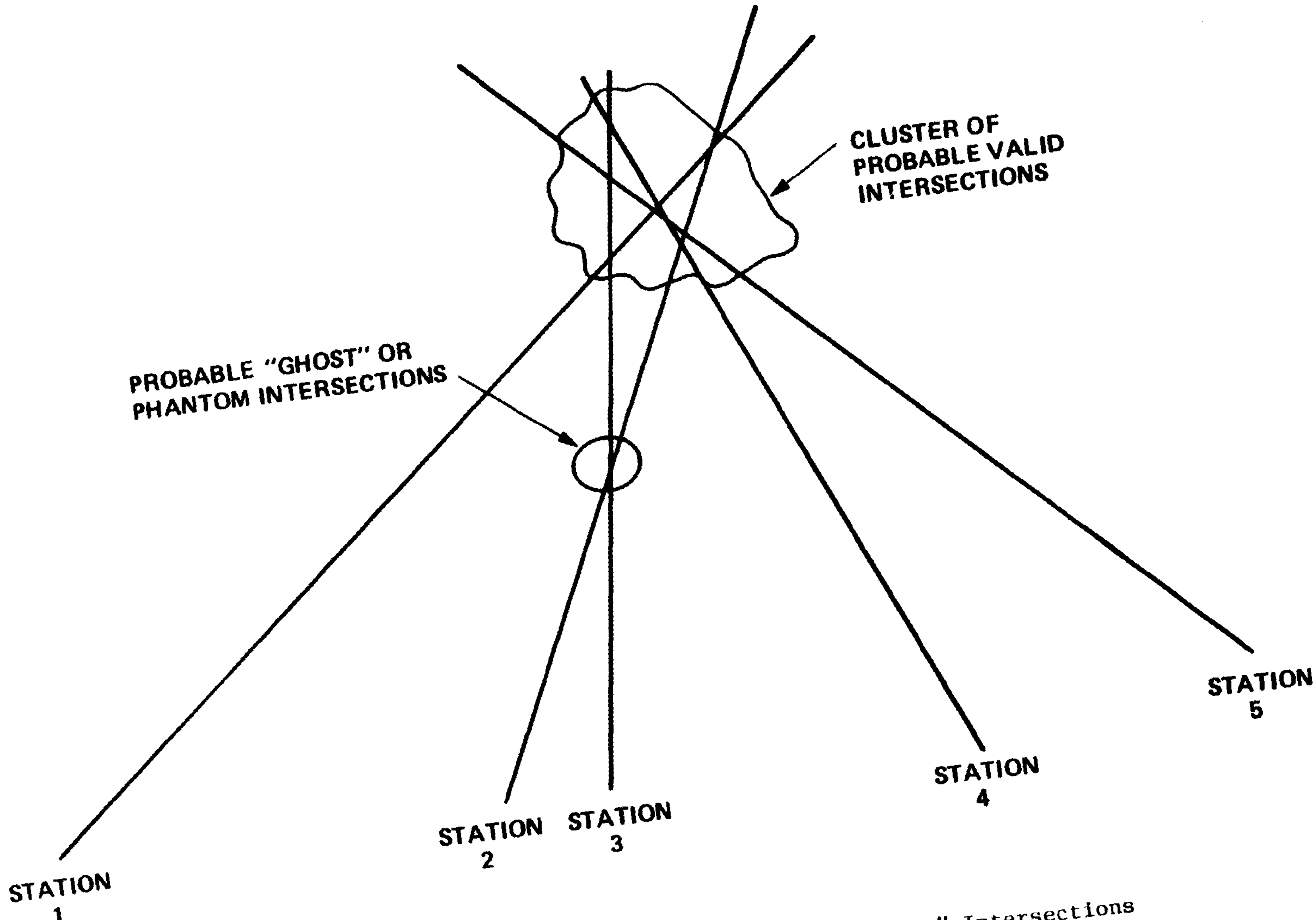

Figure 3-5. Set of Five LOBs Showing "Ghost" or "Phantom" Intersections

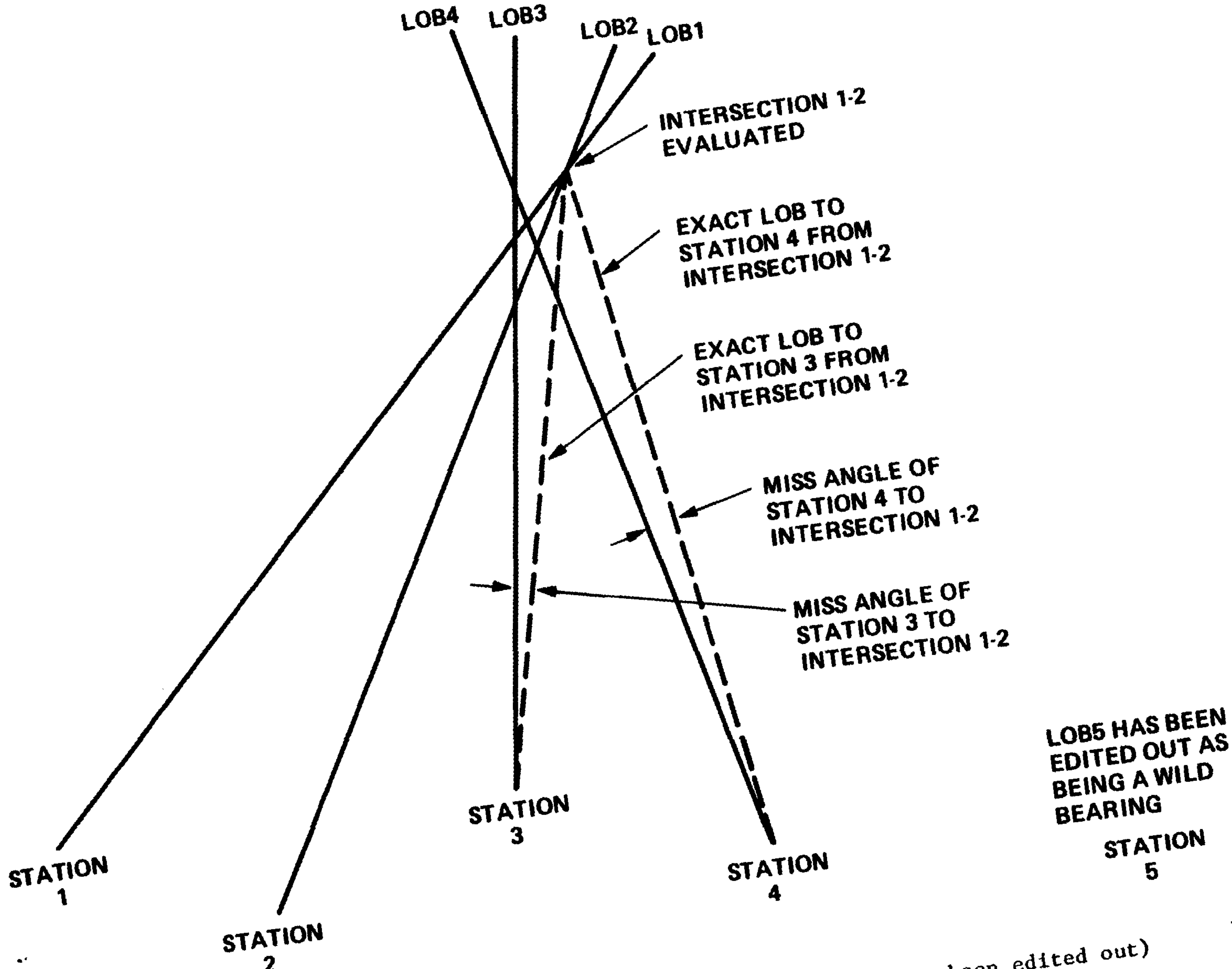

Figure 3-6. Determination of Supporting LOBs (Wild bearing has been edited out)
For Intersection 1-2

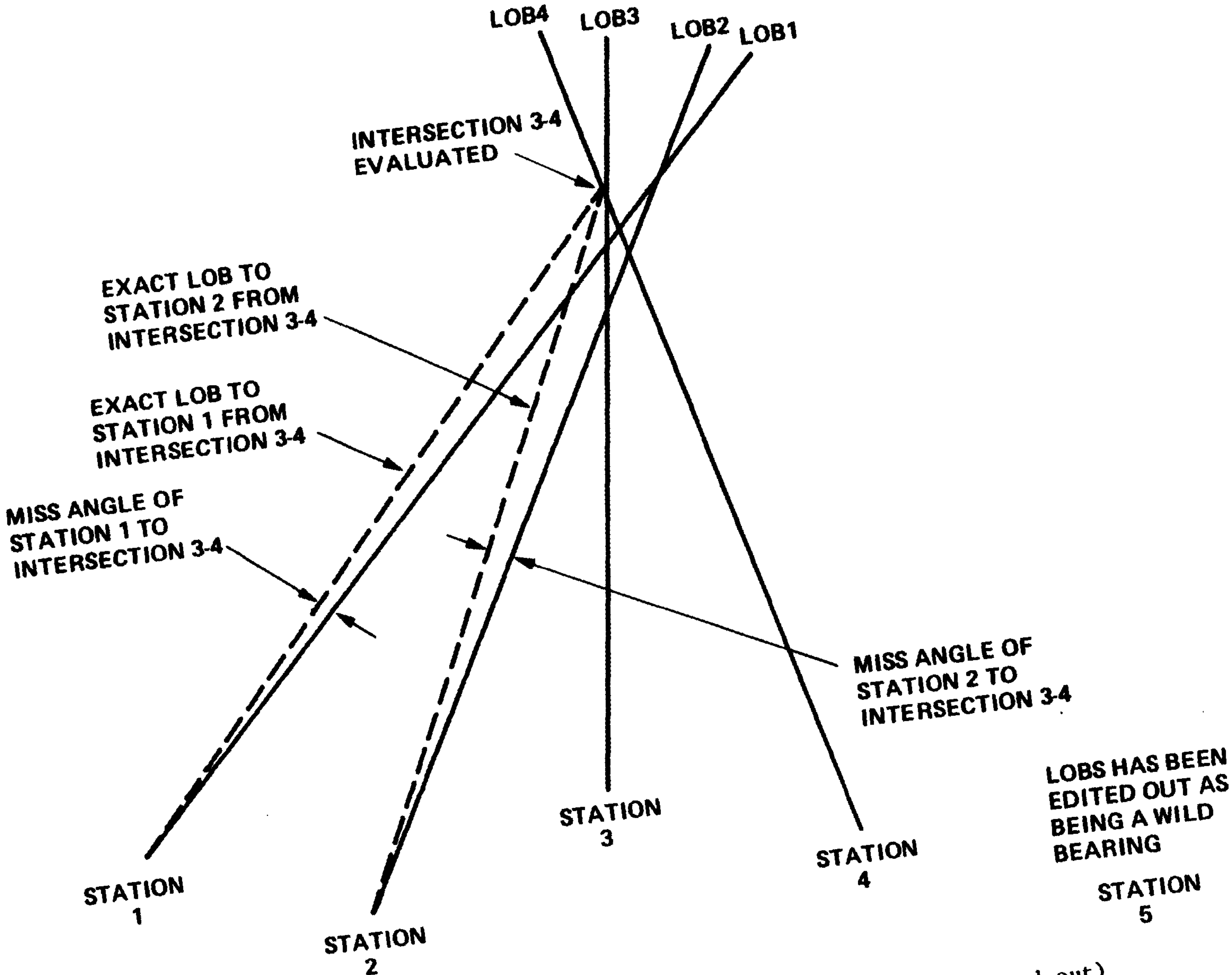

Figure 3-7. Determination of Supporting LOBs (Wild bearing has been edited out) For Intersection 3-4

The initial fix estimate is then optimized to obtain the "best" fix estimate. This optimization process uses a potential function (detailed below) as the objective function and is based on displacing the initial fix estimate systematically in four directions (north, west, south, east), and calculating the support for each displaced (trial) estimate and keeping the "best" one. This process is continued with steps decreasing in size until a "best" fix estimate is located.

The support of the trial "best" fix is calculated as the sum of the potential function weighted (which is a semi-normalized Gaussian-weighted) miss-angles between the actual LOBs and the computed LOBs to the trial location. The effect of the Gaussian weighting is to give more consideration to the LOBs associated with the smaller miss-angles.

Having found an optimized or "best" fix estimate, a "confidence" region is calculated. A "confidence" region is a region that is likely to contain the true emitter location for some percentage of all fixes on a given emitter (50% in the case of TRAILBLAZER).

The following description of TRAILBLAZER DF Fixing has been freely-adapted from the software comments. Differences between the actual code and the description of the algorithm will follow, along with comments on the methods used.

The TRAILBLAZER DF Fixing Algorithms are based mainly on heuristic and empirical reasoning, rather than purely mathematical/statistical techniques. The four main steps in the DF Fixing process are quite intermingled and relate to the following discussion as follows:

(1) Obtain initial fix estimates.

(2) Reject wild lines-of-bearing by manual screen editing and automatic rejection.

(3) Refine (optimize) the initial fix estimate.

(4) Establish a confidence region (elliptical error probable (EEP)) around the fix estimate.

3.3 DETAILED DESCRIPTION OF TRAILBLAZER FIXING

The TRAILBLAZER fix algorithm has been tailored specifically for operation with the LOB data produced by a ground-based DF network, consisting of a predetermined number of DF stations, whose locations remain invariant during the data collection process. The goal of this algorithm is its attempt to resolve multiple targets reliably, while at the same time avoiding ghosts, i.e., false targets arising from coincidental intersections of unrelated LOBs.

The TRAILBLAZER fix algorithm execution consists of three distinct processes or phases:

(1) Fix estimation (ESTMP Procedure) and wild bearing rejection (ESTMP and FINAL Procedures).
(2) Fix optimization (FPEAK Procedure).
(3) Computation of an error ellipse surrounding the established fix point (FINAL Procedure).

Figure 3-8 outlines the flow of the TRAILBLAZER DF fixing algorithm.

3.4 FIX ESTIMATION

The estimation process is the key to the fix algorithm. For an understanding of this process, the TRAILBLAZER LOB database structure must be explained. The LOB data are stored in sets of up to five LOBs, i.e., one LOB from each of five possible DF stations. A set of five LOBs results from a system response to a DF command (or from a single manual LOB entry sequence via the "demo" command). The assumption is made that to the best of the operator's judgment, the LOBs within a set are associated with a single emitter. Given that the LOBs within a set are collected simultaneously, this is a fair assumption.

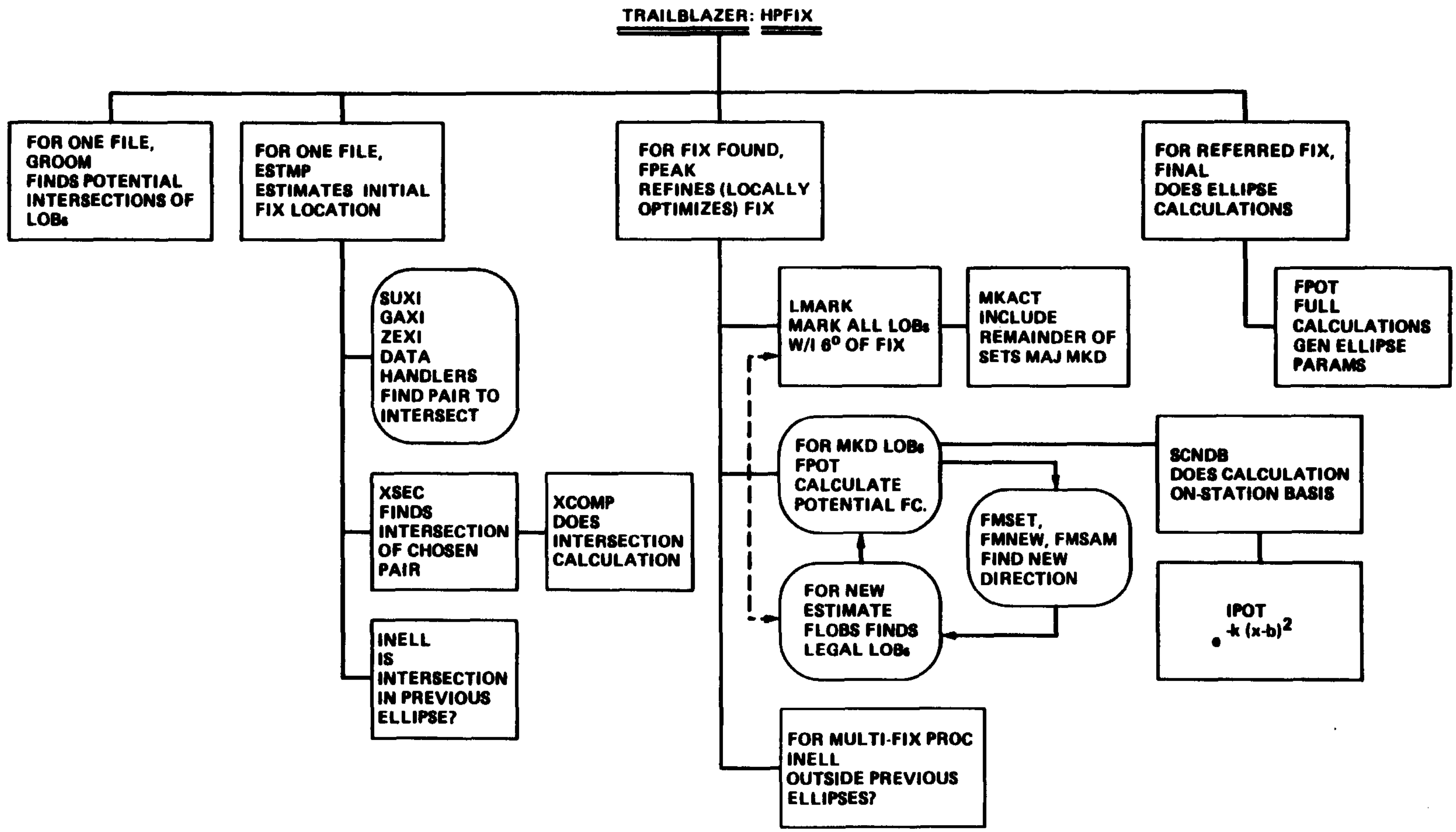

Figure 3-8. Flow of the TRAILBLAZER DF Fixing Algorithm

The objective of the estimation process is to examine all possible intersections of pairs of LOBs in the same set and all other sets, and to select that intersection which has the largest number of other LOBs in the database that miss this intersection by less than some specified miss angle. The estimation process selects that intersection which lies within the largest or strongest cluster of intersecting LOBs. It also makes a gross check to prevent duplication of previous fixes in multifix processing (as discussed in intersection criterion (2) discussed below). This process is computationally efficient because:

(1) The number of DF stations is small. Therefore, the number of intersections to be examined per set is reasonable.

(2) The locations of the DF stations are fixed. Consequently, for miss angle computations at each intersection, there are at most only three other exact LOBs that need to be computed, since two are already used for the intersection.

Each possible intersection of two LOBs is calculated directly from the geometry indicated in Figure 3-9.

The calculated intersection is validity-checked by:

(1) Verifying a true intersection, i.e., D_1 and D_2 both positive.

(2) Verifying that the angular difference between the LOBs is sufficient $LOB_2 - LOB_1 \geq 0.6$ degree.

(3) Verifying that the minimum and maximum range limitations are not exceeded for either station, $0.5 \leq D_i \leq 100$ km.

(4) Verifying that the intersection does not fall within the error ellipse of any previously computed fix (in the multifix mode only).

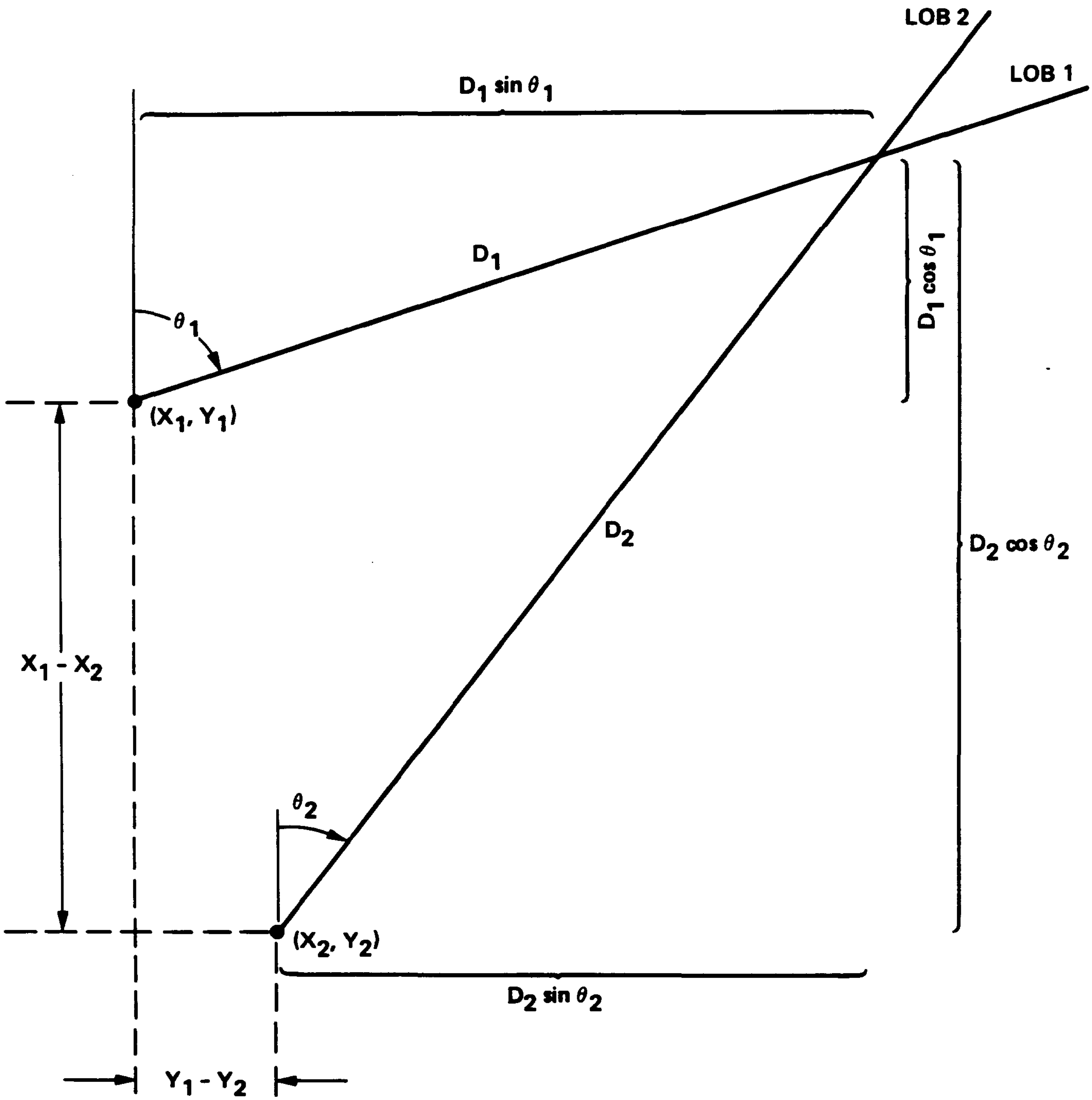

Figure 3-9. Intersection Geometry

Supporting LOBs are calculated from all the available (unused in any previous fix optimization) LOBs over all the bins. These supporting LOBs are calculated as indicated in Figure 3-10 and must fall within 3 sigma of the actual intersection to be considered as supporting it. The actual value of sigma is confused in the available source code listings. It is stated to be 2 degrees in the source code comments, but assigned a value of 8 degrees in parts of the source code.

