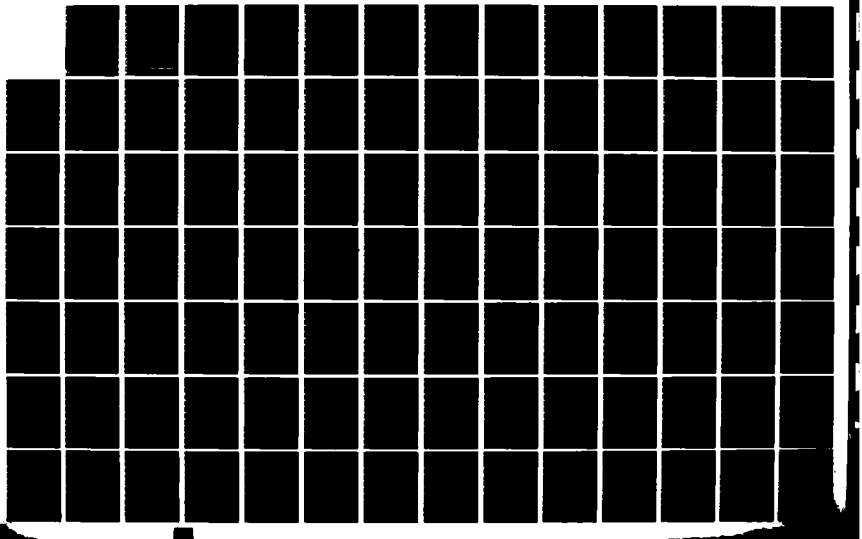
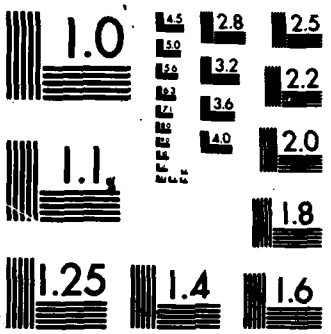
AD-A137 564 APPLICATIONS OF CLUSTER ANALYSES AND INTEGER PROGRAMMING TO MULTIPLE TARG.. (U) TRACOR INC AUSTIN TX APPLIED SCIENCES GROUP D COOPER ET AL. 21 MAR 83 1/2

UNCLASSIFIED TRACOR-T83-AU-9552-U N00014-82-C-0044 F/G 12/1 NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

(6)

FINAL REPORT

APPLICATIONS OF CLUSTER ANALYSES AND INTEGER PROGRAMMING TO MULTIPLE TARGET TRACKING

21 March 1983

Tracor Applied Sciences

Tracor Inc 6500 Tracor Lane Austin Texas 7872

Contract No. N00014-82-C-0044
Tracor Project 034-010
Document No. T83-AU-9552-U

FINAL REPORT

APPLICATIONS OF CLUSTER ANALYSES AND INTEGER PROGRAMMING TO MULTIPLE TARGET TRACKING

21 March 1983

Submitted to

Scientific Officer
Statistics and Probability Program
Office of Naval Research
800 North Quincy Street
Arlington, Virginia 22217

Attention : Dr. Ed Wegman

Submitted

Don Cooper

Don Cooper

Glenn Corser

Glenn Corser

Approved

Frank C. Harris

Frank C. Harris
Department Director
System Modeling Department

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER: T83-AU-9552-U	2. GOVT ACCESSION NO: AD-A137564
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle): Final Report: Applications of Cluster Analyses and Integer Programming to Multiple Target Tracking	5. TYPE OF REPORT & PERIOD COVERED: Final Report 1 April 1982 - 31 Mar 1983
6. PERFORMING ORG. REPORT NUMBER: T83-AU-9552-U	
7. AUTHOR(s): Don Cooper, Glenn Corser, Frank Harris	8. CONTRACT OR GRANT NUMBER(s): N00014-82-C-0044
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Tracor, Inc. Applied Sciences Group 6500 Tracor Lane Austin, TX 78721	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 65152N, RO145-TW, NR 274-299
11. CONTROLLING OFFICE NAME AND ADDRESS: Probability and Statistics Program Office of Naval Research 800 North Quincy St. Arlington, VA 22217	12. REPORT DATE: 21 March 1983
	13. NUMBER OF PAGES: 153
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): N/A	15. SECURITY CLASS. (of this report): Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE: N/A

16. DISTRIBUTION STATEMENT (of this Report)

Approved for Public Release: Distribution Unlimited

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

N/A

18. SUPPLEMENTARY NOTES

N/A

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

Target Tracking
Localization
Multi-Target Tracking
Automatic Data Sorting
Cluster Analysis
Integer Programming

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

See attachment

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

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. ABSTRACT

A Multiple-Target Tracking Algorithm has been designed and tested for use with DIFAR sonobuoys. The algorithm is implemented as a functionally modular computer program composed of five main subroutines. Operating in near real time and in less than 64K words of memory, the algorithm is capable of separating raw measurements into data sets corresponding to individual targets at the sensor level (using cluster analysis), correlating target measurements across sensors (using physical constraint and statistical tests), and selecting the most likely track scenario among the various potential scenarios (using integer programming). The process requires no operator decision and no a priori information about the number or initial conditions of the targets. The algorithm is self initializing from the raw buoy data. Using synthetic data, the algorithm was tested on several multi-target scenarios with excellent results.

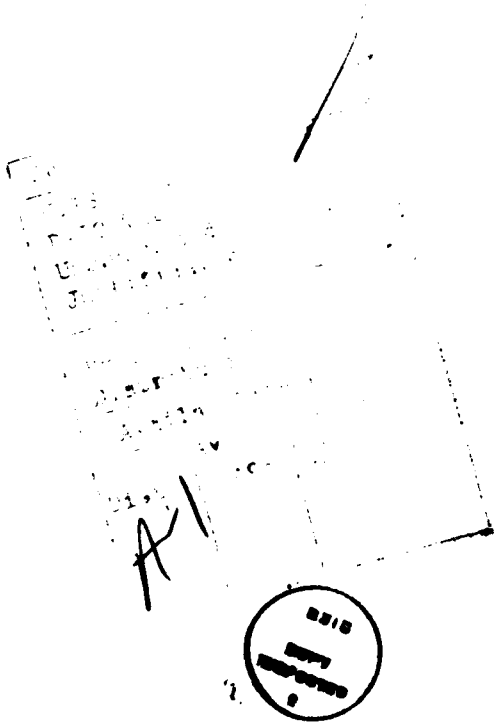


TABLE OF CONTENTS

TABLE OF CONTENTS (CONTINUED)

TABLE OF CONTENTS (CONCLUDED)

LIST OF FIGURES

LIST OF TABLES

EXECUTIVE SUMMARY

During the past year Tracor created and tested a Multiple-Target Tracking Algorithm (MTTA) designed for use with DIFAR sonobuoys. Operating in near real-time and in less than 64 K words of memory, the algorithm is capable of separating raw measurements into data sets corresponding to individual targets at the sensor level, correlating target measurements across sensors, and selecting the most likely track scenario among the various potential scenarios. The process requires no operator decision and no a priori information about the number or initial conditions of the targets. The algorithm is self initializing from the raw buoy data. Using synthetic data, the algorithm was tested on several multi-target scenarios with excellent results. This executive summary presents an overview of work accomplished on the MTTA during the past year. Included is a brief description of the algorithm's structure, conclusions of the study, and recommendations for further work.

The MTTA is implemented as a functionally modular computer program composed of five main subroutines. At present all processing is serial; however, with little effort the algorithm could be implemented in a parallel processing scheme.

Sorting Data at Sensor Level-Cluster Analysis. When plots of frequency versus bearing measurements from a DIFAR sensor are constructed, the noise free data lie along either straight or curvilinear lines. For targets that are moderately separated in bearing or frequency, the associated lines will be quite distinct. Based on these observations, experiments were conducted to test the usefulness of cluster analytic techniques to separate target data at the sensor level. They proved to be successful and Ling's (1,r) algorithm

(equivalent to a single-linkage clustering algorithm) was used to separate data at the sensor level into potential target data sets.

Automatic Cluster Extraction. While cluster analysis worked well in separating observations into proper target data sets, the structure of the algorithm made it difficult to distinguish these sets from the others created by the algorithm. This is a standard and much discussed problem in cluster analysis. Tracor was able to develop a three stage attack on the problem which has proven to be quite successful. The first step is to use Ling's isolation index to select clusters which are well isolated from the rest of the data. Next, the clusters are checked for randomness and only those that are distinctly nonrandom are selected. Lastly, a qualitative regression procedure is used to check clusters with subsets for internal consistency. Only those clusters which are isolated, nonrandom, and internally consistent are selected as target data sets.

Intersensor Correlation-Initial Guess Procedure and Constraint Rules. To correlate data from more than one sensor, a crossed-bearing, crossed-frequency, batch type algorithm is used to provide initial state vector estimates from the data contained in the various two-sensor and three-sensor cluster combinations. These estimates are then subjected to a series of physical and statistical based tests to determine their validity. The physical tests include checks on reasonableness of position and velocity estimates. The statistical tests use qualitative regression procedures to test the two sensor estimates for compatability, and Gallant's non-linear regression test to compare a three sensor estimate

and its associated two sensor estimates for consistency. Intersensor cluster combinations which pass all tests are then considered as potential target tracks.

Scenario Construction - Integer Programming. Once the potential target track set has been delineated, all that remains is to construct the set of tracks which provide the most likely scenario description. To solve this problem, Tracor has used a 0-1 integer programming set partitioning procedure to sift through the large number of potential scenarios to select the most likely one. To devise a cost function, each track is fed into Tracor's hybrid tracking algorithm and the value of the likelihood function for the associated measurement model residual stream is computed. The likelihood function for a scenario is based on the likelihood function associated with each track, and finding the optimal scenario is equivalent to maximizing the likelihood function over the set of all possible scenarios.

The MTTA was tested on four scenarios of varying difficulty; three of the scenarios contained three targets while one of the scenarios contained only two targets. For three of the scenarios, all targets were detected, sorted, and good track estimates were generated. In the remaining scenario, two of the three targets were detected and tracked, but the third target was only partially detected and no track was generated. The following conclusions were drawn from this study:

1) Observation geometry was the single most important variable for successful detection and tracking of the targets by the MTTA,

2) Strong signal-to-noise ratio is important, but not crucial to the successful operation of the MTTA, and

3) Execution time for the MTTA varies from scenario to scenario based on the complexity of the trajectories. All scenarios were of six minute duration, and the MTTA took from seven to nine minutes to run through them. Minimal parallel processing could turn the MTTA into a faster than real time batch processing scheme.

The recommendations for future work fall into two categories -- refinement of clustering and extraction procedures and MTTA extensions. Under the first category, the following topics were identified as algorithm deficiencies that need to be investigated:

1) Examination of alternative clustering approaches (CLASSY, Anderberg's overlapping algorithm).

2) Using quadratic terms in the cluster extraction regression to model CPA.