In order for an intersection to be a candidate for valid fix estimate, an intersection must satisfy the following criteria:

(1) The intersection is real, i.e., the absolute value of the difference between the LOBs is at least 0.6 degree and the directed LOB vectors must intersect. The source code does not verify that only forward-looking LOBs are considered for intersections. It is possible, in running the program, to allow reciprocal bearings and create "phantom" intersections. This, however, is unlikely because of the normal deployment geometry. Also, the intersection must be in the range of 0.5 to 100 km of the reporting stations. The source code implementation of intersection out of range fails and would loop infinitely because of an initialization problem.

(2) If this is not the first pass for fix processing, i.e., a multifix (as opposed to single fix) situation looking for multiple emitters, the intersection must fall outside the error ellipse of the immediately previous successful fix estimate (this is because all previous fix error ellipses are checked in the optimization process). This provides the capacity to resolve multiple targets.

(3.a) In the normal fix mode (as opposed to degraded) the intersection must be supported by LOBs from at least three stations. The supporting LOBs must be within a ±3 sigma miss angle of the three exact LOBs from their

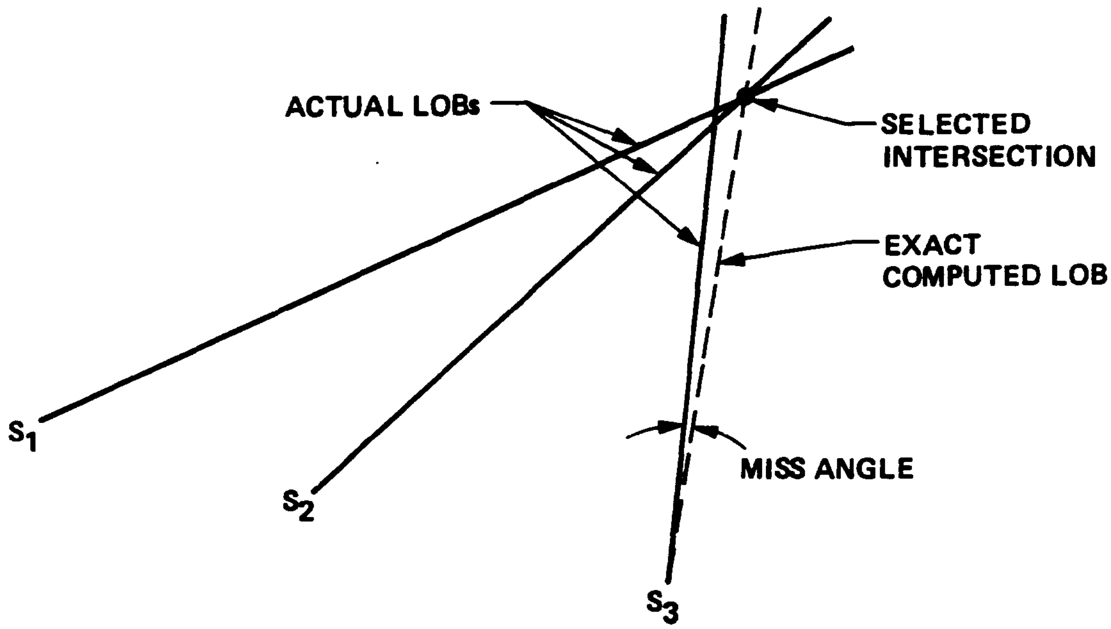

Figure 3-10. Supporting LOB Geometry

stations to the intersection. Also the supporting LOBs belong to a set of LOBs not used in any previous fix optimization computation, and in which the majority of LOBs conform to this miss angle requirement.

(3.b) In the degraded fix mode, the intersection must be supported by more than one set of intersecting LOBs from two stations whose LOBs were used in the computation of the intersection. The supporting LOBs must be within ±3 sigma of the exact LOBs from their stations to the intersection. Furthermore, these LOBs must belong to a set of LOBs not used in any previous fix computations in which the majority of LOBs conform to this miss angle requirement. The degraded mode is used only if the estimation process for the normal fix fails to yield a valid estimate which would satisfy the three station criteria.

(5) Of all the possible intersections, this intersection is supported by the largest number of LOBs that meet the miss angle and set requirements mentioned in the above criteria.

The estimation process yields a "non-ghost" intersection which is the best fix estimate (lies within the strongest clustering of LOBs). Also in the multiple fix case, the best fixed estimate lies outside the error ellipse of the previous fix. In addition, by discarding invalid intersections, the amount of computations in subsequent estimations (in the multitude fix case) is considerably reduced. Thus, for the multiple fix case, the number of computations is reasonably bounded.

In the case of multiple target examination, once the fix mode has been degraded, it remains degraded for all subsequent passes. Also, failure to obtain a fix estimate in the degraded mode precludes further passes.

3.5 FIX OPTIMIZATION

The fix optimization process seeks to improve the fix estimate by finding that location which locally maximizes a multipeaked objective function. By the nature of the estimation process, the initial estimate should be fairly close to the optimum fix location. This process also performs a final check to eliminate the duplication of previous fixes in the multifix situation.

At the outset of the optimization process, an estimate is available along with the exact LOBs associated with it from each system station. Also available are the LOBs supporting this estimate as a valid fix estimate. Before beginning optimization, that portion of the database which supports a particular fix estimate is modified to include complete sets when a majority of the LOBs supported the estimate. This increases the potential number of peaks and ridges in the objective function.

The optimization is then performed by systematically searching for a local peak in the total objective function which is a potential function of the form

$$\sum\sum \exp\ [-Ka_{ij}^{2}(x,y)]$$

where the outer sum is over the stations, the inner sum over the LOBs from each station and a_{ij} is the miss angle between the actual and computed LOB (see Figure 5-8). This objective function is applied locally in each case by: (1) using the selected portion of the database, (2) using the fix estimate as an initial reference location, (3) using its associated set of exact LOBs, and (4) using the computed potential function for the point.

The peak searching scheme is a fairly conventional pattern search optimization method (Jacoby, 1972; Gill, 1980) but without a pattern step directed along the steepest gradient. The reference location is displaced by some step size (initially 16 screen raster (resolution) points) along the axes in the following four directions: +Y,+X,-Y,-X. Only one direction is considered at a time, generating a trial location. If one of these trial locations yields a higher value of the potential function, the trial location becomes the reference, an associated set of exact LOBs is determined, and a new trial location in the same direction is attempted. This peak search scheme continues until no further improvement in the same direction can be made and the process has been repeated in all directions. Figure 3-11, Parts I through IV illustrate the fix estimate optimization process. At this point, the step size is halved and the four directions are tried again. The procedure stops either when the step size becomes too small (currently less than one raster point in screen geometry), or after a maximum number of successful improvements have been made (currently 16). Under these restrictions the process is nondivergent, that is, it stops.

Although this algorithm always stops, and gives a value, it may not converge in the sense that the final location estimate is substantially closer to the location of the nearest local peak than was the starting value. One major cause is that the objective function itself may not be sharply peaked, and may even have ridges. Thus, since the optimization algorithm does not have a pattern step, and it does not rotate axes to take advantage of the gradient, it may climb very slowly. Three other factors compound this behavior.

(1) Angles are rounded to quarter degrees.

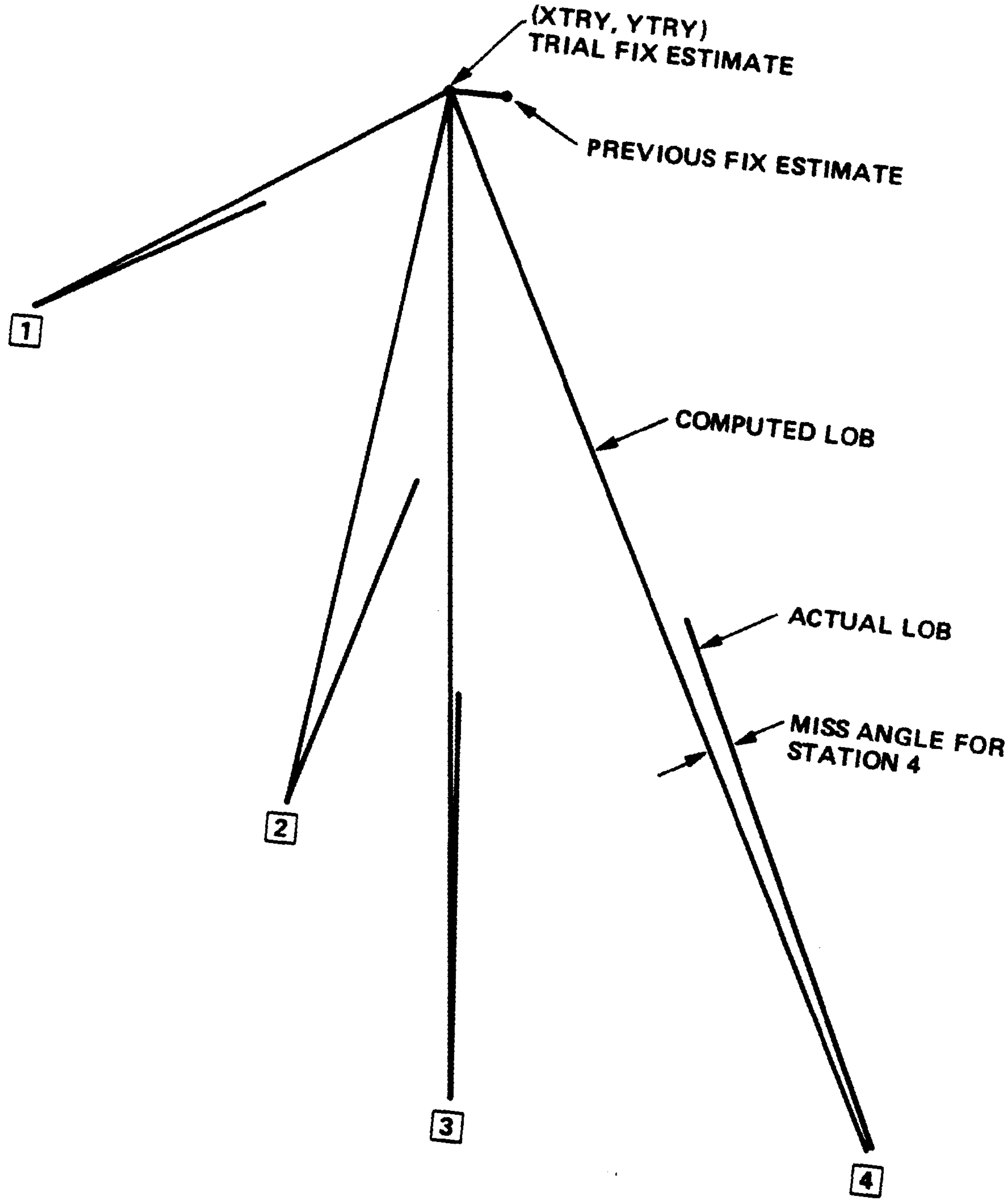

Figure 3-11. Fix Estimate Optimization, Part I

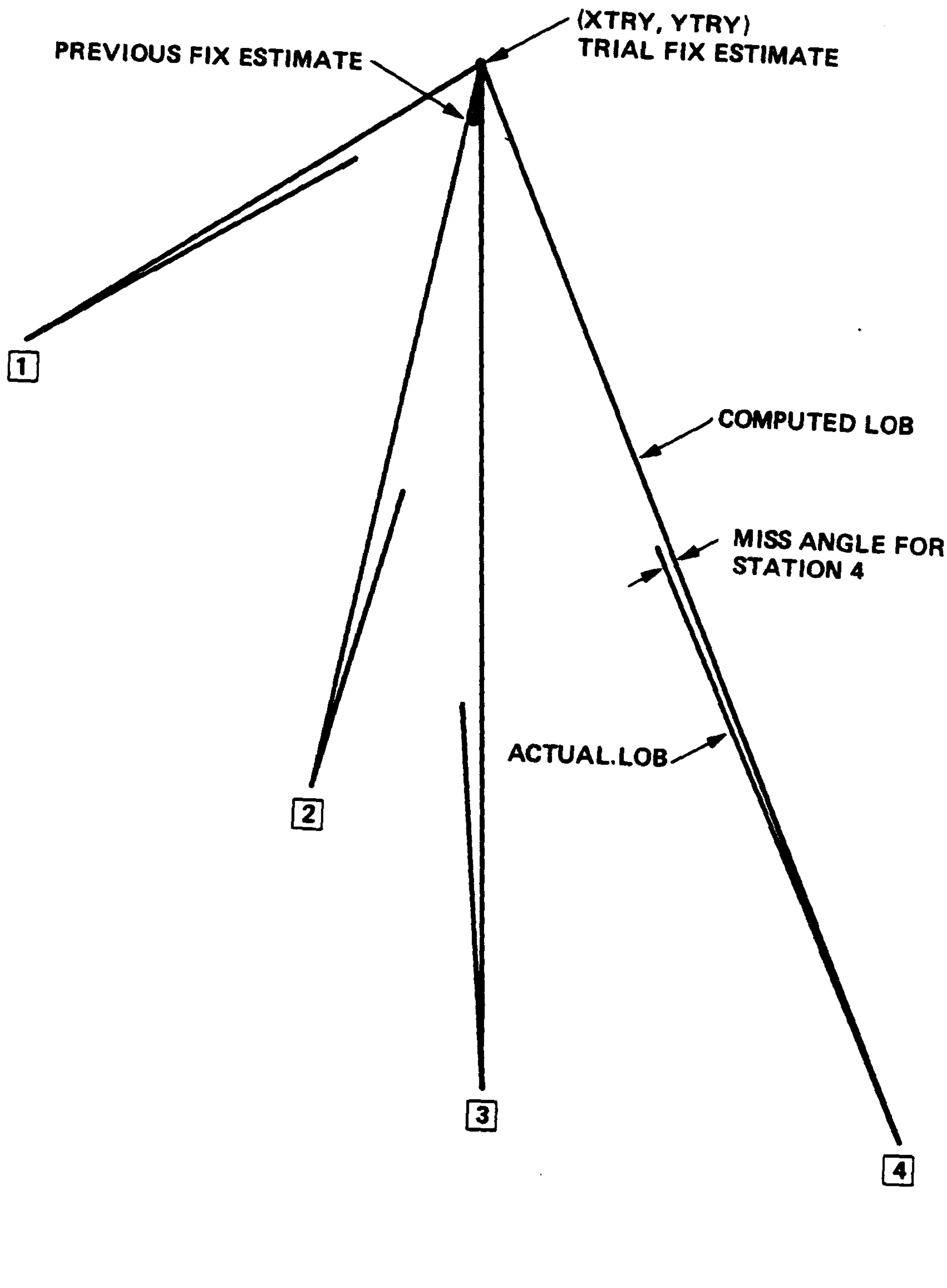

Figure 3-11. Fix Estimate Optimization, Part II

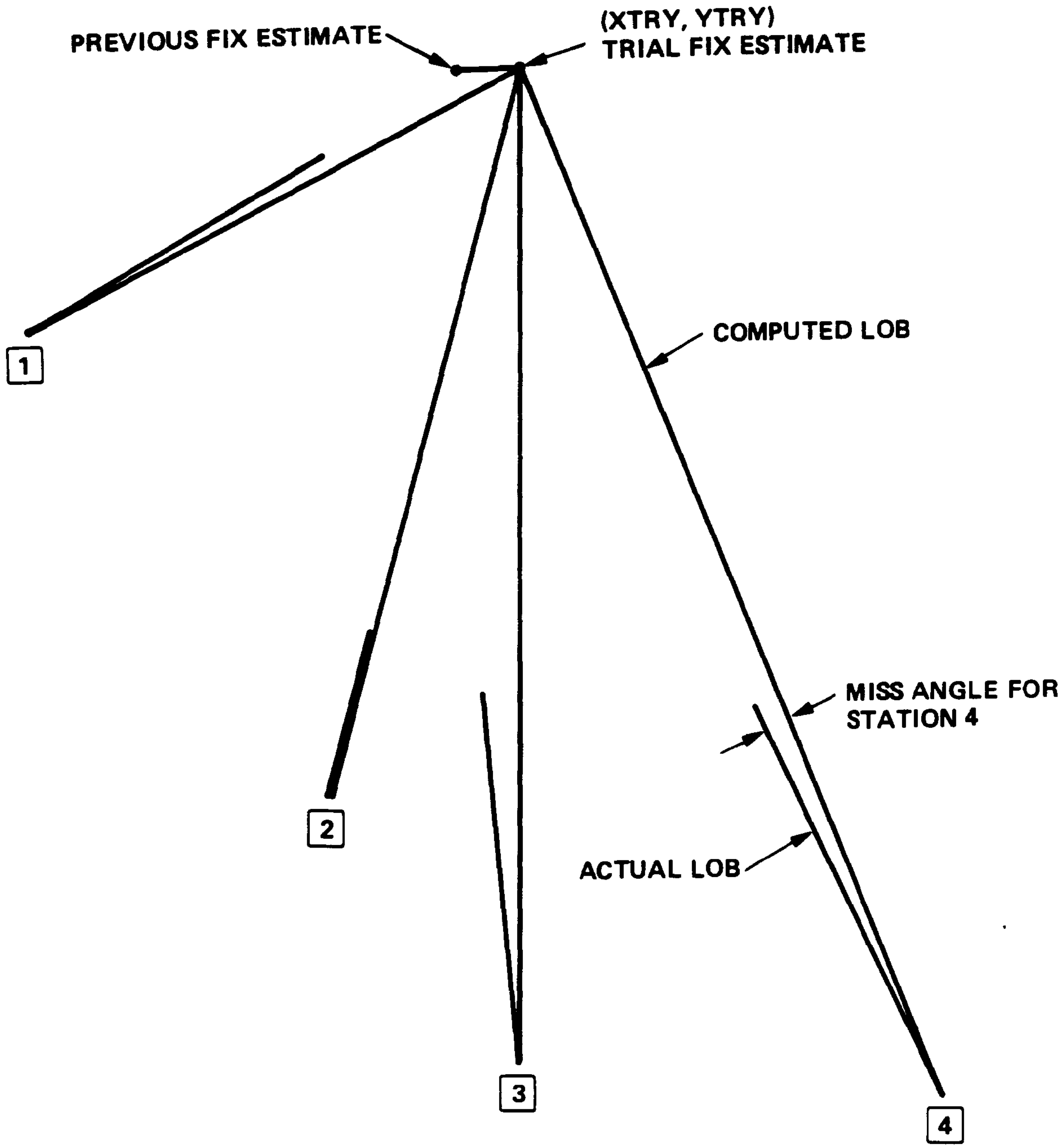

Figure 3-11. Fix Estimate Optimization, Part III

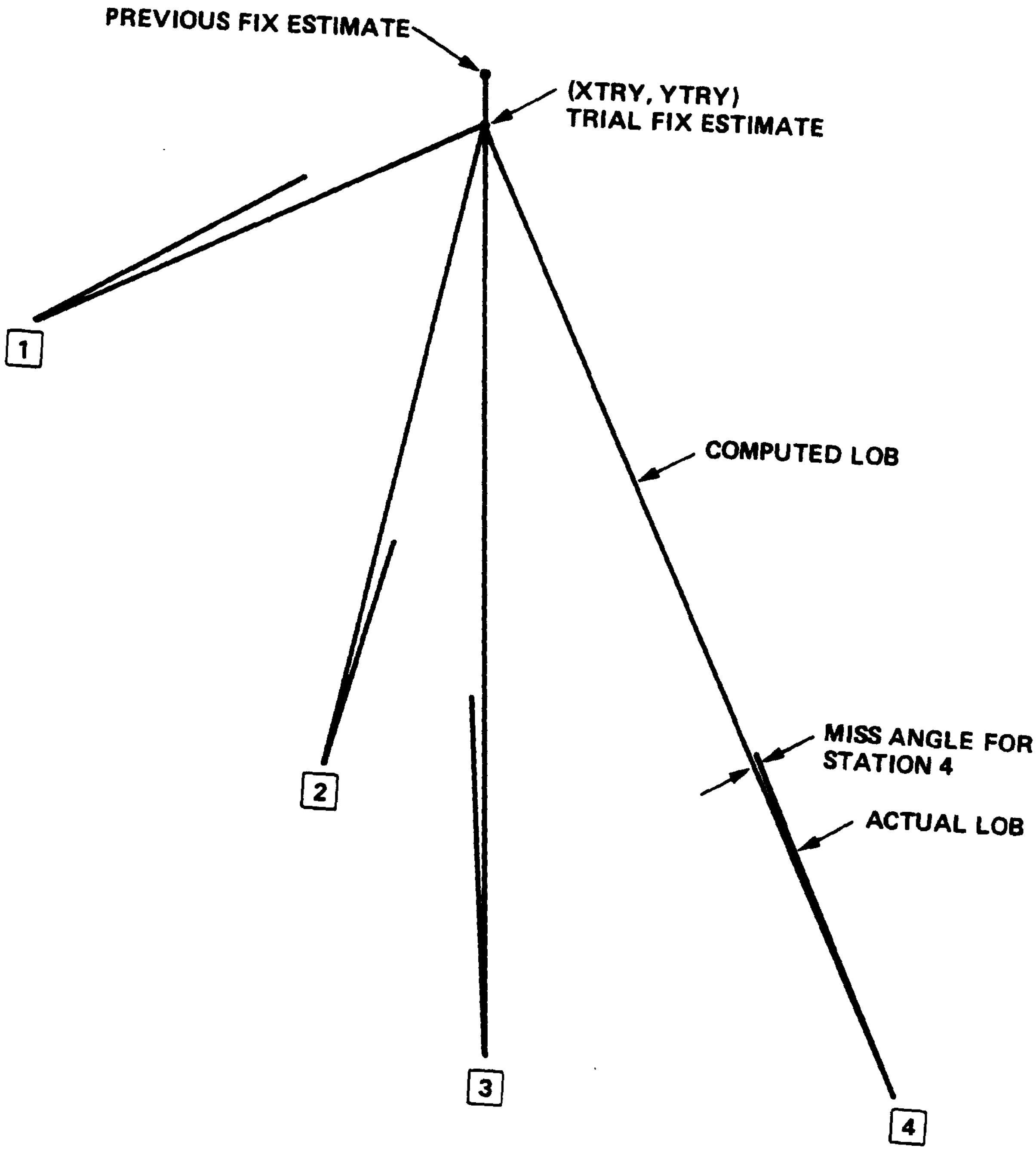

Figure 3-11. Fix Estimate Optimization, Part IV

(2) Only the first four terms of the Taylor expansion for the exponential function are used in calculating the objective function.