3) Addition of a clustering attribute based on power or SNR to help sort targets.

In the second category, all multi-target schemes must deal with the question of whether a measurement belongs to currently existing tracks or is it part of a track that has yet to be initialized. Tracor's proposed implementation of a complete

multi-target tracking algorithm would address this question as soon as a data point was acquired. If it was determined that the data point was not part of a current track, then it would be put into a data pool and at designated times the MTTA would be activated to search for and initialize new tracks. The extension to the MTTA would consist of the decision rules required to ask whether a point belongs to a track or not and then the procedures required to update a track. An additional task would be to investigate whether smoothing of raw data would help both in tracking and updating.

In summary, Tracor has developed a self initializing multi-target tracking algorithm that has performed well with simulated DIFAR data. It runs in nearly real time and requires less than 64 K words of memory. Although in this study the MTTA used Tracor's Hybrid Tracking Algorithm, any single target tracking algorithm could be used to generate scores for each potential track. What has been created is a superstructure which sorts and makes decisions independently of the particular target tracker used. Planned refinements and extensions to the algorithm would give it the ability to run for long periods of time providing a real-time picture of the current tracking scenario. It is felt that the MTTA represents a promising step towards a fully automated DIFAR based, multi-target tracking algorithm.

1.0 INTRODUCTION

For many years, the U.S. Navy has been concerned with the problems of detecting, classifying, and tracking submerged targets encountered in anti-submarine warfare (ASW) engagements. One of the problems which has been extensively investigated at Tracor is underwater target tracking with data gathered by deployed, passive sonobuoy detection systems. Past efforts have concentrated on the problem of tracking a single target. In recent years, however, greater emphasis has been placed on developing techniques which can use passive data to track multiple targets simultaneously. When no a priori information is available concerning target numbers or trajectories, the multiple target tracking problem becomes very difficult to solve. This report contains the findings from the past year of Tracor's efforts to address the problem of tracking multiple targets with passive data when no a priori information is available. Under the current contract a multi-target tracking algorithm was developed which performed very well on simulated multi-target DIFAR data. The algorithm was able to properly sort data at the sensor level, correlate data across sensors and reconstruct the tracks under consideration. It is felt that a positive step has been taken in creating an effective DIFAR based multi-target tracking algorithm.

1.1 Summary of Past Work

In the past, Tracor has studied both the single and the multiple target tracking problems. Initial emphasis was placed on developing a quick and accurate single target tracking algorithm. This work led first to Tracor's development of a batch-type tracker known as the Maximum Likelihood Estimator (MLE, Reference 1). The MLE was designed

to initialize tracking solutions when no a priori information was provided and to continue tracking a single target until all the data have been processed. However, due to the batch nature of the tracker, the MLE was forced to iterate through four different motion models to account for possible target maneuvers and to use sophisticated statistical techniques to automatically choose the most appropriate motion model. Unfortunately, this process was found to be both cumbersome and time consuming.

In an effort to develop a quicker and more efficient tracker, Tracor then developed an Extended Kalman Filter (EKF, References 2 and 3) for single target tracking applications. Since the EKF was a sequential tracker, a single, less sophisticated motion model was used with process noise incorporated into the tracker to compensate for any unmodeled trajectory changes. This single motion model substantially reduced the complexity of the tracker and helped to greatly speed up its execution time. Furthermore, if reasonably good a priori information was available for the target's state, the EKF was not only much quicker, but it was at least as accurate as the MLE. Unfortunately, when no a priori state information was available, the EKF frequently failed to converge upon an adequate track of the target's trajectory, and thus could not be counted on to track targets under all conditions.

Since a quick but accurate tracker was desired for single target-tracking applications, Tracor also developed a tracker which utilized the best characteristics of both the MLE and the EKF. This led to the development of Tracor's Hybrid Tracking Algorithm (HTA, Reference 2 and 3). The HTA utilized a batch type starter to initialize the tracker and then, after initialization had been successfully completed,

automatically switched to a sequential tracker to continue updating the target's trajectory. This tracker was found to provide the accuracy needed to initialize a trajectory when no a priori information was available, and to track as quickly and as accurately as the EKF after initialization was completed. The HTA was then chosen as Tracor's final product for single target tracking applications.

After completing the development of the HTA as a single target tracker, Tracor became involved in the development of a multiple target tracking algorithm (MTTA). This algorithm was to be developed to handle the very difficult problem of tracking multiple targets when no a priori information was available concerning target numbers or trajectories. Furthermore, only passive frequency and bearing data (the data types commonly available from deployed sonobuoys) were to be used to track the targets. In the past, several authors have developed multiple target trackers which used a priori information or active data types such as range measurements to initialize the tracker, but very few have examined the more common and much more difficult problem of tracking multiple targets with only passive data and no a priori information.

Last year, (Reference 3) the development of the MTTA was broken up into two phases. The first phase was designed to enhance the HTA's performance so it could be readily modified for further use in the MTTA. As envisioned, the HTA would serve as the core tracking algorithm which would be used to track the various targets present in a given scenario.

Once the improvements to the HTA were completed, an attack on the second, more difficult, problem of data sorting was begun. For the multiple target tracking problem, some means was needed to sort the data received from individual sensors into sets of individual target data. Without any a priori information concerning the targets that were present, none of the traditional "gating" or "nearest neighborhood" approaches could be used to sort the data. After studying frequency versus bearing time histories of the data, it was decided to try to use cluster analysis to sort the multiple target data into single target data sets. After several trials, single-linkage cluster analysis was found to be fairly successful in sorting the data when the time tag, estimated frequency, and the cosine and sine of the bearing estimates were used as object attributes. However, the results of the cluster analysis program were output as tree diagrams or dendrograms which required user interpretation to correctly sort the data. For the cases studied, all the multiple target data were generated by user dictated simulations, so it was known what type of behavior to look for in the data. With this advantage, it was possible to interpret the tree diagram outputs such that individual target data sets could be readily found. Without the benefit of this knowledge, it is doubtful that the tree diagrams could have been used to find the correct clusters of data. From these results it was decided that cluster analysis showed good potential for sorting multiple target data into individual target data sets, but a suitable cluster extraction scheme had to be developed.

1.2 Results from Current Investigation

Tracor has continued its multiple target tracking investigation through the current year. During this time,

Tracor's MTTA has evolved into a large scale program containing five separate modules. A top-down flowchart of these modules, and a brief description of each one is given in Figure 1.1. The following sections summarize the work that has been performed this year in each of the module areas.

1.2.1 Selection of Clustering Algorithm

During this year's investigation three different clustering algorithms were examined: the single-linkage clustering algorithm from CLUSTAR, Ling's generalized (k,r) algorithm, and Moody and Jardine's single-linkage, non-hierarchical overlapping B_k algorithm. Euclidean distances were used in all algorithms to generate the necessary resemblance matrix of dissimilarity coefficients. The overlapping algorithm allows different clusters to contain one, two or several points in common while the clusters maintain their own separate identity. On the other hand, the non-overlapping algorithm requires that the data points that created the overlap in the B_k algorithm be in one cluster or the other or the non-overlapping technique will force the two clusters to merge into one cluster at these points.

Initially, it had been felt that the B_k overlapping algorithm might provide some flexibility that would be useful for solving the data sorting problem. However, in the cases studied, the non-overlapping techniques have been found to be adequate for sorting the data. Furthermore, both the non-overlapping algorithms were found to be substantially faster than the overlapping algorithm. It was therefore decided to remain with the single-linkage, non-overlapping

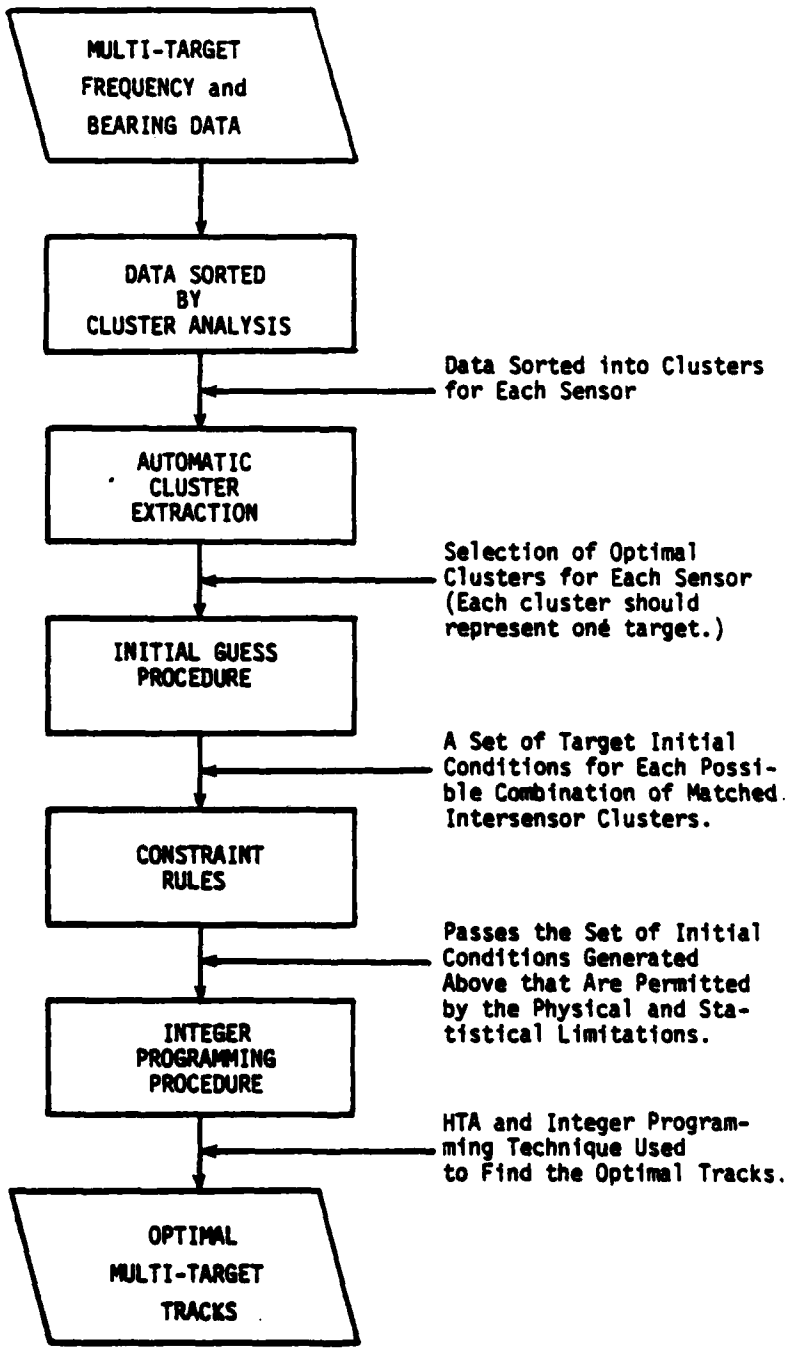


Figure 1.1. Top-Down Flowchart of Tracor's Multi-Target Tracking Algorithm (MTTA)

approach that had been used previously. Of the two remaining algorithms that were under consideration, it was decided to use Ling's (1,r) algorithm rather than his generalized (k,r) algorithm or CLUSTAR's single linkage algorithm because:

1) With k=1, Ling's (k,r) algorithm was equivalent to a single-linkage algorithm and produced the same results as CLUSTAR's algorithm,

2) Its output was more accessible and easier to use than the tree diagrams from CLUSTAR, and

3) Most importantly, Ling has developed a quantitative measure for the (1,r) algorithm that was found to be useful in implementing a procedure for automatic cluster extraction.