(3) The algorithm terminates after, at most, 16 steps.

Therefore, as often occurs in purely empirical methods, the location estimate may be fairly inaccurate and thus not well-defined.

In the multifix situation (multiple passes), the final location (a result of peak searching), is checked to determine whether the emitter lies outside the error ellipses of all previous fixes. If not, this optimized fix is unacceptable and a new fix estimate must be obtained. Note that if this occurs, the selected portion of the local database that gave rise to this unacceptable optimized estimate is removed from further consideration.

If the final location is acceptable, what remains is the computation of the parameters in an error ellipse surrounding the optimized fix estimate. The data available at this point is: (1) a selected subset of the LOB database, (2) a final optimized fix location, (3) a set of exact LOBs to that location from all system stations, and (4) and the value of the potential function for that location. This procedure is satisfactory for ellipses with small eccentricity. However, it degrades with the higher eccentricity ellipses arising from emitter ranges that are large with respect to the sensors' baseline. Also, just as it is unclear that the best fix estimate is a measure of central tendency for a known bivariate distribution, it is equally unclear that the calculated ellipse reflects a related measure of dispersion. The statistical properties of these estimates is important because intelligence processing systems to which these values are input data assume they are the estimated mean and elliptical error probable from a bivariate normal distribution.

3.6 FIX ERROR ELLIPSE COMPUTATION (CONFIDENCE REGION)

The error ellipse computation serves a dual purpose. First, its computed parameters are used in the checks to prevent duplications of previous fixes. Second, it is part of the information describing a fix.

The parameters computed are:

(1) The ellipse orientation angle of the semimajor axis in radians. This is with respect to true north, and is defined as the mean of all LOB angles used.

(2) The ellipse semi-major axis is denoted: $a = s/(tau_E)^{1/2}$, and semi-minor axis is denoted: $b = s\ (tau_E)^{1/2}$, where s is the ellipse size, and tau_E the ellipse axis ratio (semi-minor/semi-major).

In more detail, the size s is directly proportional to the root-mean squared miss angle of all LOBs used in fix computation. It is also directly proportional to the mean (potential weighted) station to the target distance. It is inversely proportional to a weak function of a number of LOBs used. This dependence is and should be weak because the LOB errors in a ground-based system are primarily due to propagation path perturbation by terrain. These errors are not zero-mean and not uniformly distributed.

It should be noted that:

(1) The RMS miss angle is forced to be no less than the system instrument accuracy of two degrees. The actual value of the rms value is confused in the available source code listings from 2 to 8 degrees.

(2) The step size itself is forced to be at least 0.5 km.

(3) The step size is tripled when the fix mode is degraded. This is based on the consideration that with only two stations providing LOBs, there is no indication of the propagation path perturbation effect; hence, a two-station fix location is of questionable value.

The ellipse axis ratio semiminor/semimajor is defined as the mean absolute deviation of all LOBs used from the mean LOB vector, i.e., the ellipse orientation angle divided by 45 degrees.

Note that when these error ellipse parameters are used, they are used to ensure that a fix estimate does not duplicate a previously determined fix. The criterion employed requires that the fix estimate be outside the previous fix error ellipse.

3.7 COMPUTATIONAL AND OPERATIONAL OBSERVATIONS

The TRAILBLAZER fix algorithm involves a considerable amount of computation, particularly in the estimation process. The first fix estimate requires examination of every possible intersection in every set of LOBs. If in the first estimation, all invalid intersections are removed from future consideration, the amount of computation in subsequent estimations (in a multiple fix case) is considerably reduced. In addition, once an estimate is optimized, the size of the LOB database is reduced since a set of LOBs is used only once in computing a final fix location. The fix estimate is then removed from further consideration. Consequently, the amount of computation in a multiple fix case is reasonable.

Since the entire fix calculation methodology is highly empirical, numerical parameters must be chosen with care. There are two key parameters: (1) the allowable miss angle used in the estimation process (currently 3 sigma), and (2) the width constant for the potential function (currently $(1/3 \text{ sigma})^2/2$). These parameters control the sensitivity (resolution) of the fix algorithm, and indirectly affect the error ellipse size (both the RMS miss angle and mean range are potential-weighted). If a change to these parameters is contemplated, the comments in the code recommend that their interrelationship remain reasonably the same. For example, the parameters to be changed should be multipliers of 3 sigma. For TRAILBLAZER, the current parameters give reasonable results according to the field test data (after removing obviously bad data attributed to hardware malfunctions, interference, operator mistakes, etc.). These results indicate that the error ellipse computed is about a 50 percent confidence ellipse. A considerable amount of testing and field experience would be required in order to optimize the parameters, given the indeterminate character of LOB error statistics for ground-based DF. It should be emphasized that this is a highly empirical process.

In view of the capabilities of the TRAILBLAZER system for LOB set separation at collection time using bin assignments and subsequent LOB editing, the fix algorithm should be used in the single-fix or manual mode for best results. However, when no such separation has been made or is possible, or when the operator does not have the time for post-collection LOB editing, the fix algorithm may be used in the multifix or automatic mode with a reasonable expectation of comparable results.

SECTION 4
OBSERVATIONS AND CONCLUSIONS

It appears that the assembly language source code that this analysis was based on might not be the latest version. Most of the problems indicated in Section 3 with bold-type underlining could all be due to an early version of the software in transition. The fact the system is deployed and operating tends to support the feeling that the analyzed software was not the current version.

APPENDIX A

A.1 ANNOTATED REFERENCE LIST

The references listed in this appendix fall into two categories: (1) books on general mathematics, (2) books and articles on direction finding techniques. The general mathematics books are included to better acquaint users with the necessary mathematical and technical background. They include Schaum's outline series which provides good examples, some introductory undergraduate level references, and more specialized and advanced text and references.

A.2 SCHAUM'S OUTLINE SERIES - SELECTED UNDERGRADUATE LEVEL OUTLINES

These outlines are valuable for obtaining an overview of selected subjects quickly. Explanatory text is developed along with fully solved examples in stand-alone, easily referenced blocks. The most current edition is not always referenced. The publisher is McGraw-Hill, New York.

Ayres, Frank, Jr. Plane and Spherical Trigonometry. 1954.

Ayres, Frank, Jr. First-Year College Mathematics. 1958.

Ayres, Frank, Jr. Matrices. 1962.

Ayres, Frank, Jr. Calculus. 1964.

Lipschutz, Seymore. Analytic Geometry. 1968.

Lipschutz, Seymore. Probability. 1968.

Rich, Barnett. Plane Geometry with Coordinate Geometry. 1963.

Scheid, Frances. Numerical Analysis. 1968.

Spiegel, Murray R. Statistics. 1961.

Spiegel, Murray R. Advanced Calculus. 1963.

Spiegel, Murray R. Probability and Statistics. 1975.

A.3 INTRODUCTORY UNDERGRADUATE TEXTS

Acton, Forman S. Numerical Methods That Work. Harper and Row, New York, 1970.

Dixon, Wilfrid J. and Massey, Frank Jr. Introduction to Statistical Analysis. Second edition, McGraw-Hill, New York, 1957.

Hamming, Richard W. Introduction to Numerical Analysis. McGraw-Hill, New York, 1971.

Hoel, Paul G. Elementary Statistics. Third edition, John Wiley and Sons, New York, 1960.

Hohn, Franz E. Elementary Matrix Algebra. Second edition, Macmillan, London, 1964.

Kells, Lyman M., Kern, Willis F., and Bland, James R. Plane and Spherical Trigonometry. McGraw-Hill, New York, 1940.

Kreyszig, Erwin. Introductory Mathematical Statistics. Wiley, New York, 1970.

Middlemiss, Ross R. Analytic Geometry. Second edition, McGraw-Hill, New York, 1955.

Pettofrezzo, Anthony J. Elements of Linear Algebra. Prentice-Hall,Inc., New Jersey, 1970.

Steinberg, David I. Computational Matrix Algebra. McGraw-Hill, New York, 1974.

A.4 SPECIALIZED REFERENCES

Ballard, Thomas B. and Hebbert, R. Scott. A Tracking Algorithm Using Bearing Only. Naval Surface Weapons Center, White Oak, Silver Spring, MD, October 1975.

Barfield, R. H. Statistical Plotting Methods for Radio Direction-Finding. J. IEEE, Vol. 94, Part IIIA, 1947.

Beale, E. M. L. Brooke Variance Classification System for DF Bearings. Journal of Research of the National Bureau of Standards D. Radio Propagation Vol. 65D, No. 3. May-June 1961.

Beale, E. M. L. Estimation of Variances of Position Lines From Fixes with Unknown Target Positions. Journal of Research of the National Bureau of Standards D. Radio Propagation Vol. 65D, No.3. May-June 1961.

Blachman, Nelson M. Position Determination from Radio Bearings. IEEE Transactions on Aerospace and Electronic Systems. May 1969.

Brown, Ronald Max. Emitter Location Using Bearing Measurement from a Moving Platform. Naval Research Laboratory, Washington, DC, June 1981.

Butterly, Peter J. Position Finding with Empirical Prior Knowledge. IEEE Transactions on Aerospace and Electronic Systems. Vol. AES-8, No. 2. March 1972.

Clark, B. L. A Comparative Evaluation of Several Bearings-Only Trackers. Naval Surface Weapons Center, Dahlgren, VA, 1980, AD-B051662.

Cooper, D. C. Statistical Analysis of Position-Fixing General Theory for Systems with Gaussian Errors. Proc. IEE, Vol.119, No.6. June 1972.

Cooper, Leon and Steinberg, David. Introduction to Methods of Optimization. W. B. Saunders Company, Philadelphia, London, Toronto.

Daniels, H. E. The Theory of Position Finding. The Journal of the Royal Stat. Soc., Series B, Vol. XIII, No.2. 1951. pp.186.

Demetry, James S. Estimation Algorithms for Location of Stationary Radiation Sources by Bearing Measurements from Moving Aircraft. Naval Postgraduate School, Monterey, CA, April 1969.

Deutsch, R., Estimation Theory. Prentice-Hall, New Jersey, 1965.
Standard book on location estimation, confidence ellipses, and mathematical estimation arising especially in radar problems.

Diaconis, Persi and Efron, Bradley. Computer-Intensive Methods in Statistics. Scientific American. May 1983.

Efron, Bradley. The Jackknife, the Bootstrap and Other Resampling Plans. Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania 19103. 1982.

Felix, Robin. High Frequency Direction Finding: Errors, Algorithms, and Outboard. Naval Postgradute School, Monterey, CA, October 1982.

Fletcher, R. and Powell, M. A Rapidly Convergent Descent Method for Minimization. Computer Journal, Vol. 6, 1963. pp. 163-168.

Foy, Wade H. Position-Location Solutions by Taylor-Series Estimation. IEEE Transactions on Aerospace and Electronic Systems Vol. AES-12, No.2. March 1976.

Gething, P. J. D. Correlation Effects on Direction-Finding Probability Regions. Proc. IEE, Vol. 114, No. 2. February 1967.

Gill, P.E. et al, Practical Optimization. Stanford University, Department of Operations Research, 1980.
A good practical guide to numerical optimization methods with extensive references.

Hodson, III, William T. FALCONFIX: A Multi-Model Approach to Fix Computation. Department of Mathematical Sciences, United States Air Force Academy, CO, June 1979.

Houston, R. S. Model Error and the Direction-Finder Problem. IEEE Transactions on Aerospace and Electronic Systems Vol. AES-16, No. 5. September 1980.

Jacoby, S. L. S., Kowalik, J. S., and Pizzo, J. T. Iterative Methods For Nonlinear Optimization Problems. Prentice-Hall, Inc., Englewood Cliffs, New Jersey. 1972.

Jannusch, Craig Michael <u>Statistical Analysis of Three High Frequency Direction Finding Algorithms with Bearing Selection Based on Ionospheric Models</u>. Naval Postgraduate School, Monterey, CA, September 1981. AD-061906.

Jenkins, H. H. and Moss, R. W. <u>An Error Reduction Technique for Loop Direction</u>. IEEE Transactions on Aerospace and Electronic Systems. November 1969.

Jennings, A., <u>Matrix Computation for Engineers and Scientists</u>. John Wiley and Sons, New York, 1977. A solution-method oriented reference book which is comprehensive in scope. It contains brief program listings in ALGOL and FORTRAN.

Jennrich, R. I. <u>Asymptotic Properties of Non-Linear Least Squares Estimators</u>. Annals of Math. Stat., Vol. 40. p. 633-643. 1969.

Kolata, Gina. <u>The Art of Learning from Experience</u>. Science, Vol. 225, p. 156-158. July 1984.

Mahapatra, Pravas R. <u>Emitter Location Independent of Systematic Errors in Direction Finders</u>. IEEE Transactions on Aerospace and Electronic Systems. Vol.AES-16, No.6. November 1980.

Mardia, K. V., Kent, J. T., and Bibby, J. M. <u>Multivariate Analysis</u>. Academic Press, Inc. London, New York, Toronto, Sydney, San Francisco. 1979.

Miller, Rupert G. <u>The Jackknife--a Review</u>. Biometrika (1974), 61,1, P.1. Printed in Great Britain.

Morrison, D. F. <u>Multivariate Statistical Methods</u>. Second Edition. McGraw-Hill, New York. 1976.

Mosteller, Frederick and Tukey, John W. Data Analysis and Regression, A Second Course in Statistics. Addison-Wesley Series in Behavioral Science: Quantitative Methods. 1977.

Rainer, Richard and Burwasser, Alex J. An Approach to HF Tactical Radio Direction Finding and Signal Monitoring. Journal of Electronic Defense. October, 1983.

Ross, W. The Estimation of the Probable Accuracy of High Frequency Radio Direction-Finding Bearings. J. IEEE, Vol. 94, Part III A. 1947.

Ross, William. Wild Bearings in High-Frequency Direction Finding. Proc. IEE, Vol. 122, No.4. April 1975.

Sherrill, M. M. Bearing Ambiguity and Resolution in Interference Direction Finders. IEEE Transactions on Aerospace and Electronic Systems, Vol. AES-5, No. 6. November 1969.

Stansfield, R. G. Statistical Theory of D. F. Fixing. J. IEEE, Vol. 94, Part IIIA. 1947.

Stiffler, Donald R. Analysis of Six Algorithms for Bearings only Ranging in an Air-to-Air Environment. Air Force Institute of Technology (AFIT-EN), Wright-Patterson AFB, Ohio, December 1982.

VanBrunt, Leroy B. Applied ECM Volume 2. E. W. Engineering, Inc. Dunn Loring, VA. 1982.

Wangsness, Dennis L. A New Method of Position Estimation Using Bearing Measurements. IEEE Transactions on Aerospace and Electronic Systems. November 1973.

Wegner, L. H. On the Accuracy Analysis of Airborne Techniques for Passively Locating Electromagnetic Emitters. The Rand Corporation, June 1971.

Wegner, L. H. Estimation in a Model That Arises From Linerization in Nonlinear Least Squares Analysis. The Rand Corporation. April 1971. AD-725021.

A.5 MILITARY SYSTEMS AVAILABLE DOCUMENTATION

Secret Document (JPL Log AA-001137) Intelligence-Electronic Warfare System Compendium (U), 30 September 1982, DOA-USAICS is a reference for SIGINT systems.

Secret Document (JPL Log AA-000254) ASAS SEWS/TCAC(D) ELINT CORRELATION (FINAL) (U), dated 4 May 1981 is a reference for ELINT systems.

Confidential document (JPL Log AA-000493) GUARDRAIL/QUICKLOOK Operation (U), dated 6 June 1978, TC30-18.

TRAILBLAZER (AN/TSQ-114) TM 32-5811-022-10-1, Technical Manual, Operators Manual. Listings - 4 volumes assembly language.

GUARDRAIL V (AN/TSQ-105,AN/USD-9) ESL-TM 928, Software Technical Description, Volumes 1-16. Listings - 1 volume (FORTRAN).

QUICKLOOK II (AN/ALQ-133) Draft Manual - OPS VAN SOFTWARE. Revision 3.20 (Spring, 1982) Changes Revision 3.21 (June 1982) Revision 3.22 (September 1981).

QUICKLOOK II Operator Course, Student Handout, Description of QUICKLOOK II System, File No. F452/H01/AN/USM-393 Operating Programs, File No. F452/H02.

QUICKLOOK II Operator Course, Description of QUICKLOOK II System, File No. F452/H01/AN/USM-393 Operating Programs, File F452/H02.

Operator's Manual Receiving Set, Countermeasures, AN/ALQ-144. Simulator Set, AN/USM-393. Test Set, Flight Line, AN/ALM-154.

Operator's Manual Receiving Set, Countermeasures, AN/ALQ-133, Simulator Set, AN/USM-393 Test Set, Flight Line, AN/ALM-154.

SAL Language Assembler Software Specification for the U420 Monitor-controller

Main-Line Applications Program, AN/USM-393. Operator Course, F233-F8 (32 sets). Listings - 1 volume (comments only for assembly language code)

APPENDIX B

FIX ESTIMATION ERROR MODEL

Page intentionally left blank

APPENDIX B

CONTENTS

Page intentionally left blank

APPENDIX B
ERROR BUDGET

1.0 PURPOSE

The purpose of this appendix is to identify all of the various error components, in the most general case, when determining lines of bearings. These lines of bearing are used in subsequent fix estimations for emitters.

2.0 SCOPE

The essential assumptions of this document are: the emitter is not moving at the time the line of bearing is measured; the sensor may be in any position, from earthbound to a moving satellite.

The type of errors considered may be classified into several categories:

(1) Sensor platform position and orientation errors.
(2) Sensor attitude.
(3) Antenna errors.
(4) Instrumentation errors.
(5) Time.

The sensor platform postion and orientation errors may be referred to as "positional errors."

Errors due to propagation effects, site selection, varying aperture versus aspect effects, and operator errors are not considered in this document. Also, errors due to the choice of algorithms or numerical computations are not considered.

3.0 POSITIONAL ERRORS OF A PLATFORM

These error sources may be broadly classified into errors in the frame of reference and errors in position measurement. The former include errors which will be present regardless of a sensor platform's actual location or measurement thereof. These are largely the result of error or uncertainty in establishing the frame of reference for exchange of position information. Position measurement errors are those due to error or uncertainty in the methods and equipment used to determine platform location within the selected frame of reference.

The geocentric coordinates and references are:

Latitude	ϕ	Phi
Longitude	λ	Lambda
Altitude	h	

Orientaton of meridian plane (Direction of North)

These coordinates are best described by the diagram in Figure 1. The geographic latitude is measured positive from the equator toward the North Pole in degrees. The geographic longitude is measured positive from the prime meridian at Greenwich toward the East in degrees. The altitude is measured from the mean sea level (the geoid) in meters and is positive in a direction away from the center of the earth. The physical sources of errors in these parameters will depend largely on the source of the data used to determine them.