1.2.2 Decision Rules for Automatic Cluster Extraction

One of the problems seen last year with the single-linkage clustering algorithm was that it continued to link or chain all the data and the clusters together until ultimately, all the data were chained together in one large, all-inclusive cluster. Naturally, the sorted data for the individual targets were usually found in smaller, subclusters contained as subsets of the large final cluster. In last year's report, available knowledge concerning the simulated scenarios was used to aid in picking the correct smaller clusters from the tree diagram outputs. This year it was sought to automate this cluster extraction process. In doing

so three different decision rules have been established to automatically extract the appropriate clusters. The three rules or tests adopted include:

1) Ling's isolation index is used to identify potentially useful clusters. It identifies clusters which are well isolated from other clusters, implying that the data in the isolated cluster have little similarity with the data in other clusters,

2) A polarization test for data randomness based on work done by Alam and Mitra (Reference 4) which seeks to detect and eliminate noisy clusters, i.e. those clusters containing a high percentage of points corresponding to background noise and not signal measurements, and

3) Lastly, a regression based test is used to make the final identifications of those isolated clusters corresponding to likely target data sets.

1.2.3 Initial Guess Procedure

After the multiple target data from individual sensors have been sorted by cluster analysis into individual target data sets, these data clusters must be properly matched with clusters from other sensors to construct tracks for the individual targets. With only passive frequency and bearing measurements, it is usually impossible to track a target with data from just one sensor. Typically, data from two or three sensors with other observation geometries must be used to track

a given target. For the multi-target case, several clusters from each of the sonobuoys will be available to try to track a target. Ultimately, one must find the right combination of clusters from the different sensors that is needed to track each of the individual targets. After the individual clusters from each sensor have been established, the next step is to get a rough idea of which combination of clusters can potentially be used to build tracks. For this purpose, a crossed-bearing, crossed-frequency least squares procedure has been employed to generate an estimate of the initial target state for each possible combination of matched, intersensor clusters. This procedure has been used previously (Reference 3) to generate a starting point for the HTA's initializer for use in tracking single targets. It has been found that with some slight modifications, this same crossed-bearing, crossed-frequency procedure may be used to provide a cursory look at how well the intersensor combination of data clusters go together to provide estimates of potential initial conditions for the targets. As will be described next, the rough estimates produced here can be examined to see if they meet certain physical and statistical constraints. For those estimates that fail the constraints tests, their corresponding intersensor cluster combination can be eliminated from further consideration. This then helps to reduce the scope of the problem under consideration before it passes on to the next MTTA module.

1.2.4 Physical and Statistical Constraints on the Initial Guess Estimates

As stated, the initial guess estimates must meet certain constraints before they are passed for further consideration by Tracor's MTTA. The first test is to see if

the position estimate falls within the physical limits of the sonobuoy's observation range and if the target's velocity estimate falls within the range of allowable values for target trajectories. For instance, if the range for the day of a given sonobuoy is set at 10,000 meters, one could safely reject any initial guess estimates that would place a target 30,000 meters from the sonobuoy. Similarly, if the initial guess of target velocity falls well outside the known range of values that are possible for target trajectories, one could safely eliminate the intersensor cluster combination that yielded that estimate. For the physical constraints routine used in Tracor's MTTA, the user inputs the maximum sonobuoy detection range and the maximum target speed, and the program rejects all combinations whose estimates are greater than 1.5 times the allowable maximum values. This allows a 50% error to be present in initial guess estimates while still accepting estimates from the upper range of allowed values. The initial guess procedure has been found to be more accurate than this 50% error tolerance, but this tolerance permits a safety cushion which prevents possible intersensor cluster combinations from being rejected prematurely. The second set of constraints applied to the initial guess estimator consists of two statistical tests that measure the consistency of the estimates produced by two or three sensors. The first test uses indicator variables (References 5 and 6) to determine if both sensors in a two sensor initial guess estimate are providing statistically compatible information. Once all two-sensor combinations have been examined, those three-sensor combinations which contain at least two "good" two-sensor combinations are tested. A nonlinear regression test developed by Gallant (Reference 7) is applied to the data to determine if the three-sensor initial guess estimate is equivalent to the

various possible two-sensor estimates. At least two of the three possible two-sensor estimates must be equivalent to the three-sensor estimate for the three-sensor inital guess estimate to pass the test.

1.2.5 Integer Programming and Optimal Track Scenario

Once the initial guess and constraints procedures have selected a set of potential target data sets P, the only remaining problem is to determine the subset of P that provides the "optimum" track scenario. In the MTTA, the optimum scenario is the one which maximize the value of the likelihood function over the set of potential scenarios. Clearly, this is a combinatorial problem that, even for moderate size problems, possesses a large number of possible solutions. In a fashion analagous to Morefield (Reference 8), Tracor has chosen to pose this problem as a 0-1 set partitioning problem using integer programming techniques to sift through the various possible track scenarios to find the optimal set of tracks.

1.2.6 Scenario Simulation Results

With one exception, Tracor's MTTA has successfully tracked all the targets in the multi-target scenarios studied in this investigation. Efforts to automate the cluster extraction process have been largely successful, and the initial guess and integer programming procedures have worked quite well at finding the right set of trajectories needed to track the multiple targets. Four different simulated scenarios have been used in this investigation, and the MTTA successfully tracked all of the targets in three of the four scenarios. For the one scenario that failed, three targets were present but the MTTA only tracked two of them. Careful

examination of the results indicated that for this particular scenario poor observation geometry caused the clustering algorithm to become confused and to merge the data from two targets into one cluster. This confusion was caused when the observational geometries for two different sonobuoys forced two different targets to have nearly identical frequency and bearing measurements during the same time period, and the non-overlapping clustering algorithm could not sort the data into individual target sets. In spite of this problem, the MTTA was still able to track two of the three targets. One of the other scenarios tested the MTTA's capability to track multiple targets with data gathered from a weak signal source that caused random noise to be included in the measurement set. For this scenario, the MTTA effectively sorted the true signals from the noise and was then able to track all the targets found in the scenario. The other two scenarios presented good geometries with strong signals, so the MTTA was able to track all the targets very well. The results from these four scenarios are felt to offer representative examples of how well the MTTA can work in multi-target scenarios when no a priori information is available to initialize the tracker.

1.3 Report Organization

Section 2 of this report includes detailed descriptions of the cluster analysis work performed under this contract. The next section describes the techniques that have been used to automate the cluster extraction process needed to sort multiple target data. In the fourth section, the initial guess procedure is described in detail. Section 5 describes the physical and statistical constraints that were used to eliminate unlikely intersensor cluster combinations. The integer programming procedure used by Tracor's MTTA is

described in the next section, which also includes details of the optimization constraints and the HTA cost function that were used by the optimization algorithm. Section 7 contains a summary of the results obtained for the various simulated scenarios that were examined during this investigation. Finally, Section 8 contains conclusions from this year's research as well as recommendations for future investigations.

2.0 DATA SORTING BY CLUSTER ANALYSIS

Cluster analysis is a field in numerical taxonomy that uses attributes which describe a set of objects to group or sort those objects according to the degree of similarity between them. For the data sorting problem, measurements from an individual target should exhibit strong similarities in their frequency and bearing estimates whereas measurements from different targets should exhibit strong dissimilarities between them. Last year's research established that single-linkage cluster analysis showed good potential for sorting multiple target data into single target data sets (Reference 3). It was shown that frequency versus bearing time histories of the measurements produced a rough, chain-like curve for each target when the data were plotted in three dimensions. Because of the chain-like behavior found in the data, single-linkage clustering algorithms were chosen because they are known to chain data together in forming clusters. Last years results also showed that Euclidean distances between object attributes were the most useful for generating the resemblance matrix of dissimilarity coefficients needed to cluster the data. This section contains the results of current studies which sought to find ways to improve the usefulness of the single-linkage clustering concept.

2.1 Description of Multi-Target Scenario Data

Before evaluating any clustering algorithm's performance, some multi-target data had to be obtained to perform the data sorting tests. For this study, simulated multiple target data were created because no real data were readily available. Frequency and bearing measurements for each target were output at fixed time intervals, and these data were

then merged by time tags to simulate a multiple target data set. This procedure was performed for each of the three sonobuoys found in a scenario, so that at the conclusion of the simulation, a separate multi-target data set was generated for each sonobuoy. The method used to generate this multiple target data was identical to the method used in last year's report to generate the multiple-linetracker data (Reference 3).

2.2 Moody and Jardine's B_k Overlapping Single Linkage Clustering Algorithm

Three cluster analysis algorithms were evaluated during the current investigation. The first, Moody and Jardine's B_k algorithm (Reference 9), is classified as an overlapping single-linkage algorithm. Overlapping means that two clusters share some points by overlapping their boundaries at these points, but they also maintain separate boundaries away from these points. An illustration of two overlapping clusters is presented in Figure 2.1.

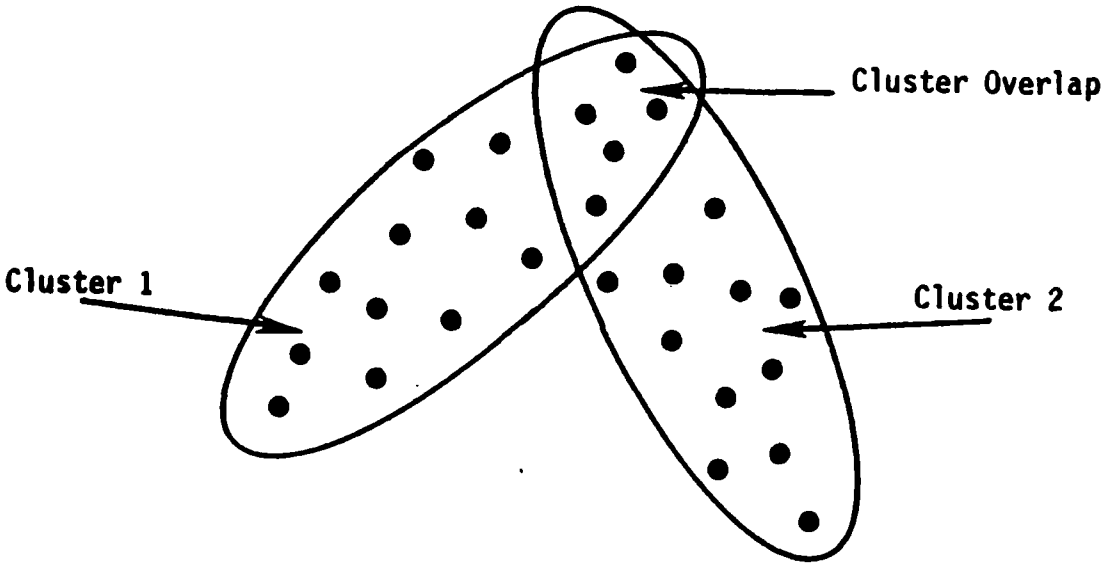


Figure 2.1 - Example of two-overlapping clusters.