3.1 FRAME OF REFERENCE ERRORS

Establishment of a frame of reference for exchange of position information on the earth involves seven major processes, all of which may introduce some error of uncertainty into any position reference within the selected framework.

3.1.1 Geodetic Errors

Two of the seven processes are the province of geodesy, and involve measuring and representing the shape of the earth.

One process involves measurement of the actual shape of the earth independent of local variations in topography. This is typically most closely represented by mean sea level, i.e., sea level independent of variations due to lunar tides, and local gravitational anomalies. The resulting geometric figure, termed a geoid, becomes the basis for subsequent representations of the earth's surface. This figure is subject to error and uncertainty due to the measurement process and to changes in the actual shape of the earth over time.

The second process is the selection of a geometrical figure close to the geoid in shape, but simpler from the standpoint of mathematical and geometric manipulation, to be used as the basis of the mapping process in a given part of the world. The figure is generally based on a very nearly spherical ellipsoid which, because of its nearness to spherical shape, is often called a spheroid. Different spheroids are in use for different parts of the world both for historical reasons and because slightly differing ellipsoids best approximate the geoid over different parts of the earth. Different spheroids are typically defined by giving the radius at the equator and the flattening. The latter is defined as the difference between the radius at the equator and that at the pole divided by the radius at the equator. Selection of a spheroid introduces error as the selected figure is only an approximation to the geoid, and may vary from the geoid irregularly over the portion of the earth being mapped.

3.1.2 Geomagnetic Errors

Airborne platforms depend on a magnetic flux goniometer during initialization of the inertial platform. Field soldiers and mobile units often have to depend on magnetic compasses for determining bearings. Although this is one of the oldest means of taking bearings, it can be very inaccurate. The earth's magnetic field tends to align with the nearest magnetic pole. However, the magnetic poles are about three kilometers from the geographic poles. Furthermore, the two poles, North and South, are not even symmetrically placed. And to complicate this, there are local variations over all the earth's surface. This angle that the compass makes with the grid lines of a military map is called the "declination" of the compass. The magnetic lines of force are not parallel to the earth's surface, except along the indefinite circle called the magnetic equator. The angle the magnetic field makes with a horizontal plane is called the dip angle or the magnetic inclination.

The declination at any one location does not remain the same year after year and changes somewhat over long periods of time. Besides these so-called secular changes, there are variations within the year and also small changes of angle throughout the day. Large erratic variations occur during "magnetic storms." These storms are often concurrent with the appearance of sun-spots. Variations from storms are infrequent enough and the other variations are sufficiently slow that it is practical to publish maps of countries and other large areas showing the magnetic declination. On these maps, points of equal magnetic declination are connected by lines. Each wiggly line is labeled with the amount and direction of the magnetic declination. These lines are called isogonic lines. The isogonic line of zero magnetic declination is indicated by a heavy line, and is called the agonic line. Maps of smaller area indicate the magnetic declination in their legend by an arrow pointing to the magnetic north and labeled with the value of the magnetic declination in degrees.

The National Space Technology Laboratory at the Naval Office in Bay Saint Louis, Mississippi has a world mathematical model of the earth's magnetic field. The model consists of an order 12 spherical harmonic series with time varying coefficients to take care of secular changes. The model is considered good for $\pm$ 5 years. However, the model is incapable of describing anomalies smaller than about 1,100 km, and has an inaccuracy of about 3,500 km. The model is updated every five years from new satellite and aircraft survey data. Local anomalies will normally deviate a few degrees of arc from the earth's main field direction, but can deviate by tens of degrees in areas where the mineral magnetite is abundant and in polar regions. For accurate orientation using the earth's magnetic field, there is no good substitute for a local survey.

3.1.3 Cartographic Errors

The remaining four processes introducing frame of reference errors fall into the province of cartography, i.e., the recording, measurement, and representation of geographic, topographic, and cultural features on the surface of the earth.

The first of these processes involves selection of one or more coordinate systems to be used to specify locations on the selected representation of a portion of the earth. In virtually all world reference systems, at least one of the coordinate frames used will apply to the selected spheroid, and the reference system used is in fact almost always the familiar geographic (latitude-longitude) coordinate system. Errors arise in this process due to errors in the measurements associated with selection of reference or registration points as bases of the coordinate system as well as in the measurement and computation involved in extending the coordinate frame from the base points through the area to be mapped.

The second process involves, in those cases where the final representation will be planar, a projection of all or a portion of the selected spheroid onto a plane according to some well defined set of mathematical and geodetic conventions. This step will often be followed by another iteration of the first step to select a reference system suitable for measurement and computation in the Euclidean plane. Errors arise in this process due to the distortion involved in the projection from the spheroid to the plane as well as in any subsequent registration and extension of the associated planar coordinate system.

The third process consists of the recording and measurement of surface features within the selected coordinate system(s). The errors inherent to this process include those associated with measurement of the features themselves, their relative locations, and their locations with respect to the coordinate systems selected.

The fourth process consists of the rendering of recorded features and associated coordinate systems into one or more forms that can be interpreted by people with a modicum of training and experience. Errors arise in this process due to distortions and simplifications imposed by the scale and resolution available in the final product, which in turn are governed in part by the current technology and in part by the limitations of the human perceptual system. A highway 10 meters in width, for example, if represented to scale on a 1:250,000 map, would be 0.04 millimeters wide and all but invisible to the naked eye.

3.2 POSITION MEASUREMENT ERRORS

3.2.1 Inertial Navigation

The four coordinates of position can be maintained by a suitably designed inertial platform. There will be essentially four type of errors with such systems:

(1) Errors in measurement and setting of initial position.
(2) Errors in platform measurement of inertial change.
(3) Errors in precision of computation of position from inertial change
(4) Cumulative errors in position, i.e., drift.

The basic component of most modern inertial navigation systems is the gyroscope. In addition to the familiar function of referencing direction (gyro compass), gyroscopes may be designed to measure rotations, to seek the local vertical, and to act as accelerometers.

3.2.2 Referenced Navigation

Referenced navigation systems are those that depend on beacons, or repeaters of known position or velocity. These may be classified by the geometry of the data processing:

(1) Hyperbolic (Decca, Loran, Omega, Satellite Aided Navigation.

(2) Circular (Sextant, Satellite Aided Navigation.

(3) Polar (TACAN).

The hyperbolic and the circular navigation systems are methods of triangulation. However, the hyperbolic method deals exclusively with the sides of the triangle, while the circular method deals with two sides and an angle. The polar method gives both a range and azimuth from the reference station.

Decca is a low frequency (70-130 kHz) hyperbolic system that triangulates by measuring the phase difference between signals from a master/slave pair of reference stations. The master/slave separation is 60 to 120 n.m. The useful range is about 240 n.m. over water. Loran A is medium frequency (2 MHz) hyperbolic system that triangulates by measuring the time difference between receipt of pulses from two stations. The range of Loran A is several hundred miles over water, but much reduced over land. Loran C is a low frequency (90-110 kHz) version of Loran A with considerably more range. OMEGA is a very low frequency hyperbolic system that triangulates by comparing the phase of signals from two beacons separated by a baseline of 5,000 to 6,000 miles. The coverage is world wide and may be used by submersibles.

Satellite-aided navigation has the most diverse possibility for use as a referenced system of navigation. The orbital elements and thus both the position and velocity of the satellite are accurately known. By combining such measurables as elevation angle, azimuth angle, ranges, difference in range, range sum, or doppler shift, fixes may be obtained that fit any of the listed categories in the first paragraph of this section. Methods that depend on measurement of the elevation angle of one or more satellites determine small circles on the earth's surface for fixes. Methods that determine distances lead to hyperbolic conic lattices for fixing.

TACAN is a UHF radio navigation system which provides both distance and bearing information of the aircraft relative to the selected ground beacon. The antenna system is the key to measuring the aximuth. The antenna system has a single, central element for transmission and reception. The parasitic elements are mounted on two concentric cylinders which rotate at fifteen revolutions per second. The inner cylinder consists of a single parasitic element which causes a single cardioid polar pattern rotating at 15 rps. The outer cylinder has nine parasitic elements that superimpose nine lobs on the cardioid pattern. This gives an amplitude modulated signal with two frequency modulations of 15 Hz and 135 Hz. The transponder further emits bearing reference pulses as the peak of each lobe points East. When the lobe which coincides with the peak of the cardioid points East, a special "North" reference pulse code is transmitted. The airborne equipment measures the phase relationship of the maximum signal amplitude relative to the North reference pulses in order to determine the bearing of the aircraft relative to the beacon. The accuracy of the azimuth is in the order of magnitude of two degrees. The distance measuring part of TACAN equipment is like radar except that the return signal comes from a beacon used to produce strong artificial echoes. The beacon will respond to numerous simultaneous interrogations. To make this possible, the pulse repetition rate of the airborne transmitter is cause to jitter in a random manner. The receiver is allowed to recognize only those pulses received that follow the same jitter pattern and ignore all other. The slant range is determined to roughly 0.25 nautical miles under most conditions.

3.2.3 Doppler Navigation

Airborne Doppler is a SHF (micro-wave) system of navigation using the terrain or water below as a reference. Depending on the particular doppler system used, some or all of the following data can be made available to the crew:

(1) Component velocities and distances run, along, across, and perpendicular to the aircraft axes.
(2) Ground speed.
(3) Drift angle.
(4) Angle of attack.
(5) Height above terrain.

If True Air Speed, Pitch, and Heading Angles are available from such sources as an inertial system, then the following secondary data may be obtained.

(1) Wind speed and direction.
(2) Climb angle.
(3) Track angle.

The Doppler systems may have various configurations of antenna beams directed at different angles toward the earth. A two beam system may be used to measure ground speed and drift. A three beam system is basically sufficient to extract all three velocity components, but a four beam symmetrical arrangement is often used.

4.0 ATTITUDE ERRORS

4.1 ATTITUDE COORDINATES

The three attitude coordinates are:

Roll Angle	α	alpha
Pitch Angle	β	beta
Yaw Angle	γ	gamma

Figure 2 serves to define each of these angles. These are the standard Euler angles as defined by a "right hand" rule. However, it should be noted that the sign of these angles vary considerably throughout published literature. See Korn and Korn, reference 2, section 14.10-6, for a discussion of this coordinate system and the diverse choice of signs. In some aireborne systems these positional coordinates are limited by preset stops which may introduce non-linear errors.

4.2 CORRELATION BETWEEN ATTITUDE AND POSITIONAL COORDINATES

With a cursory examination of these six coordinates, it is apparent that errors in three of them will produce the larger errors. An error in yaw angle alone will produce a divergence of the azimuth angle of bearing. This azimuth error will always be quite close in magnitude, but opposite in sign, to the Yaw error. Errors in longitude and latitude will produce an error in the position of the line of bearing as a function of the azimuth angle, but this does not effect the azimuth angle. If the azimuth angle is in the vicinity of zero or 180 degrees, an error of longitude will be reflected directly, and of nearly the same value, in the longitude of the fix estimations. At azimuth angles of 90 and 270 degrees, the line of bearing and consequently the fixes are unaffected by errors in longitude. The effects of errors in latitude are analogous in their effect but displaced by 90 degrees.

It is not so obvious that an error in the three remaining coordinates (altitude, roll, and pitch) should have any effect on the line of bearing. Indeed an error in altitude alone should only change the slant range and have no effect on the line of bearing. However, when coupled wih errors in roll and pitch, there is a definite mathematical relation or coupling. The significance of an error in altitude remains to be evaluated. Errors in roll and pitch (which have less effect on the error of the fix estimate than yaw, longitude, and latitude) directly cause errors in azimuth angle on the line of bearing.

5.0 ANTENNA ERRORS

Orientation with respect to the platform.
Difference between the mechanical axis and RF axis.
Beam width.

These first two antenna errors are directly related to the platform attitude coordinate errors. In fact the orientation of the antenna with respect to the platform and the difference between the mechanical axis and the RF axis are best described by Euler angles. If the axis defining these coordinates are chosen originally in coincidence, first order approximations will serve to considerably simplify the maze of trigonometric functions relating these angles. These three Euler angles can be identified as pitch, roll, and yaw. For small errors in these angles, the errors may be simply added to the corresponding platform angles. It should be noted at this point that the RF boresight error is a function of the radio frequency.

Beam width is always a function of the antenna geometry and frequency. A phased antenna system's beam width will vary considerably with change in aspect angle.

6.0 INSTRUMENTATION ERRORS

Bias (Systematic errors).
Noise (Random errors).

Bias errors, for example, are systematic errors such as boresite errors, parallax errors in instrument readings, and bezel errors. Bias errors are usually minimized by calibration procedures.

Noise errors are due to random phenomenon such as receiver noise. This noise normally produces random errors in bearings by increasing the region of uncertainty when determining the minima of a signal or the change in sign from the phase of a signal. There are many techniques of minimizing the effects of noise, depending on the source and nature of the noise (see Reference 5). In high frequency receivers, the receiver's "front end" is a high source of thermal noise, so the high gain required is usually obtained after heterodyning to a lower frequency or after further detection at the "rear end." Commonly, the band pass of filtering is reduced to the minimum that will not deteriorate the information content. The effect of impulse noise, such as noise emanating from electrical ignitions, can be minimized by amplitude clipping just above the signal level.

7.0 ERROR TABULATION

The sensor positional error is equally important in fixing, mobile, or airborne sensors. The attitude errors are most important in airborne sensors. The sensor geometric error refers to such errors as the difference between the geometric and RF axis of a direction-finding antenna, or even an optical tracker. Range is included with the geometric sensor errors for convenience only.

The specifications and tolerances will always include the units. The exact meaning of the specification and tolerance columns will depend on the instrument involved.

TABULATION OF ERROR BUDGET

SYSTEM__________
COMPONENT__________
A/N NUMBER __________
MODEL__________

CLASSIFICATION	SPECS	TOLER
1. Sensor Positional Errors		
a. Longitude		
b. Latitude		
c. Altitude		
d. Position (linear distance)		
2. Sensor Attitude Errors		
a. Reference meridian (North)		
b. Roll		
c. Pitch		
d. Yaw		
e. Rates (TBD)		
3. Sensor Geometric Errors		
a. Azimuth		
b. Elevation		
c. Range		
4. Instrument errors		
a. Bias (systematic or secular errors)		
b. Noise random errors		

References:

NOTES:

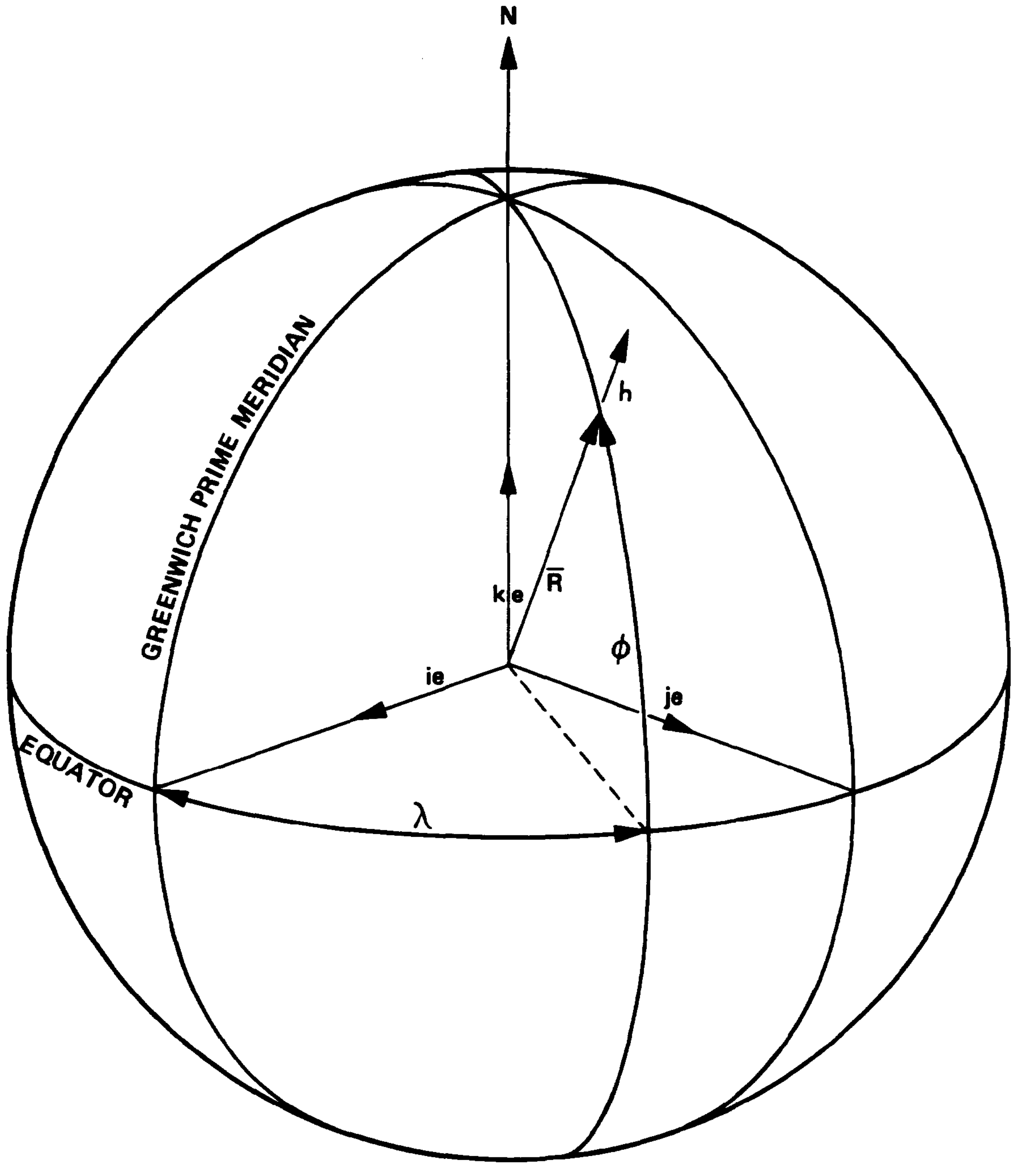

Figure 1. Geocentric Positional Coordinates

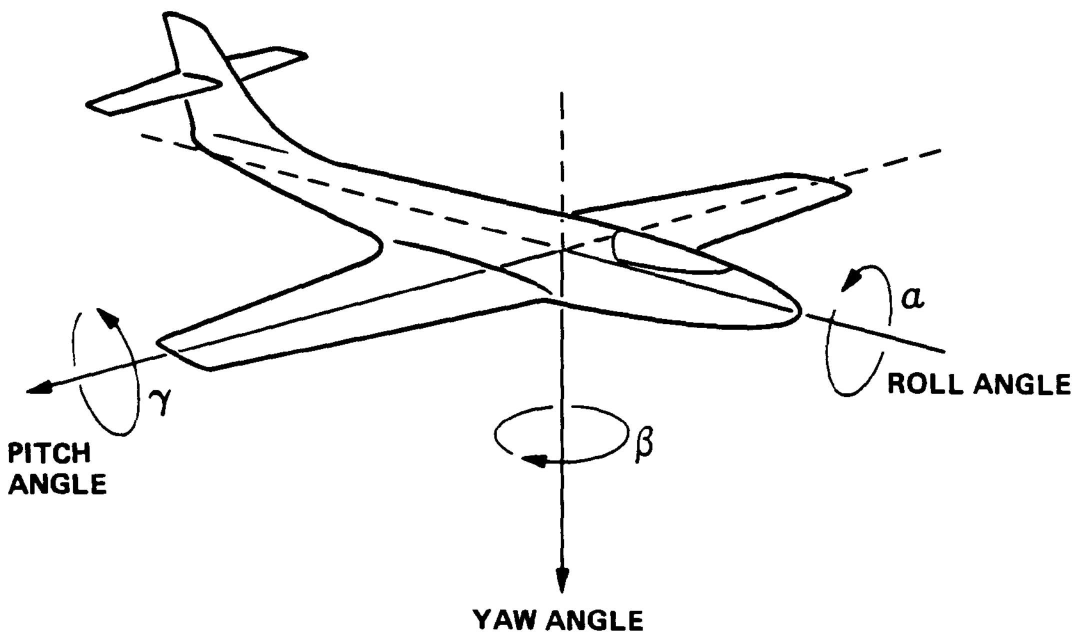

Figure 2. Attitude of the Platform

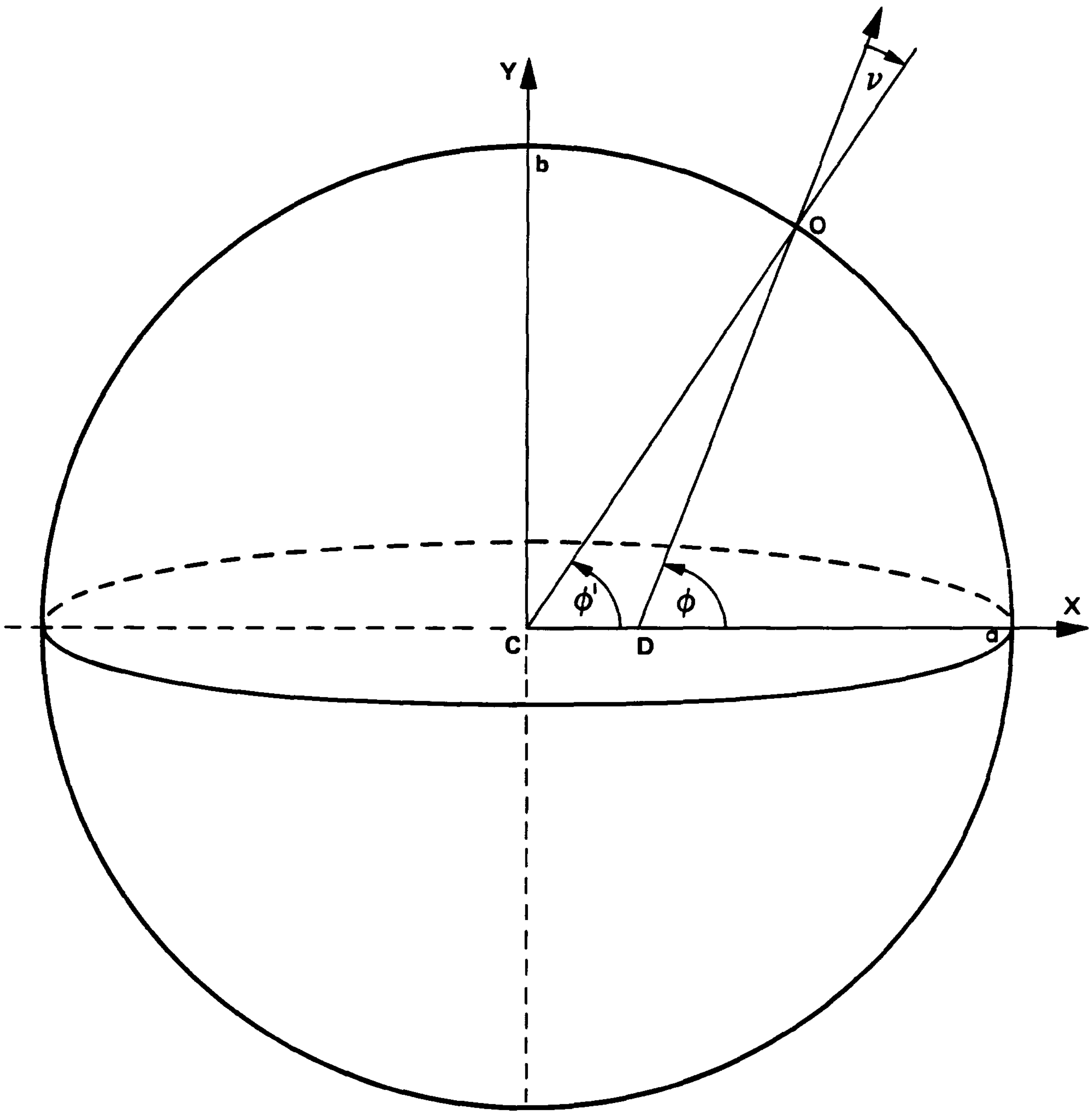

Figure 3. The Geoid and Latitude

REFERENCES

1. USAICS, Software Analysis and Management System, "Analysis of Geographic Transformation Algorithms," Jet Propulsion Laboratory, Pasadena, CA, 09 July 1985.

2. Korn and Korn, "Mathematical Handbook for Scientists and Engineers," Second Edition, McGraw-Hill Book Co., 1968.

3. Robinson, A., Sale, R., Morrison, J., "Elements of Cartography," Fourth Edition, John Wiley & Sons, Inc., 1978.

4. Smart, W. M., "Text-Book on Spherical Astronomy," Cambridge University Press, 1962.

5. Hutchinson, C. L., "The ARRL Handbook for the Radio Amateur," American Radio Relay League, Newington, CT, 1985.

6. Jerald W. Caruthers, to Col Leonard G. Mowak, "Summary of Errors Related to Magnetic Declination Data," I.O.M. from Department of the Navy, Naval Oceanographic Office, Bay Saint Louis, NSTL, Mississippi, 39522, 22 August 1984.

7. Barker, F. S., Barraclough, D. R., and Malin, S. R. C., "World Magnetic Charts for 1980-Spherical Harmonic Models of the Geomagnetic Field and Its Secular Variation," Geophys. J. R. Soc. (1981) 65, 525-533.

APPENDIX C

TRAILBLAZER ERROR BUDGET

Page intentionally left blank

TABULATION OF ERROR BUDGET

SYSTEM ______
COMPONENT ______
A/N NUMBER ______
MODEL ______

CLASSIFICATION	SPECS	TOLER
1. Sensor Positional Errors		
a. Longitude		
b. Latitude		
c. Altitude		
d. Position (linear distance)		
2. Sensor Attitude Errors		
a. Reference meridian (North)		
b. Roll		
c. Pitch		
d. Yaw		
e. Rates (TBD)		
3. Sensor Geometric Errors		
a. Azimuth		
b. Elevation		
c. Range		
4. Instrument errors		
a. Bias (systematic or secular errors)		
b. Noise random errors		

References:

NOTES:

APPENDIX D

ALGORITHMS IN STANDARD FORM

Page intentionally left blank

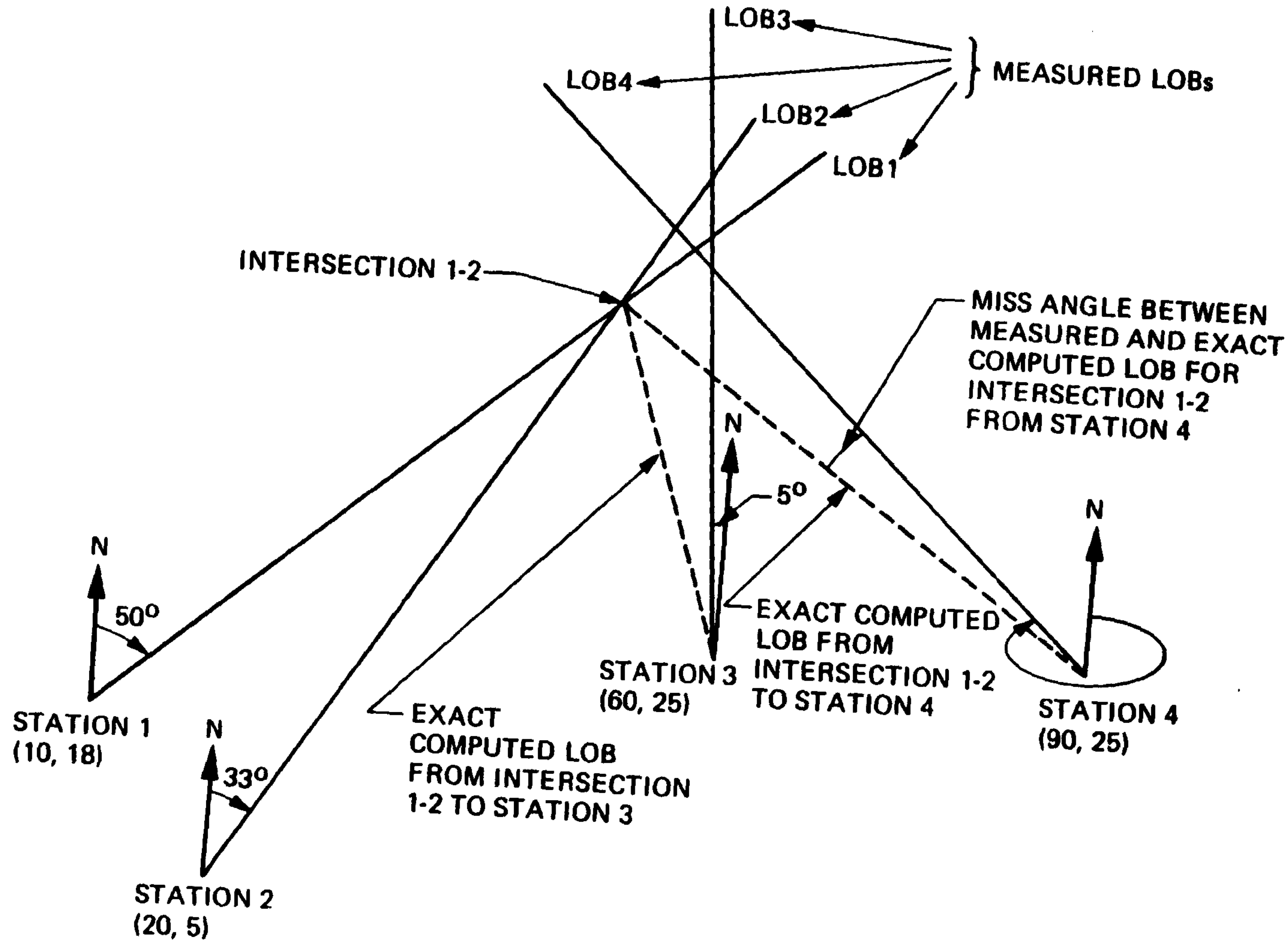

Figure D-1. Illustration for Test Case for TRAILBLAZER Algorithm in Pascal

```
PROGRAM TRAILBLAZER(INPUT,OUTPUT,TRAILIN,TRAILOUT);
(**************************************************************************
(*
   This program/algorithm has been tailored specifically for operation
   with the LOB data produced by a ground-based DF NET, consisting of a
   small fixed number of DF stations, whose locations remain invariant
   during the data collection process. The distinctive feature of this
   algorithm is its capacity to resolve reliably, multiple targets
   while at the same time avoiding GHOSTS, i.e. false targets arising
   from coincidental intersections of unrelated LOBS.

   Original program written in "ROLM ASSEMBLEY LANGUAGE".
   Translated into PASCAL by Nick Covella, JUNE 1984.


*)

CONST

    Shapefactor = 3;
    Pie         = 3.14159;
    Radian      = 180;
    Max         = 101;
    Min         = 0.6;
    Sigma       = 8;

TYPE

   Stations     = RECORD

                    Xcoord      : REAL; (* X-Location of any station. *)
                    Ycoord      : REAL; (* Y-Location of any station. *)
                    Theta       : REAL; (* Angle inputted from the
                                           user. *)
                    Alpha       : REAL; (* Calculated angle from input
                                           to TRUE NORTH *)
                    MissAngle   : REAL; (* Difference between the Alpha angle
                                           and the recalculated angle after
                                           an intersection has been found. *)
                    Semimajor   : REAL; (* in kilometers *)
                    Semiminor   : REAL; (* in kilometers *)
                    Distance    : REAL; (* approx. distance from the staions
                                           to the object being "fixed". *)
                    Reflob      : BOOLEAN; (* indicates that an LOB has
                                              been attempted by the station. *)

                    Flag        : BOOLEAN; (* indicates that an intersection
                                              for this station has been
                                              calculated. *)
                    Orgin       : INTEGER; (* indicator for station
                                              manipulation *)

                  END;

    Data        = RECORD

                    Xintercept    : REAL;
                    Yintercept    : REAL;
                    Support         REAL;
                    SupportFlag   : ARRAY[1..5] OF BOOLEAN;
                    SupCountFlag  : INTEGER;
```

```
                    Flag          : BOOLEAN;

                  END;

    Intlotype    = ARRAY[1..5] OF INTEGER; (* Data structure that will
                                              keep track of the number
                                              of LOB's for the station
                                              being accessed. *)
    Lobtype      = ARRAY[1..5] OF Stations;

    TVT          = ARRAY[1..5,1..5] OF Data; (* Table for the intersections
                                                of the lobs in each bin. *)

VAR

  MissedAngle     : REAL;
  AStation        : INTEGER;
  BinNumber       : INTEGER;
  Index           : INTEGER;
  Intlosubscript  : INTEGER;
  IntPass         : INTEGER;
  Nmax            : INTEGER;
  Nother          : INTEGER;
  Nsame           : INTEGER;
  Col             : INTEGER;
  StaNumb         : INTEGER;
  Row             : INTEGER;
  TrailOut        : TEXT;
  TrailIn         : TEXT;
  Intlo           : Intlotype;
  Lob             : Lobtype;
  ID              : Stations;
  Table           : TVT;

(*************************************************************************
PROCEDURE CLEAR(VAR Lobrec      : Lobtype;
                VAR Intloarray  : Intlotype;
                VAR Tabletype   : TVT);
(*************************************************************************
(* This subroutine clears all of the data structures and prepares them
   for either the first pass or any other subsequent passes.
*)

VAR

  I               : INTEGER;
  BinNumber       : INTEGER;
  AStation        : INTEGER;
  Intlosubscript  : INTEGER;

BEGIN

  WRITELN('          Entering CLEAR');
  I := 0;

  FOR BinNumber := 1 TO 5 DO   (* initialize the variables of the bins of
                                  each station. *)

    BEGIN
```

```
      Lobrec[BinNumber].Xcoord     := 00.000;
      Lobrec[BinNumber].Ycoord     := 00.000;
      Lobrec[BinNumber].Theta      := 00.000;
      Lobrec[BinNumber].Alpha      := 00.000;
      Lobrec[BinNumber].MissAngle  := 00.000;
      Lobrec[BinNumber].Semimajor  := 00.000;
      Lobrec[BinNumber].Semiminor  := 00.000;
      Lobrec[BinNumber].Distance   := 00.000;
      Lobrec[BinNumber].Reflob     := FALSE;
      Lobrec[BinNumber].Flag       := FALSE;
      Lobrec[BinNumber].Orgin      := 0;

    END;

  FOR BinNumber := 1 TO 5 DO

    BEGIN

      I := I + 1;
      FOR AStation := 1 TO 5 DO  (* initialize the truth-table of the bins *)

        BEGIN

          Tabletype[BinNumber,AStation].Xintercept      := 00.000;
          Tabletype[BinNumber,AStation].Yintercept      := 00.000;
          Tabletype[BinNumber,AStation].Support         := 00.000;
          Tabletype[BinNumber,AStation].SupCountFlag    := 0;
          Tabletype[BinNumber,AStation].SupportFlag[I]  := FALSE;
          Tabletype[BinNumber,AStation].Flag            := FALSE;

        END;

    END;

  FOR Intlosubscript := 1 TO 5 DO (* intersection LOB array. *)
    Intloarray[Intlosubscript] := 0;    (* initialized to -1 in actual program *

  WRITELN('            Leaving  CLEAR');

END; (* PROCEDURE "CLEAR" *)

(**********************************************************************
PROCEDURE INELLIPSE(VAR Xefix   : REAL;
                    VAR Yefix   : REAL;
                    VAR Answer  : BOOLEAN;
                        Table   : TVT);
(**********************************************************************)
(*
   This procedure will insure that a fix estimate doesn't duplictate
   the last or any previous fix. INELLIPSE returns true in the boolean
   variable Answer if the fix estimate is in the ellipse determined by
   the previous fixes.
*)

VAR

  Value1, Value1Sqr, Value2, Value2Sqr : REAL;
  Tempadd, Tempcos, Tempsin             : REAL;
  Sum                                   : REAL;
  Xcoord, Ycoord                        : REAL;
```

```
  Semiminoraxis                              : REAL;
  Semimajoraxis                              : REAL;

BEGIN

  WRITELN('            Entering INELLIPSE');

  (* get x and y coordinates of the center of the ellipse *)
  (* get the cosine of the ellipse *)
  Tempcos := COS((Yefix * Pie) / Radian);
  Tempsin := SIN((Xefix * Pie) / Radian);
  Tempadd := Tempcos + Tempsin;
  Semimajoraxis := Tempcos + Tempsin;     (* JUST FOR ARGUEMENT *)
  Value1 := Tempadd / Semimajoraxis;
  Value1Sqr := Value1 * Value1;
  Tempcos := COS((Xefix * Pie) / Radian);
  Tempsin := SIN((Yefix * Pie) / Radian);
  Tempadd := Tempcos - Tempsin;
  Semiminoraxis := Tempcos + Tempsin;     (* JUST FOR ARGUEMENT *)
  Value2 := Tempadd / Semiminoraxis;
  Value2Sqr := Value2 * Value2;
  Sum := Value1Sqr + Value2Sqr;
  IF Sum > 2 THEN
    Answer := TRUE (* indicating that the estimate was probably different
                      from any other estimate *)
  ELSE
    Answer := FALSE; (* indicating that the estimate already exists. *)

  WRITELN('            Leaving  INELLIPSE');

END; (* PROCEDURE "INELLIPSE" *)

(*****************************************************************************
PROCEDURE MSCAN(VAR Lobrec : Lobtype);
(*****************************************************************************
(*
   This procedure searches all LOBS in the local database. if the
   LOBS have been marked indicating that they have been used for a
   particular fix then they will be unmarked by this procedure.
*)

VAR

  BinNumber : INTEGER;

BEGIN

  WRITELN('            Entering MSCAN');

  FOR BinNumber := 1 TO 5 DO (* check each BINSET to see which have been
                                marked. Unmark those that have been marked. *)

     IF Lobrec[BinNumber].Flag = TRUE THEN
        Lobrec[BinNumber].Flag := FALSE;

  WRITELN('            Leaving  MSCAN');

END; (* PROCEDURE "MSCAN" *)
```



```
(*****************************************************************************)
```

```
                    Reflob      : INTEGER;
                VAR Lobrec      : Lobtype);
(*****************************************************************************)
(*
   This procedure accumlates counts of marked LOBS for each station in
   the counter NOTHER and NSAME. "SCNDB" is called by "FPOT" to per-
   form missangle variance on a station basis.
*)


VAR

  SumVariance          : REAL;
  SumPotential         : REAL;
  Temp1, Temp2, Temp3  : REAL;
  Exp                  : REAL;
  Potential            : REAL;
  BinNumber            : INTEGER;
  Sigma, Shapefactor   : INTEGER; (* constants given in the progeam *)
  X, Y                 : INTEGER; (* Substitute variables for the parameters
                                     passed to this subroutine. *)
  Scancntr             : INTEGER; (* counter of marked LOBS *)

BEGIN

  WRITELN('          Entering SCNDB');

  X := Compcnt;
  Y := Reflob;
  SumVariance := 0;
  SumPotential := 0;
  Nother := 0;
  Nsame := 0;
  IF X = 1 THEN
    Scancntr := Nother;
  For BinNumber := 1 TO 5 DO
   IF Lobrec[BinNumber].Reflob = FALSE THEN

     BEGIN

       Lobrec[BinNumber].Flag := FALSE;
       Scancntr := Nsame;
       Nother := 1;

     END;

  Lobrec[BinNumber].Reflob := TRUE;

  WHILE Lobrec[BinNumber].Flag <> TRUE DO

    BEGIN

      IF X = 1 THEN
        Scancntr := Scancntr + 1;
      IF ((Scancntr = 0) OR (X = 0)) THEN
        MissedAngle := Scancntr - X   (* LOB - reflob *);
      IF MissedAngle <= 0 THEN
        MissedAngle := -(MissedAngle);
      IF MissedAngle > Pie THEN

        BEGIN
```

```
        MissedAngle := (2 * Pie) - MissedAngle;
        Sigma := 8;
        Shapefactor := 3;
        Temp1 := Sigma * Shapefactor;
        Temp2 := MissedAngle / Temp1;
        Exp   := ((SQR(Temp2)) / 2);
        Temp3 := (((Exp * Exp)/2) + ((Exp * Exp * Exp)/6) + Exp + 1);
        Potential := 1/Temp3;
        SumPotential := Sumpotential + Potential;
        SumVariance := SumVariance + ((SQR(MissedAngle) * Potential));

      END;

    END;

  IF Nother = 0 THEN
    Scancntr := Nother
  ELSE
    Scancntr := Nsame;

  WRITELN('          Leaving  SCANDB');

END; (* PROCEDURE "SCANDB" *)

(***************************************************************************
PROCEDURE XPREP(     StIdent  : Lobtype;
                     I        : INTEGER;
                     Intx     : REAL;
                     Inty     : REAL;
                 VAR Computed : BOOLEAN);
(***************************************************************************
(*
   This procedure computes the LOB angle of the Intersection and
   returns the coordinates of the proper location in the variables
   Intx and Inty.

*)

VAR

  DistanceX : REAL;
  DistanceY : REAL;
  TotalDist : REAL;

BEGIN

  WRITELN('          Entering XPREP');


  DistanceX := StIdent[I].Xcoord - Intx;
  DistanceY := StIdent[I].Ycoord - Inty;
  TotalDist := SQRT((SQR(DistanceX)) + (SQR(DistanceY)));

  IF (TotalDist < Min) THEN

    BEGIN

      WRITELN(TrailOut);
      WRITELN(TrailOut, 'The distance from Station #',StIdent[I].Orgin:1,' is to
```

```
      WRITELN(TrailOut);
      Intx := 00.000;
      Inty := 00.000;
      Computed := FALSE;

    END;

  IF (TotalDist > Max) THEN

    BEGIN

      WRITELN(TrailOut);
      WRITELN(TrailOut,'The distance from Station #',StIdent[I].Orgin:1,' is to
      WRITELN(TrailOut,'far to obtain a proper intersection.');
      WRITELN(TrailOut);
      Intx := 00.000;
      Inty := 00.000;
      Computed := FALSE;

    END;

  WRITELN('           Leaving  XPREP');

END; (* PROCEDURE "XPREP" *)

(*************************************************************************
PROCEDURE XCOMP(     I          : INTEGER;
                     J          : INTEGER;
                     Xdist      : REAL;
                     Ydist      : REAL;
                     StIdent    : Lobtype;
                 VAR Intx       : REAL;
                 VAR Inty       : REAL;
                 VAR Computed   : BOOLEAN);
(*************************************************************************
(* This procedure is called twice by "XSEC", once for each of the two
   stations involved in the computation of their LOB intersection. If
   the intersection is negative then no intersection is computed.

   NOTE: The parameters of I and J contain the staion number of the two
         stations whose intersection is to be computed.
*)

VAR

  Dx, Dy          : REAL; (* Temporary variable for D1 and D2. *)
  CosineI, SineI  : REAL;
  CosineJ, SineJ  : REAL;
  TempDist        : REAL;

BEGIN

  WRITELN('           Entering XCOMP');

  Dx := Xdist;
  Dy := Ydist;

  IF (StIdent[I].Alpha = StIdent[J].Alpha) THEN

    BEGIN
```

```
      Intx := 00.000;
      Inty := 00.000;
      Computed := FALSE;
      WRITELN(TrailOut);
      WRITE(TrailOut,'Station #',StIdent[I].Orgin:1,' and Station #');
      WRITE(TrailOut,StIdent[J].Orgin:1,' have the same initial LOB');
      WRITELN(TrailOut);
      WRITELN(TrailOut,'trajectory....No intersection possible.');
      WRITELN(TrailOut);

    END

  ELSE

    BEGIN

      CosineI := COS(((StIdent[I].Theta) * Pie) / Radian);
      CosineJ := COS(((StIdent[J].Theta) * Pie) / Radian);
      SineI := SIN(((StIdent[I].Theta) * Pie) / Radian);
      SineJ := SIN(((StIdent[J].Theta) * Pie) / Radian);
      Xdist := (( Dx * CosineJ ) - ( Dy * SineJ ));
      Xdist := Xdist/SIN(((StIdent[J].Theta - StIdent[I].Theta) * Pie) / Radiar
      Ydist := (( Dx * CosineI ) - ( Dy * SineI ));
      Ydist := Ydist/SIN(((StIdent[J].Theta - StIdent[I].Theta) * Pie) / Radiar
      Computed := True;

    END;

  IF (Computed = TRUE) THEN

    BEGIN

      Intx := StIdent[I].Xcoord + (Xdist * SIN((((StIdent[I].Theta)
              * Pie) / Radian)));
      Inty := StIdent[I].Ycoord + (Xdist * COS((((StIdent[I].Theta)
              * Pie) / Radian)));
      Computed := TRUE;

    END;

  IF ((ABS((StIdent[I].Alpha) - (StIdent[J].Alpha)) <= 0.6)  AND
      (Computed = TRUE)) THEN

    BEGIN

      WRITELN(TrailOut);
      WRITE(TrailOut,'The difference between the angles of Station #');
      WRITE(TrailOut,StIdent[I].Orgin:1,' and Station #',StIdent[J].Orgin:1);
      WRITELN(TrailOut);
      WRITELN(TrailOut,'is too small to obtain a proper intersection.');
      WRITELN(TrailOut,'Station #',StIdent[I].Orgin:1,' and Station #',StIdent[
      WRITELN(TrailOut,'are not participating in the test data');
      WRITELN(TrailOut);
      Intx := 00.000;
      Inty := 00.000;
      Computed := FALSE;

    END;
```

```
    BEGIN

      XPREP(StIdent, StIdent[I].Orgin, Intx, Inty, Computed);
      XPREP(StIdent, StIdent[J].Orgin, Intx, Inty, Computed);

    END;

  WRITELN('           Leaving  XCOMP');

END; (* PROCEDURE XCOMP *)


(***********************************************************************)
PROCEDURE ILOBS(VAR AllStat : Lobtype; (* LOB's back to remaining stations
                                          which have inputted data into
                                          the database. *)
                    StIdent : Lobtype;
                    Intx    : REAL; (* possibly a VAR *)
                    Inty    : REAL; (* possibly a VAR *)
                    StaNumb : INTEGER;
                VAR Table   : TVT);
(***********************************************************************)
(* This procedure completes the intersection file called "INTLO".
   Reported LOBS are flagged and stored in "INTLO" as indexed by the
   repective station numbers.
*)

VAR

  AngleTheta   : REAL; (* Angle formed by the fixed point back to eah
                          station. *)
  DistX        : REAL; (* distance between the fixed x-coord and the
                          x-coord of the stations' position. *)
  DistY        : REAL; (* distance between the fixed y-coord and the
                          y-coord of the stations' position. *)
  Temp4        : REAL;
  Temp5        : REAL;
  I, J, K, N   : INTEGER;
  A, B         : INTEGER;
  SupportCount : INTEGER;
  NewData      : Lobtype;

BEGIN

  WRITELN('           Entering ILOBS');

  N := 1;
  DistX := 00.000;
  DistY := 00.000;
  K := StaNumb;

  FOR J := 1 TO StaNumb DO

    IF ((StIdent[J].Reflob = TRUE) AND (StIdent[J].Flag = TRUE)) THEN

      BEGIN

        NewData[N].Orgin := StIdent[J].Orgin;
        N := N + 1;
```

```
      END

    ELSE

      BEGIN

        NewData[K].Orgin := StIdent[J].Orgin;
        K := K - 1;

      END;

  N := 1;
  K := StaNumb;

  WRITELN(TrailOut);
  WRITELN(TrailOut);
  WRITELN(TrailOut);
  WRITELN(TrailOut,'The two stations participating in the intersection ');
  WRITELN(TrailOut,'calculations are:                                  ');
  WRITELN(TrailOut,'                    Station #',NewData[N].Orgin:1);
  WRITELN(TrailOut,'                    Station #',NewData[N + 1].Orgin:1);
  WRITELN(TrailOut);
  WRITELN(TrailOut,'The following information determines the back');
  WRITELN(TrailOut,'LOB"s to:                                    ');

  A := NewData[N].Orgin;
  B := NewData[N + 1].Orgin;

  A := NewData[A].Orgin;
  B := NewData[B].Orgin;

  CASE StaNumb OF

    2 : WRITELN(TrailOut,'                    No other station');

    3 : WRITELN(TrailOut,'                    Station #',NewData[K].Orgin:1);

    4 : BEGIN

          WRITELN(TrailOut,'                    Station #',NewData[K].Orgin:1);
          WRITELN(TrailOut,'                    Station #',NewData[K - 1].Orgin:

        END;

    5 : BEGIN

          WRITELN(TrailOut,'                    Station #',NewData[K].Orgin:1);
          WRITELN(TrailOut,'                    Station #',NewData[K - 1].Orgin:
          WRITELN(TrailOut,'                    Station #',NewData[K - 2].Orgin:

        END;

  END;

  WRITELN(TrailOut);
  WRITELN(TrailOut,'-----------------------------------------------');
  WRITELN(TrailOut,'-----------------------------------------------');
  WRITELN(TrailOut,'-------------Beginning Of Inputted Data-------------');
  WRITE(TrailOut,'----------For Intersecting Station"s ',NewData[N].Orgin:1,' Ar
  WRITE(TrailOut,NewData[N + 1].Orgin:1,'---------');
```

```
WRITELN(TrailOut,'------------------------------------------------');
WRITELN(TrailOut,'------------------------------------------------');
WRITELN(TrailOut);
WRITELN(TrailOut);
WRITE(TrailOut,'Intersecting coordinates for Station #',NewData[N].Orgin:1);
WRITE(TrailOut,' and Station #',NewData[N + 1].Orgin:1,' is:');
WRITELN(TrailOut);
WRITELN(TrailOut);
WRITELN(TrailOut,'        (',Intx:6:3,', ',Inty:6:3,')');
WRITELN(TrailOut);
SupportCount := 0;

FOR J := 1 TO StaNumb DO

  IF ((StIdent[J].Reflob = TRUE) AND (StIdent[J].Flag = FALSE)) THEN

    BEGIN

      AllStat[J].Flag := StIdent[J].Flag;
      WRITELN(TrailOut);
      WRITELN(TrailOut,'X-coordinate ',StIdent[J].Xcoord:6:3,' of Station #',
              StIdent[J].Orgin:1);
      WRITELN(TrailOut,'loaded into the system.');
      WRITELN(TrailOut);
      AllStat[J].Xcoord := StIdent[J].Xcoord;
      WRITELN(TrailOut,'Y-coordinate ',StIdent[J].Ycoord:6:3,' of Station #',
              StIdent[J].Orgin:1);
      WRITELN(TrailOut,'loaded into the system.');
      WRITELN(TrailOut);
      AllStat[J].Ycoord := StIdent[J].Ycoord;
      DistX := (Intx - AllStat[J].Xcoord);  (* could be ABS *)
      DistY := (Inty - AllStat[J].Ycoord);  (* could be ABS *)

      IF DistY = 00.000 THEN

        BEGIN

          WRITELN(TrailOut);
          WRITELN(TrailOut,' This station can not exist at the location');
          WRITELN(TrailOut,' of (  ',DistX:6:3,', ',DistY:6:3,')');
          WRITELN(TrailOut);

        END

      ELSE

        BEGIN

          IF ((DistY < 00.000) AND (DistX < 00.000)) THEN

            BEGIN
              AngleTheta := ARCTAN(DistX/DistY) * 180/Pie;
              AngleTheta := AngleTheta + 180.000;

              IF AngleTheta < 00.000 THEN
                AngleTheta := 360.000 + AngleTheta;

            END

          ELSE
```

```
  IF ((DistY < 00.000) AND (DistX > 00.000)) THEN

    BEGIN

      DistY := 00.000 - DistY;
      AngleTheta := ARCTAN(DistX/DistY) * 180/Pie;

    END

  ELSE

    BEGIN

      AngleTheta := ARCTAN(DistX/DistY) * 180/Pie;

    END;

WRITELN(TrailOut);
WRITELN(TrailOut,'The exact computed LOB angle is = ',AngleTheta:2:
WRITELN(TrailOut);
Temp4 := ABS(StIdent[J].Theta - AngleTheta);
StIdent[J].MissAngle := Temp4;
AllStat[J].MissAngle := StIdent[J].MissAngle;

IF (Temp4 >= 360.0) THEN

   Temp4 := Temp4 - 360.0;

IF (Temp4 <= (3 * Sigma)) THEN

  BEGIN

    Temp5 := Temp4 * Temp4;
    Table[A,B].Support := Temp5 + Table[A,B].Support;
    Table[B,A].Support := Temp5 + Table[B,A].Support;
    Table[A,B].SupportFlag[StIdent[J].Orgin] := TRUE;
    Table[B,A].SupportFlag[StIdent[J].Orgin] := TRUE;
    SupportCount := SupportCount + 1;
    Table[A,B].SupCountFlag := SupportCount;

  END

ELSE

  BEGIN

    Table[A,B].SupportFlag[StIdent[J].Orgin] := FALSE;
    Table[B,A].SupportFlag[StIdent[J].Orgin]:= FALSE;
    WRITE(TrailOut,'Station #',StIdent[J].Orgin:1,'"s exact LOB');
    WRITE(TrailOut,' is greater than 3 sigma.');
    WRITELN(TrailOut);

  END;

WRITELN(TrailOut);
WRITELN(TrailOut,'Difference between the back LOB angle and the');
WRITE(Trailout,'actual angle for Station #',StIdent[J].Orgin:1);
WRITE(TrailOut,' = ',Temp4:6:3,'.');
WRITELN(TrailOut);
WRITELN(TrailOut);
```

```
          END;

        END

      ELSE

        IF ((StIdent[J].Reflob = TRUE) AND (StIdent[J].Flag = TRUE)) THEN

          BEGIN

            AllStat[J].Xcoord     := StIdent[J].Xcoord;
            AllStat[J].Ycoord     := StIdent[J].Ycoord;
            AllStat[J].Alpha      := StIdent[J].Alpha;
            AllStat[J].Theta      := StIdent[J].Theta;
            AllStat[J].Reflob     := StIdent[J].Reflob;
            AllStat[J].Flag       := StIdent[J].Flag;
            AllStat[J].MissAngle  := StIdent[J].MissAngle;
            AllStat[J].Orgin      := StIdent[J].Orgin;

          END

        ELSE

          BEGIN

            AllStat[J].Xcoord     := 00.000;
            AllStat[J].Ycoord     := 00.000;
            AllStat[J].Alpha      := 00.000;
            AllStat[J].Theta      := 00.000;
            AllStat[J].Reflob     := FALSE;
            AllStat[J].Flag       := FALSE;
            AllStat[J].MissAngle  := 00.000;
            AllStat[J].Orgin      := StIdent[J].Orgin;

          END;

    WRITELN(TrailOut,'-----------------------------------------------');
    WRITELN(TrailOut,'----------------End Of Inputted Data----------------');
    WRITE(TrailOut,'----------For Intersecting Station"s ',NewData[N].Orgin:1,'
    WRITE(TrailOut,NewData[N + 1].Orgin:1,'---------');
    WRITELN(TrailOut);
    WRITELN(TrailOut,'-----------------------------------------------');
    WRITELN(TrailOut);
    WRITELN(TrailOut);
    WRITELN(TrailOut);
    WRITELN(TrailOut);
    WRITELN(TrailOut);
    WRITELN(TrailOut);
    WRITELN('            Leaving  ILOBS');
    Temp5 := 00.000;

END; (* PROCEDURE "ILOBS" *)

(******************************************************************************)
PROCEDURE XSEC(      StIdent   : Lobtype;
                     StaNumb   : INTEGER;
                     I         : INTEGER;
                     J         : INTEGER;
                 VAR Intx      : REAL;
                 VAR Inty      : REAL;
                 VAR Computed  : BOOLEAN);
```

```
(**************************************************************************)
(*
   This procedure uses the two LOBS to determine if there is an
   intersecton using the criteria for a valid fix estimate.
   intersecting coordinates are INTX and INTY. " XSEC" establishes a
   series of LOBS called INTLO from all stations to the intersection
   point.

*)

VAR

  X, Y           : REAL; (* Dummy variables for Intx and Inty *)
  Xdist          : REAL; (* Distance of X1 - X2 *)
  Ydist          : REAL; (* Distance of Y1 - Y2 *)
  List1, List2   : Intlotype; (* used for computing intercept coords. *)
  AllStat        : Lobtype; (* data structure that will contain all of
                               the LOB's that return to station that
                               did not attempt any LOB's. *)

BEGIN


  WRITELN('          Entering XSEC');


  X := Intx;
  Y := Inty;
  Xdist := 00.000;
  Ydist := 00.000;
  Intx  := 00.000;
  Inty  := 00.000;


  Xdist := (StIdent[I].Xcoord - StIdent[J].Xcoord); (* could be ABS *)
  Ydist := (StIdent[I].Ycoord - StIdent[J].Ycoord); (* could be ABS *)
  XCOMP(I,J,Xdist,Ydist,StIdent,Intx,Inty,Computed);

  IF (Computed = FALSE) THEN

    BEGIN

      WRITELN(TrailOut);
      WRITELN(TrailOut,' No intersection found from this data using');
      WRITE(TrailOut,' Station #',StIdent[I].Orgin:1,' and ');
      WRITE(TrailOut,' Station #',StIdent[J].Orgin:1);
      WRITELN(TrailOut);
      WRITELN(TrailOut);

    END;

  WRITELN('          Leaving  XSEC');

END; (* PROCEDURE "XSEC" *)

(**************************************************************************)
PROCEDURE  ZEXI(VAR Table : TVT;
                    Lob   : Lobtype);
(**************************************************************************)
```

```
   the test for validity ( i.e. > 0.6 ) then the corresponding value in
   the Truth table will be marked indicating that the intersection failed
   the validity test.
*)

VAR

  BinNumber : INTEGER;
  AStation  : INTEGER;

BEGIN

  WRITELN('            Entering ZEXI');

  FOR BinNumber := 1 TO 5 DO
    FOR AStation := 1 TO 5 DO
     IF Table[BinNumber,AStation].Flag = TRUE THEN
       IF (ABS(Lob[BinNumber].Xcoord - Lob[BinNumber].Ycoord)) > 0.6 THEN
           Table[BinNumber,AStation].Flag := FALSE;

  WRITELN('            Leaving  ZEXI');

END; (* PROCEDURE "ZEXI" *)

(*****************************************************************************
PROCEDURE RECONVERT(     Lobrec : Lobtype;
                     VAR Starec : Lobtype);
(*****************************************************************************
(* This subroutine takes the data stored in "Lobrec"and places it in the
   original order in the data structure called "Starec".
*)

VAR

  J        : INTEGER;
  Location : INTEGER;

BEGIN

  WRITELN('            Entering RECONVERT');

  FOR Location := 1 TO 5 DO

      BEGIN

        J                  := Lobrec[Location].Orgin;
        Starec[J].Xcoord   := Lobrec[Location].Xcoord;
        Starec[J].Ycoord   := Lobrec[Location].Ycoord;
        Starec[J].Theta    := Lobrec[Location].Theta;
        Starec[J].Alpha    := Lobrec[Location].Alpha;
        Starec[J].RefLob   := Lobrec[Location].RefLob;
        Starec[J].Flag     := Lobrec[Location].Flag;

      END;

  WRITELN('            Leaving  RECONVERT');

END; (* PROCEDURE "RECONVERT" *)
```



```
(*****************************************************************************
PROCEDURE CONVERT(     Starec : Lobtype;
```

```
                    VAR Lobrec : Lobtype);
(*******************************************************************************)
(* This subroutine takes the data stored in "Starec" and places it in a
   sequenced order starting in the first and subsequent cells of the data
   structure "Lobtype".
*)

VAR

  J, K      : INTEGER;
  Location  : INTEGER;

BEGIN

  WRITELN('            Entering CONVERT');

  J := 0;
  K := 6;

  FOR Location := 1 TO 5 DO

    IF Starec[Location].Reflob = TRUE THEN

      BEGIN

        J := J + 1; (* index for Lobrec *)
        Lobrec[J].Xcoord := Starec[Location].Xcoord;
        Lobrec[J].Ycoord := Starec[Location].Ycoord;
        Lobrec[J].Theta  := Starec[Location].Theta;
        Lobrec[J].Alpha  := Starec[Location].Alpha;
        Lobrec[J].Reflob := Starec[Location].Reflob;
        Lobrec[J].Flag   := Starec[Location].Flag;
        Lobrec[J].Orgin  := Location;

      END

    ELSE

      BEGIN

        K := K - 1; (* index for Lobrec *)
        Lobrec[K].Xcoord := Starec[Location].Xcoord;
        Lobrec[K].Ycoord := Starec[Location].Ycoord;
        Lobrec[K].Theta  := Starec[Location].Theta;
        Lobrec[K].Alpha  := Starec[Location].Alpha;
        Lobrec[K].Reflob := Starec[Location].Reflob;
        Lobrec[K].Flag   := Starec[Location].Flag;
        Lobrec[K].Orgin  := Location;

      END;

  WRITELN('            Leaving  CONVERT');

END; (* PROCEDURE "CONVERT" *)

(*******************************************************************************)
PROCEDURE GAXI(VAR Table    : TVT;
               VAR Computed : BOOLEAN;
                   Dspxi    : INTEGER;
                   Starec   : Lobtype;
```

```
(*******************************************************************************
(* This procedure extracts the data from the database for use in
   computing the intersection of the LOB's from the BINSETS for each
   station. Parameter "Table" will contain truth-value assignments for
   valid intersections in the Binsets of each Station.
 *)
VAR

  I, J        : INTEGER; (* index for the data structure LOBREC *)
  BinNumber   : INTEGER;
  AStation    : INTEGER;
  NextOne     : INTEGER;
  MaxSupport  : INTEGER;
  TempStation : INTEGER;
  LoopCntr    : INTEGER;
  Location1   : INTEGER;
  Location2   : INTEGER;
  Location3   : INTEGER;
  Location4   : INTEGER;
  X1          : REAL;
  Y1          : REAL;
  Intx        : REAL;
  Inty        : REAL;
  MinExactLob : REAL;
  Lobrec      : Lobtype;
  AllStations : Lobtype; (* keeps track of the back LOB's to the
                            stations whose intersection has not yet
                            been calculated. *)

BEGIN

  WRITELN('            Entering GAXI');

  AStation  := 2;
  TempStation := AStation;
  BinNumber := 1;
  Intx := 00.000;
  Inty := 00.000;

  CASE StaNumb OF

    2 :  LoopCntr := 1;
    3 :  LoopCntr := 3;
    4 :  LoopCntr := 6;
    5 :  LoopCntr := 10;

  END;

  CONVERT(Starec,Lobrec);

  WHILE Dspxi <= LoopCntr DO

    BEGIN

       IF  TempStation = StaNumb + 1 THEN

          BEGIN

            AStation := AStation + 1;
            TempStation := AStation;
            BinNumber := BinNumber + 1;
```

```
      LobrecCBinNumber - 1].Flag := FALSE;

     END;

   IF ((LobrecCBinNumber].Reflob = TRUE) AND
      (LobrecCTempStation].Reflob = TRUE)) THEN

    BEGIN

      XSEC(Lobrec,StaNumb,BinNumber,TempStation,Intx,Inty,Computed);

    END;

   Location3 := LobrecCBinNumber].Orgin;
   Location4 := LobrecCTempStation].Orgin;

   IF Computed = FALSE THEN

    BEGIN

      TableCLocation3,Location4].Flag := FALSE;
      TableCLocation4,Location3].Flag := FALSE;

    END

   ELSE

    BEGIN

      TableCLocation3,Location4].Xintercept := Intx;
      TableCLocation3,Location4].Yintercept := Inty;
      TableCLocation4,Location3].Xintercept := Intx;
      TableCLocation4,Location3].Yintercept := Inty;
      TableCLocation3,Location4].Flag := TRUE;
      TableCLocation4,Location3].Flag := TRUE;
      LobrecCBinNumber].Flag := TRUE;
      LobrecCTempStation].Flag := TRUE;
      ILOBS(AllStations,Lobrec,Intx,Inty,StaNumb,Table);

      FOR I := 1 TO StaNumb DO

        IF AllStationsCI].Flag = FALSE THEN

          BEGIN

            J := LobrecCI].Orgin;
            StarecCJ].MissAngle := AllStationsCI].MissAngle;

          END;

    END;

   Dspxi := Dspxi + 1;
   TempStation := TempStation + 1;
   LobrecCTempStation - 1].Flag := FALSE;

  END;

RECONVERT(Lobrec,Starec);
```

```
WRITELN(TrailOut,'This is the table that shows the relation of data');
WRITELN(TrailOut,'between any of the stations.');
WRITELN(TrailOut);
WRITE(TrailOut,'------------ -------                ---------           --------
WRITE(TrailOut,'    -------');
WRITELN(TrailOut);
WRITE(TrailOut,'Intersecting Stations               Xintercept          Yinterc-
WRITE(TrailOut,'    Support');
WRITELN(TrailOut);
WRITE(TrailOut,'------------ -------                ---------           --------
WRITE(TrailOut,'    -------');
WRITELN(TrailOut);
MinExactLob := 10000000000.000;
NextOne := 5;
MaxSupport := 0;

FOR BinNumber := 1 TO 5 DO
  FOR TempStation := 1 TO 5 DO

    BEGIN

      WRITE(TrailOut,'Station #',BinNumber:1,' Station #',TempStation:1);
      WRITE(TrailOut,'                ');
      WRITE(TrailOut,Table[BinNumber,TempStation].Xintercept:6:3);
      WRITE(TrailOut,'              ');
      WRITE(TrailOut,Table[BinNumber,TempStation].Yintercept:6:3);
      WRITE(TrailOut,'        ',Table[BinNumber,TempStation].SupCountFlag:6)
      WRITELN(TrailOut);

      IF (((Table[BinNumber,TempStation].SupCountFlag < NextOne) AND
           (Table[BinNumber,TempStation].SupCountFlag > MaxSupport)) AND
           (Table[BinNumber,TempStation].Flag = TRUE)) THEN

        BEGIN

          MaxSupport := Table[BinNumber,TempStation].SupCountFlag;
          Location1 := BinNumber;
          Location2 := TempStation;

        END;

    END;


FOR I := 1 TO 5 DO

  BEGIN

    WRITELN(TrailOut);
    WRITELN(TrailOut,'_______ __ _____');
    WRITELN(TrailOut,'Station #',I:1,' data:');
    WRITELN(TrailOut,'------- -- ----');
    WRITELN(TrailOut);
    WRITELN(TrailOut,'X-coord    = ',Starec[I].Xcoord:6:3);
    WRITELN(TrailOut,'Y-coord    = ',Starec[I].Ycoord:6:3);
    WRITELN(TrailOut,'Theta      = ',Starec[I].Theta:6:3);
    WRITELN(TrailOut,'Alpha      = ',Starec[I].Alpha:6:3);
    WRITELN(TrailOut,'Reflob     = ',Starec[I].Reflob);
    WRITELN(TrailOut,'Flag  D-22 = ',Starec[I].Flag);
    WRITELN(TrailOut,'Semimajor  = ',Starec[I].Semimajor:6:3);
    WRITELN(TrailOut,'Semiminor  = ',Starec[I].Semiminor:6:3);
```

```
      WRITELN(TrailOut,'Distance    = ',Starec[I].Distance:6:3);
      WRITELN(TrailOut,'Orgin       = ',Starec[I].Orgin:2);
      WRITELN(TrailOut);

    END;

  WRITELN(TrailOut,'From the data submitted by each station and the data');
  WRITELN(TrailOut,'extracted from various calculations the best "FIX"');
  WRITELN(TrailOut,'location of the object attempting to transmit is at: ');
  WRITELN(TrailOut);
  WRITE(TrailOut,'            (',Table[Location1,Location2].Xintercept:6:3,',  ');
  WRITE(TrailOut,Table[Location1,Location2].Yintercept:6:3,')');
  WRITELN(TrailOut);
  WRITELN(TrailOut);
  WRITELN(TrailOut);
  WRITELN(TrailOut);
  WRITELN(TrailOut);

  WRITELN('            Leaving GAXI');

END; (* PROCEDURE "GAXI" *)

(*****************************************************************************)
PROCEDURE SUXI(VAR Dspxi : INTEGER);
(*****************************************************************************)
(* This procedure sets up the diplacement variable ( parameter Dspxi )
   and sets up the ability to extract data from the database through
   use of the displacement variable.
*)

BEGIN

  WRITELN('            Entering SUXI');

  Dspxi := 1;  (* -1 in actual program *)

  WRITELN('            Leaving  SUXI');

END; (* PROCEDURE "SUXI" *)

(*****************************************************************************)
PROCEDURE LMARK(VAR Table  : TVT;
                    Lobrec : Lobtype );
(*****************************************************************************)
(* This procedure marks LOBS acceptable for use in fix estimation. Marked
   LOBS for optimixed fix estimation are used by "FPEAK" to obtain a best fix.
*)

VAR

  BinNumber : INTEGER;
  I         : INTEGER;

BEGIN

  WRITELN('            Entering LMARK');

  I := 1;

  FOR BinNumber := 1 TO 5 DO
```

```
    IF ((Lobrec[BinNumber].Xcoord = Lobrec[BinNumber + 1].Xcoord) AND
        (Lobrec[BinNumber].Ycoord = Lobrec[BinNumber + 1].Ycoord)) THEN
         Table[AStation,BinNumber].Flag := TRUE;
    I := I + 1;

  WRITELN('            Leaving  LMARK');

END; (* PROCEDURE "LMARK" *)

(***************************************************************************)
PROCEDURE FIRSTPASS(Lobrec  : Lobtype;
                    StaNumb : INTEGER;
                    Table   : TVT;
                    Intlo   : Intlotype);
(***************************************************************************
(* This sub does whatever.

*)

VAR

  Dspxi    : INTEGER;
  Computed : BOOLEAN;


BEGIN

  WRITELN('            Entering FIRSTPASS');

  Computed := FALSE;
  SUXI(Dspxi);
  GAXI(Table,Computed,Dspxi,Lobrec,StaNumb);

  WRITELN('            Leaving FIRSTPASS');

END; (* PROCEDURE "FIRSTPASS" *)

(***************************************************************************)
PROCEDURE LOADDATA(VAR Lobrec  : Lobtype;
                   VAR StaNumb : INTEGER);
(* This procedure prompts the user for input to the TRAILBLAZER program.
*)

VAR

  Temp6       : INTEGER;
  StationCntr : INTEGER;

BEGIN

  WRITELN('            Entering LOADDATA');


  REPEAT

    WRITELN(TrailOut);
    WRITELN(TrailOut,' How many stations will be reporting LOB"s');
    WRITELN(TrailOut,' on the transmitting object.');
    READ(TrailIn,StaNumb);
    WRITELN(TrailOut,StaNumb);
```

```
  Temp6 := StaNumb;
  IF ((Temp6 <= 0) OR (Temp6 >= 6)) THEN

    BEGIN

      WRITELN(TrailOut);
      WRITELN('This value ',StaNumb:2, ' is invalid');
      WRITELN('Try again with a value from 1 to 5.');
      WRITELN(TrailOut);

    END;

UNTIL ((Temp6 > 0) AND (Temp6 < 6));

REPEAT

  BEGIN

    Temp6 := Temp6 - 1;

    REPEAT

      REPEAT

        WRITELN(TrailOut);
        WRITELN(TrailOut, 'Input the Station that is "FIXING".');
        READ(TrailIn, StationCntr);
        WRITELN(TrailOut, StationCntr);

        IF ((StationCntr <= 0) OR (StationCntr >= 6)) THEN

          BEGIN

            WRITELN(TrailOut);
            WRITELN(TrailOut, 'This value ',StationCntr:2, ' is invalid');
            WRITELN(TrailOut, 'Try again with value from 1 to 5.');
            WRITELN(TrailOut);

          END;

      UNTIL ((StationCntr > 0) AND (StationCntr < 6));

      IF Lobrec[StationCntr].Reflob = TRUE THEN

        BEGIN

          WRITELN(TrailOut);
          WRITELN(TrailOut, ' Station #',StationCntr:1, ' has already been');
          WRITELN(TrailOut, ' processed. Try again with a new station.');
          WRITELN(TrailOut);

        END;

    UNTIL Lobrec[StationCntr].Reflob = FALSE;

    WRITELN(TrailOut);
    WRITELN(TrailOut, 'Input the x-coordinate of Station #',StationCntr:1);
    READ(TrailIn, Lobrec[StationCntr].Xcoord);
    WRITELN(TrailOut, Lobrec[StationCntr].Xcoord:6:3);
    WRITELN(TrailOut);
```

```
      READ(TrailIn,Lobrec[StationCntr].Ycoord);
      WRITELN(TrailOut,Lobrec[StationCntr].Ycoord:6:3);
      WRITELN(TrailOut);
      WRITELN(TrailOut,'Input the angle, in relation to true North, formed');
      WRITELN(TrailOut,'by Station #',StationCntr:1,' and the possible ');
      WRITELN(TrailOut,'location ( FIX ) of the transmitting object');
      READ(TrailIn,Lobrec[StationCntr].Theta);
      WRITELN(TrailOut,Lobrec[StationCntr].Theta:6:3);
      WRITELN(TrailOut);
      IF Lobrec[StationCntr].Theta > 180.00 THEN
         Lobrec[StationCntr].Theta := Lobrec[StationCntr].Theta - 360.00;
      Lobrec[StationCntr].Alpha := 90 - Lobrec[StationCntr].Theta;
      Lobrec[Stationcntr].Reflob := TRUE;
      WRITELN(TrailOut);

    END;

  UNTIL (Temp6 = 0);

  WRITELN('            Leaving  LOADDATA');

END; (* PROCEDURE "LOADDATA" *)

(***********************************************************************)
PROCEDURE ESTMP(Lobrec  : Lobtype;
                IntPass : INTEGER;
                Index   : INTEGER;
                Intlo   : Intlotype;
                Table   : TVT);
(***********************************************************************)
(*
   This procedure is called by "HPFIX" to obtain a best fix
   estimate. A single fix or one of mutilple fixes in either
   the normal mode or the degraded mode may be requested.
   "ESTMP" calls the following procedures and/or functions:

        SUBS                 COMMENTS
        ----                 --------

      1) GAXI  : Extracts two Lines of Berings (LOBS) for
                 computing intersection.
      2) INELL : To determine and ensure that a fix estimate
                 doesn't duplicate the last or any previous fix.
      3) LMARK : Marks LOBS acceptable for use in fix estimation.
      4) MSCAN : Clears marks from database.
      5) SCNDB : Accumulates counts of marked LOBS for each station.
      6) SUXI  : Sets up the ability to extract data from each station.
      7) XSEC  : Uses two LOBS to determine if there is an intersection
                 using the criteria for a valid fix estimate.
      8) ZEXI  : Clears appropiate indicators when an intersection
                 fails validity.

*)

VAR

  Dx, Dy          : REAL;
  Xefix           : REAL;  (* x coordinate of estimate *)
  Yefix           : REAL;  (* y coordinate of estimate *)
  TempXefix       : REAL;  (* temporary variable for Xefix *)
  TempYefix       : REAL;  (* temporary variable for Yefix *)
```

```
    X1, Y1          : REAL;  (* input coordinates of the stations. *)
    Intx, Inty      : REAL;
    Bin             : INTEGER;
    BinNumber       : INTEGER;
    Nsame           : INTEGER;
    Nother          : INTEGER;
    Nmax            : INTEGER;
    Dspxi           : INTEGER;
    Lobcounter      : INTEGER;
    Newcounter      : INTEGER;
    Cntfunction     : INTEGER;
    StaNumb         : INTEGER;
    StationCntr     : INTEGER;
    Mode            : BOOLEAN;  (* normal or degraded *)
    Result          : BOOLEAN;
    Computed        : BOOLEAN;
    Answer          : BOOLEAN;
    Response        : CHAR;
    DbRecord        : Lobtype;

BEGIN

    WRITELN('           Entering ESTMP');

    Dx      := 00.000;
    Dy      := 00.000;
    Xefix   := 00.000;
    Yefix   := 00.000;
    Nsame   := 0;
    Nother  := 0;
    Nmax    := 0;  (* maximum LOB counter ^*)

    LOADDATA(Lobrec,StaNumb);

    FIRSTPASS(Lobrec,StaNumb,Table,Intlo);

    IF IntPass > 1 THEN

       BEGIN

         INELLIPSE(Xefix,Yefix,Answer,Table);
         TempXefix := Xefix;
         TempYefix := Yefix;

         IF Answer = TRUE THEN

           BEGIN

             ZEXI(Table,Lobrec);
             CLEAR(Lobrec, Intlo,Table);

           END

         ELSE

           LMARK(Table,Lobrec);

         WHILE Index > -1 Do

           BEGIN
```

```
    IF Lobrec[Bin].Flag = TRUE THEN

      BEGIN

        Cntfunction := 0;
        SCNDB(Cntfunction,Index,Lobrec);

      END

    ELSE

      Lobcounter := Lobcounter + 1;

    Index := Index - 1;

  END;

IF Nother = 0 THEN

  BEGIN

    IF Mode = TRUE THEN
      IF Nsame >= 4 THEN

        BEGIN

          Newcounter := Nsame + Nother;
          IF Nmax >= Newcounter THEN

            BEGIN

              Nmax := Newcounter;
              Xefix := Intx;
              Yefix := Inty;
              Nsame := 0;
              Nother := 0;
              Newcounter := 0;

            END

          ELSE

            BEGIN

              ZEXI(Table,Lobrec);
              CLEAR(Lobrec,Intlo,Table);
              Nsame := 0;
              Nother := 0;
              Nmax := 0;
              MSCAN(Lobrec);

            END;

        END;

  END

ELSE

  BEGIN
```

```
            Newcounter := Nsame + Nother;
            IF Nmax >= Newcounter THEN

              BEGIN

                Nmax := Newcounter;
                Xefix := Intx;
                Yefix := Inty;
                Nsame := 0;
                Nother := 0;

              END;

          END;

      END;

   WRITELN('            Leaving ESTMP');

END; (* PROCEDURE "ESTMP" *)

(*************************************************************************************)
(*************************************************************************************)
(*************************************************************************************)
(* MAIN PROGRAM *)

BEGIN

   REWRITE(TrailOut);
   RESET(TrailIn);
   WRITELN(TrailOut);
   WRITELN(TrailOut);
   WRITELN(TrailOut,'----------------------------------------');
   WRITELN(TrailOut,'**** BEGINNING PROGRAM TRAILBLAZER ****');
   WRITELN(TrailOut,'----------------------------------------');
   WRITELN(TrailOut);
   WRITELN(TrailOut);

   Index := 0;
   IntPass := 1; (* First pass through the system *)

   CLEAR(Lob,Intlo,Table);

   ESTMP(Lob,IntPass,Index,Intlo,Table);

   WRITELN(TrailOut);
   WRITELN(TrailOut);
   WRITELN(TrailOut,'----------------------------------------');
   WRITELN(TrailOut,'**** PROGRAM TRAILBLAZER COMPLETED ****');
   WRITELN(TrailOut,'----------------------------------------');
   WRITELN(TrailOut);
   WRITELN(TrailOut);

END. (* MAIN PROGRAM "TRAILBLAZER" *)
```

```
-------------------------------------
**** BEGINNING PROGRAM TRAILBLAZER ****
-------------------------------------


 How many stations will be reporting LOB"s
 on the transmitting object.
        4

Input the Station that is "FIXING".
        1

Input the x-coordinate of Station #1
 10.000

Input the y-coordinate of Station #1
 18.000

Input the angle, in relation to true North, formed
by Station #1 and the possible
location ( FIX ) of the transmitting object
 50.000


Input the Station that is "FIXING".
        2

Input the x-coordinate of Station #2
 20.000

Input the y-coordinate of Station #2
 5.000

Input the angle, in relation to true North, formed
by Station #2 and the possible
location ( FIX ) of the transmitting object
 33.000


Input the Station that is "FIXING".
        3

Input the x-coordinate of Station #3
 60.000

Input the y-coordinate of Station #3
 25.000

Input the angle, in relation to true North, formed
by Station #3 and the possible
location ( FIX ) of the transmitting object
-5.000
```



```
Input the Station that is "FIXING".
```