Note how the boundaries for clusters 1 and 2 overlap at the apex of the data points in Figure 2.1; they are still classified as separate clusters, but share five points in common. For non-overlapping techniques, the five ambiguous data points in question may have been in either cluster 1 or cluster 2, they may have been grouped in a cluster by themselves, or they may have served as a link to join all the data into one new cluster. However, the points in question could not be contained in two separate clusters as was illustrated in Figure 2.1.

It was originally felt when the B_k algorithm was programmed that such an overlapping scheme might be useful for cases when a non-overlapping clustering algorithm could not clearly differentiate the data from two different targets. Certain geometries are possible where a given sensor may not be able to clearly spot the differences between frequency and bearing estimates for two different targets, and it was feared that a non-overlapping algorithm would chain the two different data sets into one cluster at the point of intersection between the two data sets. Clearly if cluster analysis was to sort multiple target data into clusters for each separate target, one would not like the two data sets to be combined into one cluster and then try to use this cluster as data from a single target. It was hoped that this non-overlapping technique would be useful for preventing these ambiguities from confusing the cluster analysis program. However, after closely examining the performance of the B_k algorithm, it was decided to continue usage of non-overlapping, single linkage clustering algorithms for the following reasons:

1) The B_k algorithm was found to be much slower computationally than the non-overlapping algorithms,

2) It was felt that if sufficient automatic cluster extraction rules could be developed for a non-overlapping algorithm, then data from the two clusters could be sorted before the data sets merged together. This would prevent the need for an overlapping algorithm, and

3) Finally, it was felt that automatic cluster extraction rules could be perfected for at least one of the clustering algorithms. Since Ling had already looked at some aspects of this problem for his (1,r) algorithm, it was decided to expand upon this work and concentrate on using the (1,r) algorithm to automate the cluster extraction process.

2.3 Selection of Ling's (1,r) Clustering Algorithm

After eliminating Moody and Jardine's B_k overlapping clustering algorithm from further consideration, two non-overlapping, single linkage clustering algorithms were left. One of the candidate algorithms was the CLUSTAR single linkage algorithm (Reference 10) that was used to produce the clustering tree diagrams found in last year's report. This algorithm was found to perform well, but the dendrogram outputs were not considered to be very useful for automatic cluster extractions. Having had the advantage of knowing in advance

what data clusters to look for, it was possible to find the desired clusters in the tree diagram outputs. Without this a priori knowledge, it was doubtful that the correct data clusters could have been correctly picked from the output. Clearly, either CLUSTAR's output had to be modified to make it easier to obtain the clustered points, or another algorithm had to be found which performed as well but whose output could be easier to use in the cluster extraction process.

The second candidate clustering algorithm available was Ling's (k,r) algorithm (References 11 and 12). This algorithm requires at least k points to be within some distance r in similarity before the data points can be grouped together into a cluster. For k = 1, Ling points out that his (1,r) algorithm is identical to a single-linkage clustering algorithm. Since CLUSTAR's single-linkage algorithm had performed so well, Ling's (1,r) algorithm was compared to it to see if the (1,r) algorithm could be used. Indeed, Ling's (1,r) algorithm output the same results, but it had a few advantages which made it preferable to CLUSTAR's algorithm. First, the clustering results output by the algorithm provided a set of data points for each new cluster, not clustering tree diagrams as were seen before. Second, Ling had proposed two different decision making rules that could be used with his (1,r) algorithm to aid in the automatic extraction of isolated clusters. Since we have sought to automate the cluster extraction process, it was hoped that these decision rules would be beneficial in picking the isolated clusters. Furthermore, both algorithms took about the same amount of processing time to sort the data, so neither could be penalized due to their respective computing cost. Since both algorithms produce the same results for about the same cost, it was

decided to use Ling's (1,r) algorithm because the output was easier to use and because some decision making rules were available to aid in automating the cluster extraction process.

2.4 Data Normalization

It should be recalled from last year's report that the raw attribute data for each measurement had to be normalized in some fashion to give each attribute roughly the same range of numerical values before the resemblance matrix could be generated. For instance, 360 second trajectories have been used to generate the multi-target data, so the raw time units vary from 0 to 360. Over such a short time interval and for the center frequency values and geometries used in our simulations, the Doppler shifted frequency measurements typically vary by only 0.1 to 0.2 Hz. The bearings typically change by at most 20° to 30°, so the cosine and sine of the bearing measurements vary by only a few tenths. Unfortunately, this large variation in time values as compared to the change in the frequency and the cosine and sine of the bearing estimates caused the clustering algorithm severe problems when the raw attributes are used. When Euclidean distances were used to generate a resemblance matrix for the data, the large difference in time units dominated the dissimilarity coefficient while differences in the other attributes were virtually ignored. Because of this problem, the raw attribute measurements must somehow be scaled to provide the same order of magnitue of differences between sample measurements for each of the attributes.

Last year, experiments were run to test the five different data standardization techniques available in CLUSTER to find the single one that best fit our problem. The one which was finally picked scaled each of the attributes so that their respective range of values fell between 0 and 1.

Specifically, in last year's report, each of the attributes were scaled by the following transformation:

$$\overline{X}_{ij} = \frac{X_{ij} - \min\limits_{j} (X_{ij})}{\max\limits_{j} (X_{ij}) - \min\limits_{j} (X_{ij})}$$

where:

subscript i refers to the ith sample
subscript j refers to the jth attribute
max (X_{ij}) is the maximum value over all i samples of the jth attribute
min (X_{ij}) is the minimum value over all i samples of the jth attribute.

This normalization process worked adequately for our investigation, but one big problem was encountered with this technique. The maximum and minimum values used for normalization varied with each data set, so no two data sets were normalized in the same fashion. This was felt to cause large fluctuations in the values for the dissimilarity coefficients for the different data sets found in last years clustering tree diagrams. Due at least in part to this variation in the data normalization process, the cutoff threshold for halting the clustering process varied substantially from one data set to another. Because of these problems, a standard data normalization procedure was instituted in this year's investigation. The following procedure has been used to normalize the input attribute data for each measurement set:

1) $\overline{t} = t/120$
2) $\overline{\cos\beta} = 5 * \cos\beta$

3) $\overline{\sin\beta} = 5^* \sin\beta$

4) $\bar{f} = 10^* f$

This procedure yields a consistent technique for scaling all the attributes for every problem. Furthermore, for the scenarios investigated here, this normalization procedure allows each attribute to vary by about two or three units. Tests have shown the data normalization procedure described above works very well, so this technique was used to normalize all data simulated in this study.

3.0 AUTOMATIC CLUSTER EXTRACTION

After Ling's (1,r) algorithm sorts the data into clusters, some way is needed to pick the best clusters out of the set. Last year when the tree diagrams were used, the a priori knowledge of how the data should be properly sorted was used to pick the best set of clusters. However, with Ling's (1,r) algorithm, we have sought to automate the process by which the best clusters are chosen. Thus, we are not forced to rely on user interpretation to find the best results and can make the procedure more usable for real applications. In this report, the process of automating the procedure for picking the best clusters is referred to as automatic cluster extraction. First isolated clusters are identified, and then two tests are employed to cull both noise clusters and internally inconsistent clusters. Following is a detailed discussion of this automatic cluster extraction process.

3.1 Single Linkage Clustering and Cluster Formations

A clustering procedure is said to be hierarchical if, for any given pair of clusters produced by the procedure, either one cluster is a subset of the other or they are disjoint. Thus, a hierarchical clustering procedure produces a nested structure that can be represented by a tree diagram or a dendrogram.

Single-linkage clustering is a hierarchical procedure that is based on the "nearest neighbor" concept. A brief description of single linkage clustering follows. Given a set of objects

$$S_o = \{ O_1, O_2, \ldots, O_n \}$$

and a distance d that is defined for each object pair $\alpha(O_i, O_j)$, the single linkage-clustering procedure first determines the closest pair of objects in the set and then joins them to form a cluster C. By defining the distance between cluster C and any object O_k to be

$$d(C, O_k) = \min \left\{ d(O_i, O_k): \ O_i \in C \right\}$$

and forming a new object set S that consists of C and all points of S_o not contained in C (i.e. $S_1 = \{C\} \cup \{S_o/C\}$), the above process can be repeated using S_1 in place of S_o. If the distance between two clusters is defined to be

$$d(c', c'') = \min \left\{ d(O_i, O_j): \ O_i \in C', O_j \in C'' \right\}$$

then the above procedure can be repeated for sets S_2, S_3, --- until all objects are grouped into one large cluster.

Single linkage clustering is a nonmetric procedure in the following sense. If the original distances $d(O_i, O_j)$ are replaced by their rank orderings (i.e., the smallest is replaced by 1, the next smallest by 2, etc.), and the above algorithm is applied to the set S_o, then the sets S_1, S_2, S_3,... will be unchanged (i.e., the same collection of clusters will be produced). Thus, each distance d on the object set S_o can be replaced by a symmetric nxn matrix, M_{d,s_o}, having integral entries ranging from 1 (denoting the distance between the closest pair of objects) to n(n-1)/2 (denoting the distance between the two objects that are farthest apart). Figure 3.1 illustrates the concepts discussed above. The fact that each pair (S_o, d) can be replaced by the pair (S_o, M_{d,s_o}) results in a great

$$S_o = \{O_1, O_2, O_3\}$$

Object Combinations	Distance	Rank Ordering
$d(O_1, O_2)$	.5	2
$d(O_1, O_3)$	2	3
$d(O_2, O_3)$	.25	1

$M_{d,\ s_o}$

	O_1	O_2	O_3
O_1	0	2	3
O_2	2	0	1
O_3	3	1	0

$S_1 = \{C_1, O_1\}$, $C_1 = \{O_2, O_3\}$

$S_2 = \{C_2\}$, $C_2 = \{O_1, O_2, O_3\}$

Figure 3.1. Clustering Example

simplification since for $S_o = \{0_1, 0_2, \ldots, 0_n\}$, there are only $(n(n-1)/2)!$ distinct possible choices for M_{d,s_o}. This fact will be exploited when measures of cluster significance are considered since the output from a clustering algorithm consists of a large number of clusters, some of which are contained in, and some of which contain, the target data sets sought.

3.2 The Isolation Index

When performing cluster analysis on a set of objects, each cluster is formed at a particular clustering level. For single linkage clustering, these levels can be taken to be integers beginning with 1. Formally, the single linkage clustering level r of a cluster C is defined by

$$r = \min \{k: \ C \in S_k\}.$$

Some authors refer to r as the "time of birth" of cluster C. If the sets S_1, S_2, S_3, ... are called clustering sets, then the clustering level of a cluster C is just the index of the clustering set in which C first appears. The index k of set S_k will be referred to as the clustering level k.

After a cluster has first formed at level r, it may remain unaltered for many future clustering levels. However, when C is not equal to the entire set of objects, it follows that eventually C will be augmented to form a new cluster C'. A cluster that remains unchanged for a number of levels is said to be well isolated. The following definition quantifies an isolation index measure possessed by cluster C:

$$i(C) = r' - r,$$

where r' is the clustering level of the augmented cluster C' that is described above. The clustering level r' can be thought of as the "time of death" of cluster C. In Figure 3.1, cluster C, is "born" at level 1 and "dies" at level 2. Therefore, $i(C_1) = 2 - 1 = 1$.

3.3 Isolation Index Distribution - Survival Function

Recall that the matrix M_{d,S_o} determines the clusters that are produced by a single-linkage clustering procedure. By choosing a null probability model on M_{d,S_o}, it is possible to use probabalistic techniques to assess the significance of single-linkage clusters. The simplest and most conservative model to impose on the space of all possible random matrices M_{d,s_o} is to assume that each matrix is equiprobable. Under this assumption, the mathematics are tractable and the statistical theory resulting from the use of this model can be interpreted as establishing bounds for other models (i.e., a cluster that is judged to be statistically insignificant according to this model is unlikely to be significant under any other model). Under the equiprobable assumption, the isolation index of a cluster of size 1 becomes a random quantity whose distribution can be calculated. Specifically, if C is a cluster of size j with clustering level r, then

$$\text{Prob}\,[i(c) = s] = \frac{\frac{1}{s}\binom{j(n-j)}{1}\binom{N - r - j(n-j)}{s-1}}{\binom{N-r}{s}}$$

where $N = \binom{n}{2}$. The probability of the event $\{i(C) \geqq s\}$ being small implies that the chance of a cluster with this isolation index appearing simply by accident or at random is small.

Thus, a good assumption is that these data "belong together," i.e. came from the same target. Therefore, the survival function is used to help select the set of candidate clusters; a probability threshold is set and those clusters with survival function values falling below the threshold are passed on for further testing.

3.4 Polarization Test and Cluster Extraction

Once the isolation index has been used to identify clusters of data that possess greater than expected similarities, these clusters must then be analyzed to determine their randomness and internal consistency. Two primary reasons for doing this take advantage of the way in which single linkage clustering joins data together (see section 3.1):

1) Random noise can appear as an isolated cluster, either when targets are present or when just random noise is clustered, and

2) As larger and larger clusters are formed from the data, isolated clusters can become subsets of larger isolated clusters. Eventually, there may be several isolated clusters contained in one large isolated cluster, and they may or may not be observations for the same target.

This section discusses the test for randomness employed by the MTTA. To motivate the selection of this test, imagine a cluster of data triples (time, frequency, and

bearing) projected onto the frequency-bearing plane, (Figure 3.2 provides several examples). Then,

1) Figure 3.2.a represents random noise, frequency and bearing estimates are scattered randomly about,

2) Figure 3.2.b represents data from a target showing significant bearing change but little Doppler change,

3) Figure 3.2.c represents data for a target that shows little bearing change but substantial frequency change, possibly approaching CPA with the sonobuoy, and

4) Figure 3.2.d represents data from a target with little bearing change or frequency change. This is a typical plot for a target in which a large amount of data has been gathered over a short time interval.

If a grid is imposed on the plane, then the number of observations occurring in a cell can be tallied and the number of observations occurring in a particular row or column can also be tallied. Then for clusters of random noise (as in Figure 3.2.a), the row totals should be roughly equal and the column totals should all be roughly equal while for target clusters, such as those displayed in Figures 3.2.b and c, either the row or column totals (or both) will display very distinct departures from equality. If a multinomial distribution is assumed for these row and column totals, then Alam and Mitra (Reference 4) have devised a test to determine

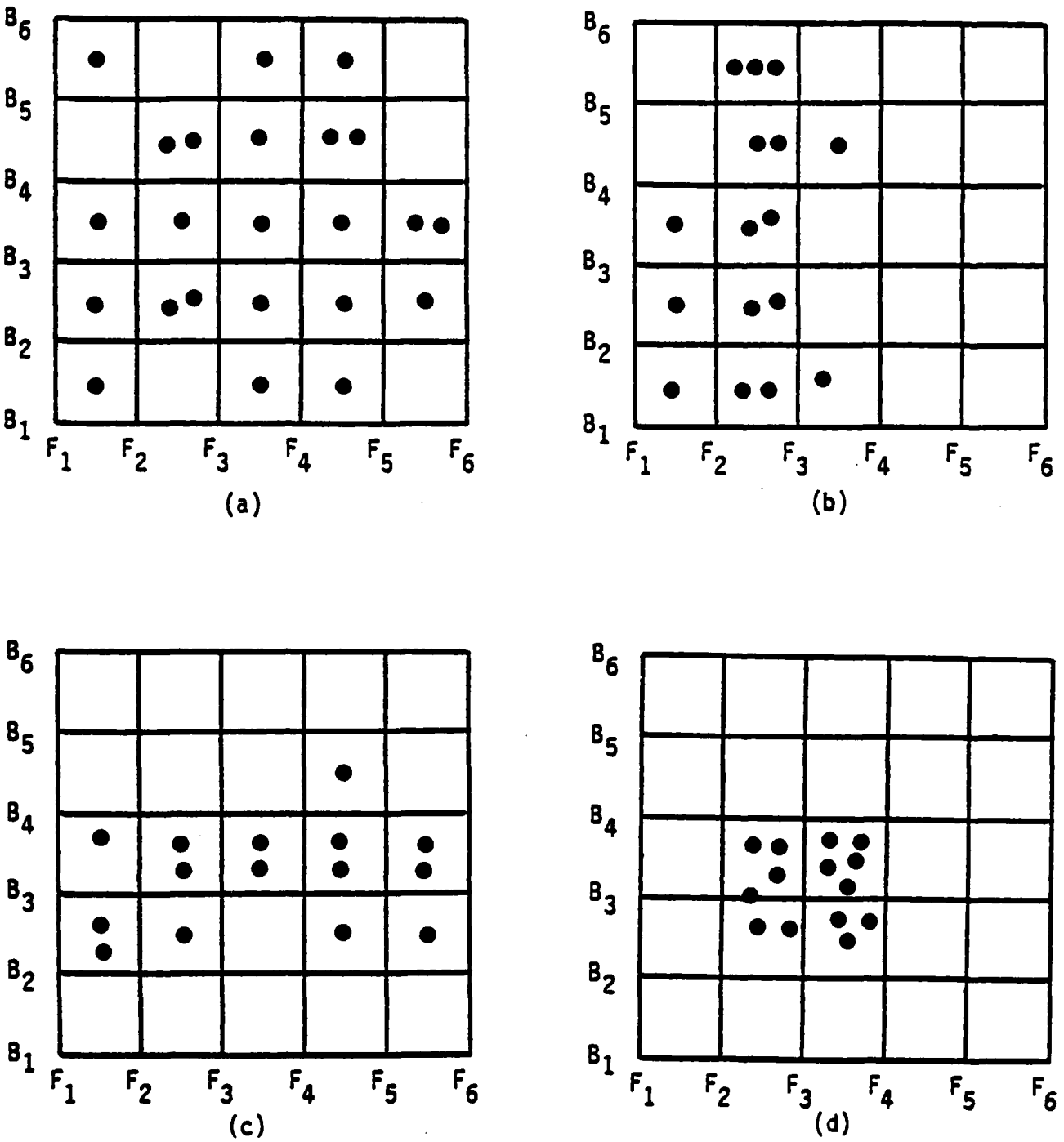


Figure 3.2. Bearing/Frequency Plot Examples

if the mass of the points is spread out evenly over the row and column totals or if it is concentrated, or polarized, in just a few of the rows and columns. A brief explanation of the test follows.

A partial ordering of the vectors $P = (p_1,\ldots,p_n)$ is defined by the relation:

$$\vec{P} > \vec{P'} \text{ if } \sum_{i=1}^{j} p_{(i)} \geq \sum_{i=1}^{j} p'_{(i)} \; ; \; j = 1, 2, \ldots, k$$

where $p_{(i)}$ is the ith largest value among $p_1,\ldots,p_k$. Then, a multinomial distribution with probability vector $\vec{P}$ is said to be less polarized than the multinomial distribution with probability vector $\vec{P'}$ if $\vec{P} * \vec{P'}$. Note that

$$\left(\frac{1}{k}, \frac{1}{k}, \ldots, \frac{1}{k}\right) < \left(\frac{1}{k-1}, \frac{1}{k-1}, \ldots, \frac{1}{k-1}, 0\right) < (1, 0, 0, \ldots, 0).$$

Let $\vec{R} = (r_1,\ldots,r_k)$, $\Sigma r_i = n$ be the k row totals associated with a particular cluster and consider the following test:

$$H_o: \vec{R} = \vec{R'} \text{ against the alternative}$$

$$H_1: \vec{R} > \vec{R'}$$

where $\vec{R'}$ is the equally likely case. Alam and Mitra consider the test statistic

$$T(\vec{R}) = \quad r_i^2/n$$

and show that H_o is rejected for large values of T. The critical region of the test is

$$\{\vec{R} \mid T(\vec{R}) > C\} \text{ where}$$
$$P\{T(\vec{R}) > C \mid \vec{R} = \vec{R}'\} = \alpha.$$

For $\vec{R}' = (1/k, 1/k, \ldots, 1/k)$ Alam and Mitra show that

$$T^* = kT(\vec{R}) - n$$

is asymptotically distributed as $\chi^2 k-1$. A similar test can also be conducted for the column totals.

In applying this test, the grid size selected was 25 by 25, with bearing running from 0 to 2π radians and frequency from 148.8 to 151.2 Hz. In their paper, Alam and Mitra briefly discuss the convergence properties of the test and show that n must be fairly large before the $\chi^2 k-1$ approximation is very good. This was observed in the current study, for large clusters ($n \geq 60$) testing at the .995 percentage point of the χ^2_{24} distribution was very effective at distinguishing noise clusters from data clusters. However, for clusters of size less than forty, the test still discriminated very well but the χ^2_{24} distribution did not provide a good critical value. Clusters composed mostly of observational data had high values of T* while clusters composed mostly of noise had much lower values of T*. However, the values of T* found for noise clusters were higher than the cut-off points supplied by the χ^2_{24} distribution.

At this point there were two basic choices:

1) Compute the conditional distribution function CDF of $T(\vec{R})$ directly using the recursion formula presented by Alam and Mitra, or

2) Determine if some simple, "rule-of-thumb" existed which would allow reasonable decisions to be made.