4

```
Input the x-coordinate of Station #4
 90.000

Input the y-coordinate of Station #4
 25.000

Input the angle, in relation to true North, formed
by Station #4 and the possible
location ( FIX ) of the transmitting object
 312.000




The two stations participating in the intersection
calculations are:
                   Station #1
                   Station #2

The following information determines the back
LOB"s to:
                   Station #3
                   Station #4

----------------------------------------------------
----------------------------------------------------
--------------Beginning Of Inputted Data-------------
----------For Intersecting Station"s 1 And 2---------
----------------------------------------------------
----------------------------------------------------


Intersecting coordinates for Station #1 and Station #2 is:

       (50.525, 52.005)


X-coordinate  60.000 of Station #3
loaded into the system.

Y-coordinate  25.000 of Station #3
loaded into the system.


The exact computed LOB angle is = -19.334


Difference between the back LOB angle and the
actual angle for Station #3 =  14.334.


X-coordinate  90.000 of Station #4
loaded into the system.

Y-coordinate  25.000 of Station #4
loaded into the system.
```

```
The exact computed LOB angle is = -55.624


Difference between the back LOB angle and the
actual angle for Station #4 =  7.624.

-------------------------------------------------
----------------End Of Inputted Data-------------
----------For Intersecting Station"s 1 And 2-----
-------------------------------------------------
```

```
The two stations participating in the intersection
calculations are:
                   Station #1
                   Station #3

The following information determines the back
LOB"s to:
                   Station #2
                   Station #4

-------------------------------------------------
-------------------------------------------------
--------------Beginning Of Inputted Data----------
----------For Intersecting Station"s 1 And 3------
-------------------------------------------------
-------------------------------------------------


Intersecting coordinates for Station #1 and Station #3 is:

       (57.151, 57.564)


X-coordinate  20.000 of Station #2
loaded into the system.

Y-coordinate  5.000 of Station #2
loaded into the system.


The exact computed LOB angle is =  35.252


Difference between the back LOB angle and the
actual angle for Station #2 =  2.252.


X-coordinate  90.000 of Station #4
loaded into the system.

Y-coordinate  25.000 of Station #4
loaded into the sustem.
```

The exact computed LOB angle is = -45.249

Difference between the back LOB angle and the
actual angle for Station #4 = 2.751.

---------------End Of Inputted Data----------------
---------For Intersecting Station"s 1 And 3---------

The two stations participating in the intersection
calculations are:
Station #1
Station #4

The following information determines the back
LOB"s to:
Station #2
Station #3

-------------Beginning Of Inputted Data------------
---------For Intersecting Station"s 1 And 4---------

Intersecting coordinates for Station #1 and Station #4 is:

(55.434, 56.124)

X-coordinate 20.000 of Station #2
loaded into the system.

Y-coordinate 5.000 of Station #2
loaded into the system.

The exact computed LOB angle is = 34.726

Difference between the back LOB angle and the
actual angle for Station #2 = 1.726.

X-coordinate 60.000 of Station #3
loaded into the system.

Y-coordinate 25.000 of Station #3
loaded into the system.

The exact computed LOB angle is = -8.346

Difference between the back LOB angle and the
actual angle for Station #3 = 3.346.

----------------End Of Inputted Data----------------
----------For Intersecting Station"s 1 And 4----------

The two stations participating in the intersection
calculations are:

Station #2
Station #3

The following information determines the back
LOB"s to:

Station #1
Station #4

--------------Beginning Of Inputted Data-------------
----------For Intersecting Station"s 2 And 3----------

Intersecting coordinates for Station #2 and Station #3 is:

(56.793, 61.656)

X-coordinate 10.000 of Station #1
loaded into the system.

Y-coordinate 18.000 of Station #1
loaded into the system.

The exact computed LOB angle is = 46.986

Difference between the back LOB angle and the
actual angle for Station #1 = 3.014.

X-coordinate 90.000 of Station #4

loaded into the system.

Y-coordinate 25.000 of Station #4
loaded into the system.

The exact computed LOB angle is = -42.174

Difference between the back LOB angle and the
actual angle for Station #4 = 5.826.

---------------End Of Inputted Data---------------
---------For Intersecting Station"s 2 And 3---------

The two stations participating in the intersection
calculations are:

Station #2
Station #4

The following information determines the back
LOB"s to:

Station #1
Station #3

-------------Beginning Of Inputted Data-------------
---------For Intersecting Station"s 2 And 4---------

Intersecting coordinates for Station #2 and Station #4 is:

(54.024, 57.393)

X-coordinate 10.000 of Station #1
loaded into the system.

Y-coordinate 18.000 of Station #1
loaded into the system.

The exact computed LOB angle is = 48.178

Difference between the back LOB angle and the
actual angle for Station #1 = 1.822.

X-coordinate 60.000 of Station #3
loaded into the system.

Y-coordinate 25.000 of Station #3
loaded into the system.

The exact computed LOB angle is = -10.452

Difference between the back LOB angle and the
actual angle for Station #3 = 5.452.

---------------End Of Inputted Data----------------
---------For Intersecting Station"s 2 And 4---------

The two stations participating in the intersection
calculations are:
Station #3
Station #4

The following information determines the back
LOB"s to:
Station #1
Station #2

-------------Beginning Of Inputted Data-------------
---------For Intersecting Station"s 3 And 4---------

Intersecting coordinates for Station #3 and Station #4 is:

(57.435, 54.322)

X-coordinate 10.000 of Station #1
loaded into the system.

Y-coordinate 18.000 of Station #1
loaded into the system.

The exact computed LOB angle is = 52.558

Difference between the back LOB angle and the actual angle for Station #1 = 2.558.

X-coordinate 20.000 of Station #2
loaded into the system.

Y-coordinate 5.000 of Station #2
loaded into the system.

The exact computed LOB angle is = 37.198

Difference between the back LOB angle and the
actual angle for Station #2 = 4.198.

----------------End Of Inputted Data----------------
---------For Intersecting Station"s 3 And 4---------

This is the table that shows the relation of data
between any of the stations.