The second alternative was chosen because:

1) Implementation of the Alam and Mitra CDF algorithm would have been a formidable programming task, particularly for $n > 5$,

2) Including the calculation as a subroutine would have imposed a substantial computation penalty, and

3) A simple and reasonable "rule-of-thumb" value was readily available.

Primary considerations involved in selecting a cut-off value were:

1) It should not reject H_o for very small clusters ($n \leq 5$) no matter how concentrated its members are,

2) It should reject H_o for small clusters ($5 \leq n \leq 15$) only if most of their members are concentrated in one or two cells, and

3) It should reject H_o for moderate sized clusters ($15 < N < 40$) only if a significant number (40% or better) of their members are concentrated in one or two cells.

Figure 3.3 contains an example of several graphs that were constructed for various values of T*. Simulation results had shown that for small time periods (≤5 minutes), clustered target bearing and frequency data would usually be confined to two or three cells, with some noise points possibly scattered about in other cells. Thus, curves were constructed for T* as a function of x_1, x_2, x_3 where:

1) x_i was the number of points in cell i,
2) x_1 ranged from 0 to n,
3) x_2 ranged from 0 to $n-x_1$,
4) x_3 ranged from 0 to $n-x_1-x_2$, and
5) the remaining $n-x_1-x_2-x_3$ points were considered to be placed one to a cell.

Clearly, the value of T* is invariant to which cells contain the points, so that the choice of x_1, x_2, x_3 is not important.

The dashed lines of Figure 3.3 indicate the two cutoff points, the leftmost line corresponds to the .995

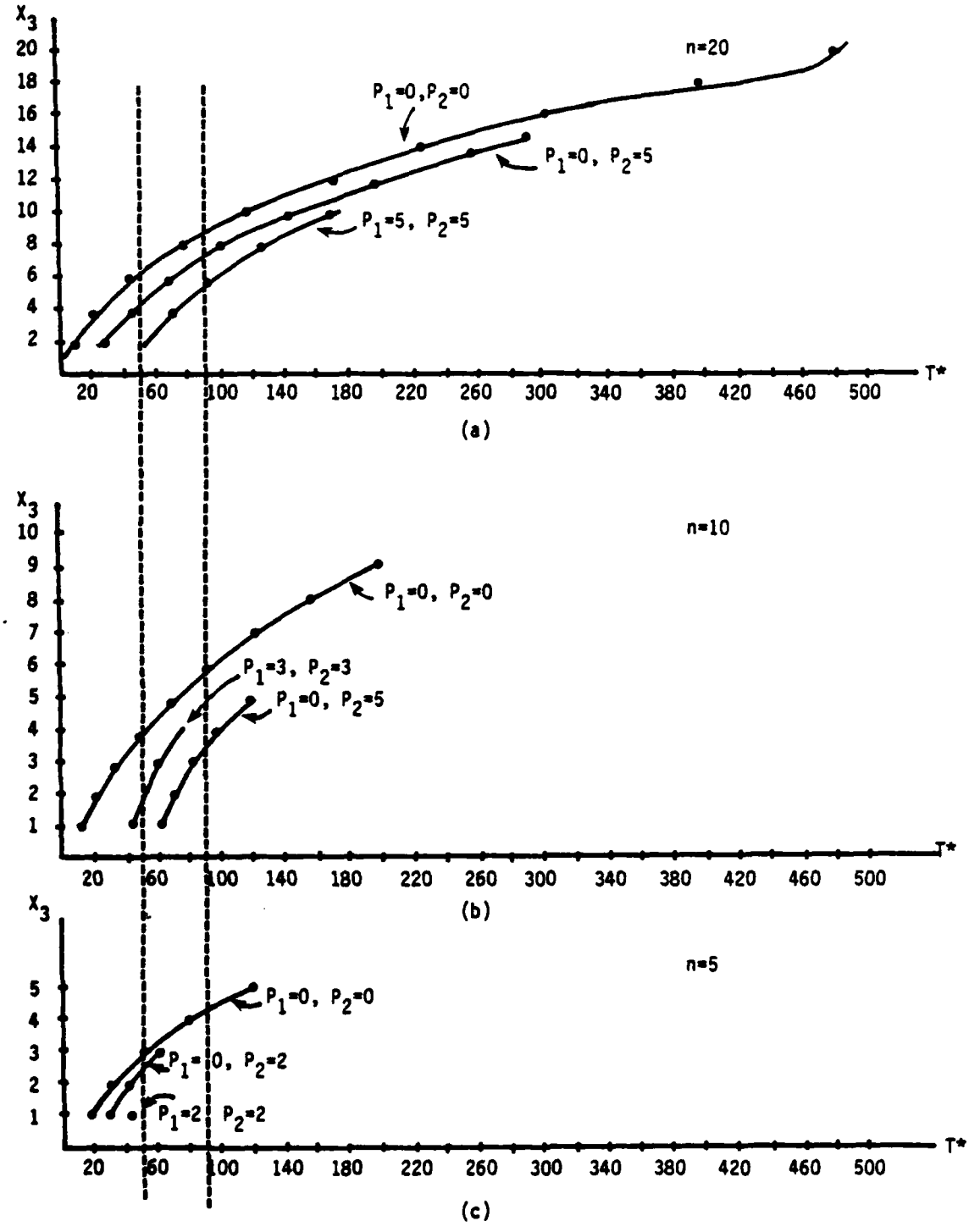


Figure 3.3. Plots to Determine Polarization Test Cut off Point

percentage point of the χ^2_{24} distribution and the rightmost line corresponds to twice this number. Keeping in mind that the hypothesis being tested is

$$H_o: \vec{R} = \vec{R}' = (1/k, \ldots, 1/k) \text{ versus}$$
$$H_1: \vec{R} > \vec{R}',$$

it is clear that the rule-of-thumb value requires substantially higher cut-off values than does the approximation value. Additionally, the rule-of-thumb value will not reject H_o for $n < 5$ and requires almost total concentration in a single cell for n near 5. For $n \geq 10$, concentrations in the 40% to 60% range are required for single cell polarization and in the 80% - 90% range for two and three cell polarizations. The comparable χ^2_{24} approximation values are at least 20% - 30% below this. Using the rightmost line as the cut-off has worked quite well in this study. In order for a cluster to be declared "random" it must accept H_o for both the row and column totals.

3.5 Internal Consistency and Cluster Extraction

This section discusses the final stage in the cluster extraction process, the test for internal consistency. Figure 3.4 illustrates the logic flow for the procedure.

At this point in the extraction process, the clusters under consideration are assumed to be well-isolated, non-random groups of data. What remains is to examine for consistency each cluster that contains two or more isolated clusters as subsets. Basically, the idea is to determine whether the information contained in all the isolated subclusters is compatible or not.

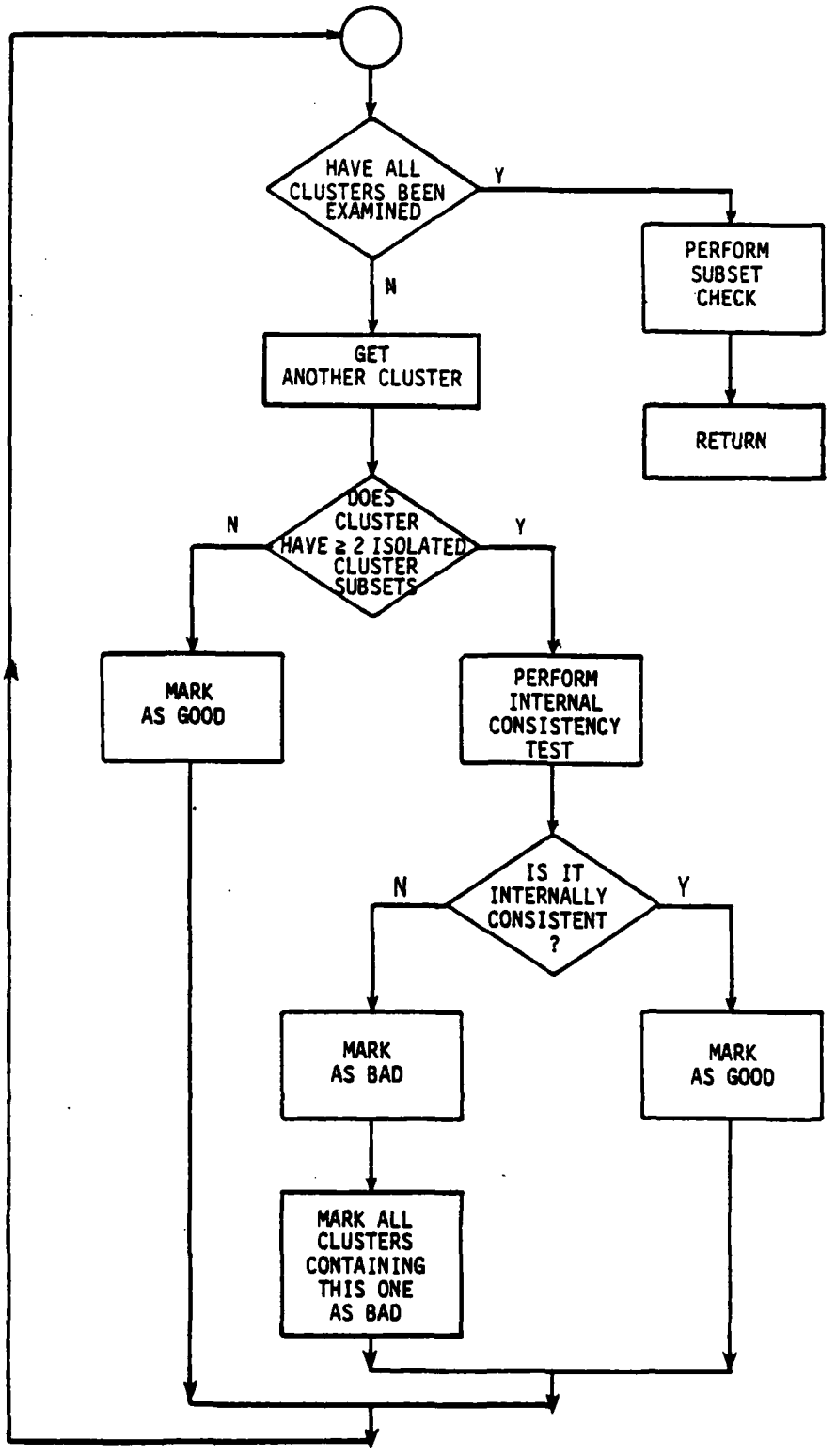


Figure 3.4. Flowchart of the Internal Consistency Test

3.5.1 Regression and Internal Consistency

One of the primary difficulties in finding an internal consistency test for isolated clusters which contain other isolated clusters as subsets is that the frequency and bearing estimates are functions of time within these clusters. Typically, these isolated subclusters correspond to time segments of observations for one or more targets; they start as small groups of two or three points and then chain outward until the cluster under consideration is formed. Under these circumstances, usual measures of similarity such as means, ranges, modes, etc., can be very misleading. For a particular target, the average frequency, bearing sine, or bearing cosine from the first part of an observation period can be quite different from the average computed from the last part of the observation period. Based on these considerations, regression techniques were used to develop the internal consistency test.

When frequency, bearing sine, and bearing cosine for a particular target are plotted against time for short intervals (less than six minutes), they are nearly linear functions and can be approximated fairly well by straight lines. Measurements from two different and distinguishable targets define different straight lines, and this forms the basis for the use of single-linkage clustering techniques. However, as was shown in Section 3.1, the single-linkage algorithm creates ever larger clusters by joining points or clusters to already existing clusters. If clusters are joined from two different and distinguishable targets, then the problem of internal consistency can be cast as a regression problem to determine whether a set of a data defines one or two regression lines. The lines could differ by a change in intercept, a change in slope, or by both.

Neter and Wasserman (Reference 6) give a procedure for doing this. The basic formulation of the model is:

1) Let Y_i be the measurement under examination (frequency, bearing sine, or bearing cosine) and let T_i be the time of the measurement,

2) Let I_i be an indicator variable attached to Y_i such that $I_i = 0$ if Y_i comes from the first cluster and $I_i = 1$ if Y_i comes from the second cluster, and

3) The model becomes

$$Y_i = B_o + B_1 T_i + B_2 I_i + B_3 T_i * I_i + \epsilon_i$$

and the response (regression function) is

$$E(Y) = B_o + B_1 T + B_2 I + B_3 T * I.$$

To understand the function of the indicator variable in this model, note that for measurements from the first cluster (I=0),

$$\begin{aligned} E(Y) &= B_o + B_1 T + B_2 (0) + B_3 (0) \\ &= B_o + B_1 T \end{aligned}$$

and for measurements from the second cluster (I=1),

$$\begin{aligned} E(Y) &= B_o + B_1 T + B_2 (1) + B_3 (1)(T) \\ & (B_o + B_2) + (B_1 + B_3) T. \end{aligned}$$

Thus, B_2 measures the difference in intercept estimates between the two lines and B_3 measures the differences in slope estimates between the two lines. Therefore, the test for equality of the two regression lines corresponds to a test of the hypothesis

$$H_o: \; B_2 = B_3 = 0 \text{ versus}$$
$$H_1: \; B_2 \text{ or } B_3 \neq 0.$$

The test statistic is

$$\frac{SS\ (B_2,\ B_3 | B_o,\ B_1)}{2} \div \frac{SSE}{n_1 + n_2 - 4} = F^*$$

where

1) SS $(B_2, B_3 | B_o, B_1)$ is the reduction in the regression sum of squares due to the inclusion of I, and T*I in the model,

2) SSE is the error sum of squares for the model, and

3) n_1 is the number of objects in cluster 1 and n_2 is number of objects in cluster 2.

The above procedure outlines a method for determining whether a set of measurements coming from two different clusters is internally consistent or not. This is done for each of the three measurements available i.e. frequency, bearing sine, and bearing cosine. A weighted decision scheme is then used based on the three calculated

F-values (F_f, F_{BS}, and F_{BC}), to determine whether the two clusters contain consistent information or not. Considerations used to establish this scheme include:

1) Frequency information should be weighted less than bearing sine and bearing cosine information. This is done because frequency is only measured to the nearest tenth of a Hertz, so a normal Doppler shift of only 0.1 Hz can indicate a spurious inconsistency,

2) The bearing sine (bearing cosine) measurements from two targets may be quite close while the bearing cosine (bearing sine) measurements are substantially different. Thus, only one of the bearing component measurements is needed to indicate inconsistency, and

3) When in doubt, pass a cluster. The result of passing a cluster that should not be passed is an increase in processing time for other areas of the MTTA algorithms. The result of not passing a cluster that should be passed is the possible loss of a target track.

The weighting scheme (to be discussed below) depends on a method developed by Suich and Derringer, (Reference 13) and extended by Ellerton (14) to examine the significance of subsets of the regression variables. Typically, this subset of variables describes some particular characteristic, such as trend or curvature, of the response

system. In our case, the characteristic being examined is the tendency of the data to be best described by two regression lines instead of one.

Let $\hat{Y}_f(\vec{x})$ represent the full regresssion model and let $\hat{Y}_r(\vec{x})$ represent the reduced regression model, that is, the full model minus the variable subset under consideration. The quantity to be tested, γ, is the ratio of the average, squared expected differences in prediction of the two models to the average variance of these predicted differences;

$$\gamma = \frac{\text{average expected squared prediction difference}}{\text{average variance of predicted differences}}.$$

The test statistic is the usual F-ratio, which in this case is,

$$\frac{SS\ (B_2,\ B_3 | B_o,\ B_1)}{2} \div \frac{SSE}{n_1+n_2\ -\ 4}$$

and the hypothesis tested is

$$H_o:\ \gamma = \gamma_o \text{ versus}$$
$$H_1:\ \gamma > \gamma_o.$$

Acceptance of the null hypothesis indicates that either the full model should be reassessed as a predictor or that the error variation in the data is simply too large relative to the variation in predictive power of $\hat{Y}_f(\vec{x})$ and $\hat{Y}_r(\vec{x})$. Essentially, if there are large errors relative to the range covered by the data, then a subset of variables may, under the usual F-test, appear to be significant when in fact they are simply fitting the errors in the data. The purpose of this test is to detect that situation.

In the extraction algorithm γ_o was set equal to 2, and the critical value then became about four times the F-value used to make the standard test:

H_o: $B_2 = B_3 = 0$ versus
H_1: at least one of B_1, $B_2 \neq 0$.

3.5.2 Weighting Scheme

Using the regression analysis just described, the interval consistency test reduces to the following, letting F_u stand for the "usual" regression F critical value and F_E stand for the value generated by applying Ellerton's work, the weighting scheme used is:

1) If $\begin{Bmatrix} F_f^* < F_u \\ F_f^* \geq F_u \end{Bmatrix}$ then $\begin{Bmatrix} W_f = 0 \\ W_f = 1 \end{Bmatrix}$;

2) If $\begin{Bmatrix} F_{BS}^* < F_u \\ F_u < F_{BS}^* < F_E \\ F_{BS}^* \geq F_E \end{Bmatrix}$ then $\begin{Bmatrix} W_{BS} = 0 \\ W_{BS} = 1 \\ W_{BS} = 2 \end{Bmatrix}$;

3) If $\begin{Bmatrix} F_{BC}^* \leq F_u \\ F_u < F_{BC}^* < F_E \\ F_{BC}^* \geq F_E \end{Bmatrix}$ then $\begin{Bmatrix} W_{BC} = 0 \\ W_{BC} = 1 \\ W_{BC} = 2 \end{Bmatrix}$;

4) $W = W_f + W_{BS} + W_{BC}$, and

5) If $W \geq 2$ then reject the hypothesis that the two clusters are consistent.

Basically, this scheme says that frequency variation alone is not enough to reject the idea of consistency, instead it must reject in concert with either bearing sine or cosine. On the other hand, bearing sine or cosine measurements are enough by themselves to reject consistency if the indication of multiple cluster data is extremely strong. Middle of the road rejection values from both bearing cosine and sine measurements are also enough to reject consistency. Because a false rejection of the consistency hypothesis is much more costly than a false acceptance, all critical values were set with $\alpha = .005$.

If a particular cluster possesses exactly two isolated subclusters, then the above procedure is fine as it stands, the cluster will be accepted or rejected based on the final value of W. However, if a given isolated cluster contains more than two isolated clusters as subsets, then another procedure for rejecting consistency is needed. Suppose the cluster in question has k isolated subclusters, then there are $\binom{k}{2}$ cluster pairs to be examined for consistency. If it is assumed that all clusters come from the same track, then k-1 of the potential pairs should link together or show consistency. Thus, for clusters with k isolated subclusters ($k > 2$), the consistency hypothesis will be rejected if fewer than k-1 cluster pairs pass the two cluster consistency test. Experience has shown this to be a fairly liberal criterion.

3.6 Automatic Cluster Extraction Summary

In order to be selected as a target data set a cluster must possess three characteristics:

1) Isolation from the rest of the data,

2) Polarization of observations in a few frequency or bearing bins, and

3) Internal consistency among isolated subclusters.

Cluster isolation is measured using Ling's survival function. Those clusters possessing a survival function value of less than 10^{-4} are considered well isolated and selected for further testing. Once a cluster is considered isolated it is subjected to the Alam and Mitra polarization test to determine if most of its members are concentrated in a few bins or not. Those clusters whose observations are concentrated in a few bins are determined to be nonrandom collections and are passed on to the final test. Clusters remaining at this point are examined for subclusters which happen to be isolated clusters also. Those clusters which contain no isolated subclusters are considered internally consistent and declared target data sets. Clusters containing isolated subclusters are given a regression test for internal consistency. Those that pass the test are passed as target data sets.

All the above tests are conducted using hypothesis tests with alpha values set high to allow marginal groups to pass through and be selected. It is felt that it is better to pick a bad cluster as a potential target data set than reject a cluster that is a target data set.

4.0 INITIAL GUESS PROCEDURE

Once the sonobuoy data have passed through the cluster formation and cluster extraction algorithms, they have been grouped into sets of clusters for each sensor that correspond to potential target data sets. The next step is to solve the intersensor correlation problem which seeks to determine which data clusters from each sensor contain observations on the same target. For this study, simultaneous observations for a target were required from either two or three sensors because, in general, a single sonobuoy cannot provide enough information with which to track a target. For even a small number of sensors and targets, the number of potential combinations can become fairly large. For example, consider a 3 target scenario observed by three sonobuoys and assume that each buoy hears each target perfectly and that the cluster formation and extraction routines perfectly separate the data. There are then three data clusters associated with each sensor, and these can be combined into 27 (i.e., 3*3 + 3*3 + 3*3) potential two-cluster intersensor combinations and 27 (3*3*3) potential three-cluster intersensor combinations.

To help separate unlikely combinations from likely combinations, an initial guess procedure and a set of guess evaluation criteria were developed. This section describes the initial guess procedure itself and the following section discusses the evaluation criteria.

4.1 Crossed-Bearing, Crossed-Frequency Initial Guess Procedure

An initial guess procedure which used crossed-bearing and crossed-frequency information to generate

initial conditions for target tracks was described in last year's report (Reference 3). It was used to generate a reasonable guess of a target's position and velocity. This guess was then passed to the initializers of either Tracor's EKF or HTA so that the tracker could more quickly converge onto an acceptable set of initial conditions for the target. This procedure used the frequency and bearing measurements from two or more sensors in a least squares formulation to estimate the target's initial position and velocity. The estimates from this procedure were found to be reasonably good and greatly improved the speed and the initialization characteristics of Tracor's two trackers.