Intersecting Stations	Xintercept	Yintercept	Support
Station #1 Station #1	0.000	0.000	0
Station #1 Station #2	50.525	52.005	2
Station #1 Station #3	57.151	57.564	0
Station #1 Station #4	55.434	56.124	2
Station #1 Station #5	0.000	0.000	0
Station #2 Station #1	50.525	52.005	2
Station #2 Station #2	0.000	0.000	0
Station #2 Station #3	56.793	61.656	0
Station #2 Station #4	54.024	57.393	0
Station #2 Station #5	0.000	0.000	0
Station #3 Station #1	57.151	57.564	0
Station #3 Station #2	56.793	61.656	0
Station #3 Station #3	0.000	0.000	0
Station #3 Station #4	57.435	54.322	2
Station #3 Station #5	0.000	0.000	0
Station #4 Station #1	55.434	56.124	2
Station #4 Station #2	54.024	57.393	0
Station #4 Station #3	57.435	54.322	0
Station #4 Station #4	0.000	0.000	0
Station #4 Station #5	0.000	0.000	0
Station #5 Station #1	0.000	0.000	0
Station #5 Station #2	0.000	0.000	0
Station #5 Station #3	0.000	0.000	0
Station #5 Station #4	0.000	0.000	0
Station #5 Station #5	0.000	0.000	0

```
------- -- ----

X-coord    =   10.000
Y-coord    =   18.000
Theta      =   50.000
Alpha      =   40.000
Reflob     =   TRUE
Flag       =   FALSE
Semimajor  =    0.000
Semiminor  =    0.000
Distance   =    0.000
Orgin      =    0


Station #2 data:
------- -- ----

X-coord    =   20.000
Y-coord    =    5.000
Theta      =   33.000
Alpha      =   57.000
Reflob     =   TRUE
Flag       =   FALSE
Semimajor  =    0.000
Semiminor  =    0.000
Distance   =    0.000
Orgin      =    0


Station #3 data:
------- -- ----

X-coord    =   60.000
Y-coord    =   25.000
Theta      =   -5.000
Alpha      =   95.000
Reflob     =   TRUE
Flag       =   TRUE
Semimajor  =    0.000
Semiminor  =    0.000
Distance   =    0.000
Orgin      =    0


Station #4 data:
------- -- ----

X-coord    =    90.000
Y-coord    =    25.000
Theta      =   -48.000
Alpha      =   138.000
Reflob     =   TRUE
Flag       =   FALSE
Semimajor  =     0.000
Semiminor  =     0.000
Distance   =     0.000
Orgin      =     0
```

```
Station #5 data:
------- -- ----

X-coord    =  0.000
Y-coord    =  0.000
Theta      =  0.000
Alpha      =  0.000
Reflob     =  FALSE
Flag       =  FALSE
Semimajor  =  0.000
Semiminor  =  0.000
Distance   =  0.000
Orgin      =  0

From the data submitted by each station and the data
extracted from various calculations the best "FIX"
location of the object attempting to transmit is at:

          (50.525,   52.005)



---------------------------------------
**** PROGRAM TRAILBLAZER COMPLETED ****
---------------------------------------
```

APPENDIX E

DATA BASE ENTRIES FOR TRAILBLAZER

Page intentionally left blank

PART I
TRAILBLAZER

TABLE OF CONTENTS

1. TRAILBLAZER Overview

1 TRAILBLAZER PROCESS DESCRIPTION:
TRAILBLAZER is a manned, ground-based direction finding system. The system functions by intercepting targets at the master control station (MCS) and providing at least two DF bearings from some combination of MCS and remote slave station (RSS) subsystems. Successful deployment of the system requires a line-of-sight (LOS) to the target area and to the other subsystems.

SYNONYM: AN/TSQ-114

2. Utilizes Analysis for HPFIXM

TRAILBLAZER

Utilizes Structure

COUNT	LEVEL	NAME
1	1.0	TB_Hpfixm
2	1.1	TB_Datset
3	1.2	TB_Disf
4	1.3	TB_Estat
5	1.4	TB_Hpfix
6	1.4.1	TB_Autcl
7	1.4.2	TB_Dbcpy
8	1.4.2.1	TB_Memr
9	1.4.3	TB_Estat
10	1.4.4	TB_Estmp
11	1.4.4.1	TB_Suxi
12	1.4.4.2	TB_Gaxi
13	1.4.4.3	TB_Xsec
14	1.4.4.3.1	TB_Ilobs
15	1.4.4.3.1.1	TB_Atan2
16	1.4.4.3.2	TB_Lfin
17	1.4.4.3.3	TB_Xcomp
18	1.4.4.3.4	TB_Xprep
19	1.4.4.3.4.1	TB_Cos
20	1.4.4.3.4.2	TB_Sin
21	1.4.4.4	TB_Inell
22	1.4.4.5	TB_Lmark
23	1.4.4.5.1	TB_Sclmk
24	1.4.4.6	TB_Scndb
25	1.4.4.6.1	TB_Ipot
26	1.4.4.7	TB_Zexi
27	1.4.4.8	TB_Mscan
28	1.4.5	TB_Final
29	1.4.5.1	TB_Atan2
30	1.4.5.2	TB_Cos
31	1.4.5.3	TB_Flobs
32	1.4.5.3.1	TB_Atan2
33	1.4.5.3.2	TB_Lfin
34	1.4.5.4	TB_Fpot
35	1.4.5.4.1	TB_Scndb
36	1.4.5.4.1.1	TB_Ipot
37	1.4.5.4.2	TB_Sqrt
38	1.4.5.5	TB_Fsqrt
39	1.4.5.6	TB_Sin
40	1.4.5.7	TB_Scndb
41	1.4.5.7.1	TB_Ipot
42	1.4.5.8	TB_Sqrt
43	1.4.6	TB_Fpeak
44	1.4.6.1	TB_Flobs
45	1.4.6.1.1	TB_Atan2
46	1.4.6.1.2	TB_Lfin
47	1.4.6.2	TB_Fmnew
48	1.4.6.2.1	TB_Fmsam
49	1.4.6.3	TB_Fmsam

COUNT	LEVEL	NAME
50	1.4.6.4	TB_Fmset
51	1.4.6.5	TB_Lmark
52	1.4.6.5.1	TB_Sclmk
53	1.4.6.6	TB_Mkact
54	1.4.6.7	TB_Fpot
55	1.4.6.7.1	TB_Scndb
56	1.4.6.7.1.1	TB_Ipot
57	1.4.6.7.2	TB_Sqrt
58	1.4.6.8	TB_Inell
59	1.4.6.9	TB_Zexit
60	1.4.6.10.	TB_Mscan
61	1.4.6.11.	TB_Zexi
62	1.4.7	TB_Groom
63	1.4.8	TB_Stcpy
64	1.4.9	TB_Mkusd
65	1.4.10.	TB_Mclr
66	1.4.11.	TB_Svell
67	1.4.12.	TB_Odcpy
68	1.4.12.1	TB_Mabin
69	1.4.12.2	TB_Mscan
70	1.4.12.3	TB_Whbin
71	1.4.12.3.1	TB_Lfin
72	1.4.12.4	TB_Xy2ll
73	1.5	TB_Lamp
74	1.6	TB_Pchk
75	1.7	TB_Store

Utilizes Matrix

Explanation of the Utilizes Matrix:

The rows are input PROCESS names, and the columns are PROCESSES UTILIZED by (or a SUBPART of) the rows.

(i, j) value	meaning
U	Column j is UTILIZED by Row i
S	Column j is a PART of Row i
B	Column j is both UTILIZED by, and a PART of Row i

	1 TB_Datset	2 TB_Disf	3 TB_Estat	4 TB_Hpfix	5 TB_Lamp	6 TB_Pchk	7 TB_Store	8 TB_Autcl	9 TB_Dbcpy	10 TB_Estmp	11 TB_Final	12 TB_Fpeak	13 TB_Groom	14 TB_Stcpy	15 TB_Mkusd
1 TB_Hpfixm	U	U	U	U	U	U	U								
2 TB_Hpfix			U					U	U	U	U	U	U	U	U
3 TB_Dbcpy															
4 TB_Estmp															
5 TB_Xsec															
6 TB_Ilobs															
7 TB_Xprep															
8 TB_Lmark															
9 TB_Scndb															
10 TB_Final															

*** Matrix empty for Rows 11 thru 16 and Columns 1 thru 15

TRAILBLAZER

Utilizes Matrix

	16 TB_Mclr	17 TB_Svell	18 TB_Odcpy	19 TB_Memr	20 TB_Suxi	21 TB_Gaxi	22 TB_Xsec	23 TB_Inell	24 TB_Lmark	25 TB_Scndb	26 TB_Zexi	27 TB_Mscan	28 TB_Ilobs	29 TB_Lfin	30 TB_Xcomp
1 TB_Hpfixm															
2 TB_Hpfix	U	U	U												
3 TB_Dbcpy				U											
4 TB_Estmp					U	U	U	U	U	U	U	U			
5 TB_Xsec													U	U	U
6 TB_Ilobs															
7 TB_Xprep															
8 TB_Lmark															
9 TB_Scndb															
10 TB_Final										U					

Utilizes Matrix

	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
11 TB_Flobs														U	
12 TB_Fpot										U					
13 TB_Fpeak								U	U		U	U			
14 TB_Fmnew															
15 TB_Odcpy												U			
16 TB_Whbin														U	

	31 TB_Xprep	32 TB_Atan2	33 TB_Cos	34 TB_Sin	35 TB_Sclmk	36 TB_Ipot	37 TB_Flobs	38 TB_Fpot	39 TB_Fsqrt	40 TB_Sqrt
1 TB_Hpfixm										
2 TB_Hpfix										
3 TB_Dbcpy										
4 TB_Estmp										
5 TB_Xsec	U									
6 TB_Ilobs		U								
7 TB_Xprep			U	U						
8 TB_Lmark					U					
9 TB_Scndb						U				
10 TB_Final		U	U	U			U	U	U	U
11 TB_Flobs		U								
12 TB_Fpot										U
13 TB_Fpeak							U	U		
14 TB_Fmnew										
15 TB_Odcpy										
16 TB_Whbin										

Utilizes Matrix

	41 TB_Fmnew	42 TB_Fmsam	43 TB_Fmset	44 TB_Mkact	45 TB_Zexit	46 TB_Mabin	47 TB_Whbin	48 TB_Xy211
1 TB_Hpfixm								
2 TB_Hpfix								
3 TB_Dbcpy								
4 TB_Estmp								
5 TB_Xsec								
6 TB_Ilobs								
7 TB_Xprep								
8 TB_Lmark								
9 TB_Scndb								
10 TB_Final								
11 TB_Flobs								
12 TB_Fpot								
13 TB_Fpeak	U	U	U	U	U			
14 TB_Fmnew		U						
15 TB_Odcpy						U	U	U
16 TB_Whbin								

3. Attribute Report

REPORT SPECIFICATION:

1 tree-level H='TREE LEVEL' COL=14
2 mathematical-field H='MATHEMATICAL FIELD' COL=25

*** No SYSTEM-PARAMETERS

	TREE LEVEL	MATHEMATICAL FIELD
TB_Atan2	leaf	trigonometry
TB_Cos	leaf	trigonometry
TB_Dbcpy	middle	data_base_handling
TB_Estmp	middle	logical
TB_Final	middle	multivariate_statistics
TB_Flobs	middle	logical
TB_Fmnew	middle	optimization
TB_Fmsam	leaf	optimization
TB_Fmset	leaf	optimization
TB_Fpeak	middle	optimizations
TB_Fpot	middle	optimization
TB_Gaxi	leaf	data_base_handling
TB_Groom	leaf	data_base_manipulation
TB_Hpfix	root	logical
TB_Hpfixm	root	logical
TB_Ilobs	middle	N/A
TB_Inell	leaf	multivariate_statistics
TB_Ipot	leaf	optimization
TB_Mabin	leaf	data_base_handling
TB_Mkact	leaf	data_base_manipulation
TB_Mscan	leaf	data_base_handling
TB_Odcpy	middle	N/A
TB_Sclmk	leaf	data_base_handling
TB_Scndb	middle	optimization
TB_Sin	leaf	trigonometry
TB_Stcpy	leaf	data_base_handling
TB_Suxi	leaf	data_base_manipulation
TB_Svell	leaf	data_base_handling
TB_Whbin	middle	logical
TB_Xcomp	leaf	trigonometry
TB_Xprep	middle	N/A
TB_Xsec	middle	trigonometry
TB_Zexi	leaf	data_base_handling

4. Index and Dictionary Descriptions

```
 1      TB_HPFIX.MOD
 2         TB_FIXD
 3            TB_.ENGU
 4            TB_.DEGU
 5            TB_.PTY
 6            TB_.QUIT
 7            TB_DATSET
 8            TB_DISF
 9            TB_ESTAT
10            TB_HPFIX
11            TB_LAMP
12            TB_PCHK
13            TB_STORE
14         TB_HPFIX
15            TB_ESTAT
16            TB_.FORK
17            TB_AUTCL
18            TB_DBCPY
19            TB_ESTMP
20            TB_FINAL
21            TB_FPEAK
22            TB_GROOM
23            TB_MCLR
24            TB_ODCPY
25            TB_STCPY
26            TB_SVELL
27            TB_XY2LL
28         TB_DBCPY
29            TB_.ENGU
30            TB_.DEGU
31            TB_MEMR
32         TB_ESTMP
33            TB_.ENGU
34            TB_.DEGU
35         TB_FINAL
36            TB_.ENGU
37            TB_.DEGU
38            TB_ATAN2
39            TB_COS
40            TB_SIN
41            TB_SQRT
42         TB_FPEAK
43            TB_.ENGU
44            TB_.DEGU
45         TB_GROOM
46         TB_ODCPY
47            TB_XY2LL
48            TB_MEMR
49         TB_STCPY
50            TB_.ENGU
```

51 TB_.DEQU
52 TB_SVELL

1 TB_HPFIX.MOD PROCESS

DESCRIPTION:
HPFIX is the main module for processing a fix command received from either operator. It computes, displays, and saves fix data as required by the fix commands.

2 TB_FIXD PROCESS

DESCRIPTION:
FIXD is jumped to by COMRET to complete the task of processing the FIX command.

3 TB_.ENQU PROCESS

DESCRIPTION:
Entry point for .ENQU system call

4 TB_.DEQU PROCESS

DESCRIPTION:
Entry point for .DEQU system call.

5 TB_.PTY PROCESS

DESCRIPTION:
Entry for .PTY instruction call.

6 TB_.QUIT PROCESS

DESCRIPTION:
Entry point for .QUIT instruction.

7 TB_DATSET PROCESS

DESCRIPTION:
DATSET is called by FIXD when a fix has been successfully computed. It generates a formatted table of fix information

called the DATSET generated fix table.

8 TB_DISF PROCESS

DESCRIPTION:
DISF is called by FIXD to display each successfully calculated fix and by DISP when processing a DISPLAY command, or during display change processing to re-display fix information.

9 TB_ESTAT PROCESS

DESCRIPTION:
The function of this module/proc is to queue error status messages for display on the AN/UYQ-10 and to assure that the messages are displayed for three seconds. Any module may call ESTAT whenever display of an error/status message is required.

10 TB_HPFIX PROCESS

DESCRIPTION:
HPFIX is called by FIXD to compute a fix and to provide all fix information in a FIX DATA TABLE containing:
the fix point latitude
the fix point longitude
the fix ellipse orientation angle major axis relative to NORTH
the length in km of the fix error ellipse semi-major axis
the length in km of the fix error ellipse semi-minor axis
the display unit number
the LOB display header
the LOB's involved in the fix computation.

11 TB_LAMP PROCESS

DESCRIPTION:
LAMP is called to program panel button lights. It combines the panel unit number and the desired function into a command word, and sends the command to the selected panel.

12 TB_PCHK PROCESS

DESCRIPTION:
This routine simply checks to see if the page currently on display has function which depend on the page, e.g. to see if the current page number is one which permits text editing.

13 TB_STORE PROCESS

DESCRIPTION:
STORE is called by COMRET as a result of a STORE command and by FIXD for each fix in a multi-fix request. STORE saves the latest set of fix data in FIXO

14 TB_FORK PROCESS

DESCRIPTION:
Entry point for .FORK system call.

15 TB_AUTCL PROCESS

DESCRIPTION:
AUTCL is called by HPFIX before computing the first multi-fix, and by CLEAR to process the fix portions of the PURGE and DONE commands or in response to the ELIM command where no specific fixes were specified.

16 TB_DBCPY PROCESS

DESCRIPTION:
DBCPY is called by HPFIX to obtain a local copy of that part of the LOB database which is displayed and not removed. The global database referred to is DATAO/DATA1.

17 TB_ESTMP PROCESS

DESCRIPTION:
ESTMP is called by HPFIX to obtain a best fix estimate for either a single fix or one of multiple fixes in either the normal or degraded mode.

18 TB_FINAL PROCESS

DESCRIPTION:
FINAL is called by HPFIX, after an optimized fix point has been determined, to calculate the parameters of the error ellipses.

19 TB_FPEAK PROCESS
DESCRIPTION:
FPEAK is called by HPFIX to optimize the fix estimate

obtained from ESTMP.

20 TB_GROOM PROCESS

DESCRIPTION:
GROOM is called by HPFIX to initialize the local database for fix computations or to re-initialize it if the fix mode has been degraded to reconsider rejected intersections.

21 TB_MCLR PROCESS

DESCRIPTION:
MCLR is a globally available utility used to update the header text and draw the page border divider and/or graphics for a page on the appropriate AN/UYQ-10 display unit.

22 TB_ODCPY PROCESS

DESCRIPTION:
ODCPY is called by HPFIX, after successful computation of a fix, to store all fix related data in a file.

23 TB_STCPY PROCESS

DESCRIPTION:
STCPY is called by HPFIX to store a local copy of station locations and screen geometry for use in fix computations.

24 TB_SVELL PROCESS

DESCRIPTION:
SVELL is called by HPFIX to save fix and related ellipse parameters.

25 TB_XY2LL PROCESS

DESCRIPTION:
XY2LL is called by HPFIX following completion of a successful fix. It takes X, Y screen coordinates and performs a translation and rotation to get X, Y coordinates relative to North being directly vertical. An inverse Gnomonic projection is the done to get the latitude and longitude of the point.

26 TB_MEMR PROCESS

DESCRIPTION:
MEMR services memory requests.

SYNONYM: TB_ MEMR

27 TB_ATAN2 PROCESS

DESCRIPTION:
ATAN2 is a globally available utility which determines the floating point radian angle whose tangent is X/Y. X and Y are double precision arguments supplied to the function.

28 TB_COS PROCESS

DESCRIPTION:
COS is a globally available utility which, passed an angle in floating point radians, calculates its cosine.

29 TB_SIN PROCESS

DESCRIPTION:
SIN is a globally available utility which, given an angle in floating point radians, calculates its sine.

30 TB_SQRT PROCESS

DESCRIPTION:
SQRT is a globally available utility used to approximate the square root of an input argument.

APPENDIX F

USAMS ALGORITHM ANALYSIS SERIES

1. Analysis of Geographic Transformation Algorithms
   JPL D-181
   DTIC #ADA 129182
   Dated: July 9, 1985

2. Correlation Algorithm Report
   JPL D-182 UAA-003
   DTIC #ADA 129181
   Dated: September 15, 1982

3. Applications of Correlation Techniques for Battlefield Identification I
   JPL D-179 UAA-006
   Dated: June 1984

4. Cross-Correlation: Statistics, Templating, and Doctrine
   JPL D-184
   DTIC #ADA 155624
   Dated: February 29, 1984

5. Intelligence Algorithm Methodology I
   JPL D-183 UAA-004
   DTIC #ADB 078293
   Dated: August 15, 1983

6. Intelligence Algorithms in Target Analysis and Planning (TAP)
   JPL D-178 UAA-007
   DTIC #ADB 092402L
   Dated: November 30, 1984

7. Intelligence Algorithm Methodology II: An Intelligence/Electronic Warfare (IEW) Tactical Sensors Model

   JPL D-185 UAA-008

   Dated: 1985

8. Intelligence/Electronic Warfare (IEW) Direction-Finding and Fix Estimation Analysis Report

   Volume 1, Overview

   JPL-180, Vol. 1 UAA-001

9. Intelligence/Electronic Warfare (IEW) Direction-Finding and Fix Estimation Analysis Report

   Volume 3, GUARDRAIL

   JPL-180, Vol. 3 UAA-001

   Dated: December 1985

10. A Non-Standard Probabilistic Position-Fixing Model

    JPL D-186 UAA-009

    Dated: June 1985

11. A Collection of Area of Interest (AOI) Algorithms

    JPL D-171 UAA-011

    Dated: July 1985

12. Power of Statistical Tests Used in Correlation Techniques for Battlefield Identification

    JPL D-2793 UAA-016

    Formerly Technical Memorandum No. 5

    Dated: August 1985

13. Testing and Combination of Confidence Ellipses: A Geometric Analysis

    JPL D-2782 UAA-013

    Formerly Technical Memorandum No. 2

    Dated: August 5, 1985

14. Wild Bearings Analysis
JPL D-2783 UAA-014
Formerly Technical Memorandum No. 3
Dated: July 10, 1985

15. Collection and Analysis of Specific ELINT Signal Parameters
JPL D-2781 UAA-012
Formerly Technical Memorandum No. 1
Dated: June 23, 1985

16. IEW Sensor Error Budget for DF Fix Estimations
Technical Memorandum No. 4
Dated: August 14, 1985

17. Confidence Ellipse Research Software
JPL D-2786 UAA-015
Technical Memorandum No. 6
Dated: August 8, 1985

18. The Power of Statistical Tests - Software
JPL D-2788 UAA-017
Technical Memorandum No. 7
Dated: December 2, 1985

19. Collection and Analysis of Specific Elint
Signal Parameters: Final Report
JPL D-2787 UAA-016
Technical Memorandum No. 8
Dated: December 9, 1985

20. Intelligence/Electronic Warfare (IEW) Direction-Finding
and Fix Estimation Analysis Report
Volume 2, TRAILBLAZER
JPL D-180, Vol. 2 UAA-001
Dated: December 20, 1985