For a bearing estimate from sensor i, the crossed-bearing equation is:

$$x \sin \beta_i - y \cos \beta_i = x_i \sin \beta_i - y_i \cos \beta_i$$

where

β_i = the bearing estimate for sensor i,
(x_i, y_i) = the position components for sensor i,
(x, y) = the position components for the target.

Similar crossed-bearing equations can also be obtained for any other observing sensors. The crossed-frequency formulation uses the bearing and frequency estimates from a given sensor to generate target velocity estimates. The actual crossed-frequency equation for buoy i is:

$$\dot{x} \cos \beta_i + \dot{y} \sin \beta_i = c \left[\frac{f_o}{f_i} \left(1 + \frac{\dot{x}_i \cos \beta_i + \dot{y}_i \sin \beta_i}{c} \right) - 1 \right]$$

where

β_i = Bearing estimate from buoy i,
f_i = Doppler-shifted frequency estimate from buoy i,
f_o = Unshifted center frequency transmitted by the target,
c = Speed of sound in the water,
$(\dot{x},\dot{y})$ = Velocity components for the target,
$(\dot{x}_i,\dot{y}_i)$ = Velocity components for the buoy.

Once again, similar crossed-frequency measurement equations can be generated for other sensors that observe the target.

A motion model is needed to correlate the target's position and velocity estimates and to provide a means of mapping all measurements back to an initial epoch so that a batch-type, least squares estimate can be generated for the target's initial position and velocity. Since only a small data stream which covers a short time span is used in the initial guess procedure, a linear motion model was selected to describe the target's trajectory. The linear motion model used is:

$$x = x_o + \dot{x}_o \Delta t$$
$$y = y_o + \dot{y}_o \Delta t .$$

Following Bard (Reference 15), the crossed-bearing and crossed-frequency equations can be combined with the motion model in a two-equation least squares model to estimate (x_o, y_o, $\dot{x}_o$, $\dot{y}_o$).

4.2 Measurement Weighting Equations

Last year, least squares estimates for the target's initial state vector were generated by weighting all the measurements equally. This resulted in good position estimates but relatively poor velocity estimates. It was felt that weighting the measurements might improve the target's velocity estimates.

Associated with each frequency and bearing measurement is a standard deviation value which describes statistically the accuracy of the measurement. These standard deviations are computed as functions of the estimated signal-to-noise ratio for the received signal. Typically, the measurement standard deviations are used to compute a measurement weight which enables the data to be processed by a weighted least-squares tracking algorithm. With the crossed-bearing and crossed-frequency initial guess algorithm, the measurement equations are functions of the frequency and bearing estimates. Some method has to be used which computes standard deviations of the crossed-bearing and the crossed-frequency measurements as functions of the standard deviations computed for the frequency and bearing estimates. Following Young (Reference 16), a first order Taylor series approximation is used to compute the standard deviation for the modified measurement equation. If we use G to represent either the crossed-bearing or the crossed-frequency measurement function and f and β to represent the frequency and bearing estimates, respectively, the following equation is used to compute the measurement weights:

$$\sigma_G^2 = \left(\frac{\partial G}{\partial f}\right)^2 \sigma_f^2 + \left(\frac{\partial G}{\partial \beta}\right)^2 \sigma_\beta^2 .$$

In the above formulation, the assumption has been made that the buoy positions and their velocities are perfectly known, so no buoy position or velocity errors are allowed to propagate through this standard deviation equation. Specifically for the bearing equation, the measurement equation, G, is:

$$G = x_i \sin \beta_i - y_i \cos \beta_i .$$

Since no buoy position errors are encountered and since the frequency estimate does not contribute to this measurement equation, the crossed-bearing standard deviation becomes a function of the bearing estimate only

$$\sigma_G^2 = \left(\frac{\partial G}{\partial \beta}\right)^2 \sigma_\beta^2$$

$$\sigma_G^2 = (x_i \cos \beta_i + y_i \sin \beta_i)^2 \sigma_{\beta_i}^2 .$$

Conversely, the crossed-frequency measurement equation is a function of both the bearing and frequency estimates

$$G = c \left[\frac{f_o}{f_i} \left(1 + \frac{\dot{x}_i \cos \beta_i + \dot{y}_i \sin \beta_i}{c} \right) - 1 \right] .$$

To obtain the crossed-frequency standard deviation, it is assumed that no errors are encountered in obtaining buoy state values so that they do not contribute to the sources for the crossed frequency measurements:

c - speed of sound in the water,
$(\dot{x}_i, \dot{y}_i)$ - velocity components of buoy i,
f_o - unshifted center frequency value for the transmitted target signal.

f_i - Doppler shifted frequency received by the sonobuoy.

The crossed-frequency equation for sigma then reduces to:

$$\sigma_G^2 = \left(\frac{\partial G}{\partial \beta}\right) \sigma_\beta^2 + \left(\frac{\partial G}{\partial f_i}\right)^2 \sigma_{f_i}^2$$

$$\sigma_G^2 = \left[\frac{f_o}{f_i}\left(-\dot{x}_i \sin \beta_i + \dot{y}_i \cos \beta_i\right)\right]^2 \sigma_{\beta_i}^2 + \left[\frac{-cf_o\left(1 + \frac{\dot{x}_i \cos \beta_i + \dot{y}_i \sin \beta_i}{c}\right)}{f_i^2}\right]^2 \sigma_{f_i}^2 .$$

Using the weights generated by these equations in the estimation procedure has greatly improved the velocity estimates without affecting the position estimates from the MTTA's IG algorithm. The IG algorithm now in use in the MTTA, therefore, is a weighted least-squares crossed-bearing, crossed-frequency algorithm.

5.0 PHYSICAL AND STATISTICAL CONSTRAINTS ON THE INITIAL GUESS ESTIMATES

After initial guess estimates are generated for each of the possible intersensor cluster combinations, these estimates must be examined to determine which ones provide initial conditions which are plausible in real world encounters. As stated previously, there are twenty-seven possible two-cluster intersensor combinations, and twenty-seven, three-cluster intersensor combinations for the three-target, three-sensor scenario shown in Figure 7.1. Figure 5.1 shows all the possible bearing line-of-sight intersections that would result from combining bearings from two sensors in the crossed-bearings procedure. Notice the large number of intersections that result from this simple example. Although it is not as easy to illustrate, a large number of possible intersections also result from using the crossed-bearing procedure for all possible three-cluster, intersensor combinations. This section examines ways to use physical and statistical constraints to eliminate the unlikely combinations produced by the initial guess procedure.

5.1 Physical Constraints Test

The first step in eliminating implausible initial guess values is to examine the estimates in terms of sonobuoy detection limits and maximum submerged target performance levels. These limits are referred to as the physical constraints on the problem. These constraints have been left as user defined inputs so that they may be varied for the different classes of sonobuoys or the different types of targets that might be encountered in real engagements.

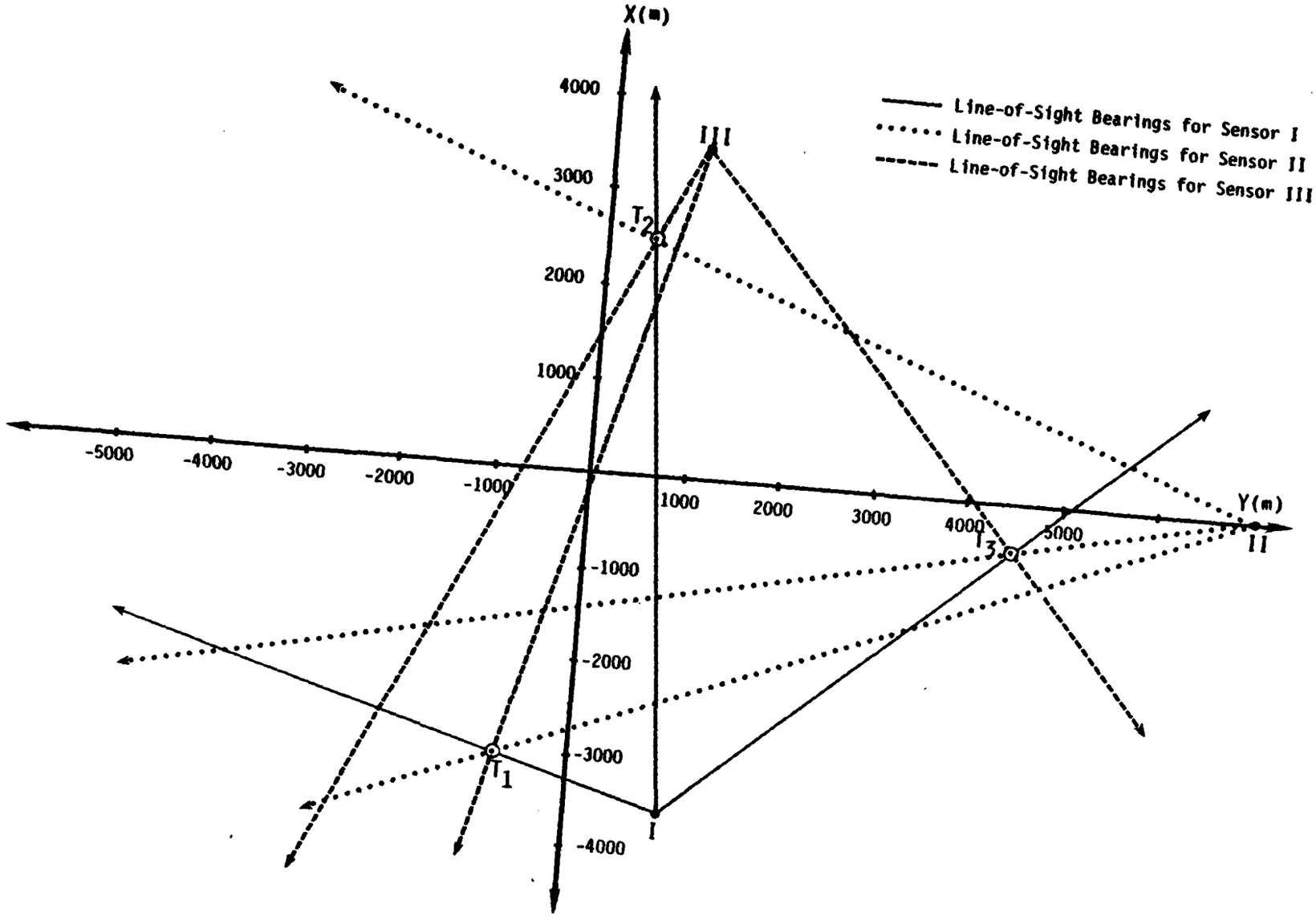


Figure 5.1. Plot of the Line-of-Sight Bearing Intersections Resulting from the Crossed-Bearings Procedure for Scenario 1

The first physical constraint fixes a maximum detection range for the sonobuoys to eliminate candidate initial guess estimates that fall well outside the detection range of the sonobuoy pattern. In real applications, an operator uses parameters such as the class of sonobuoys used, the minimum detectable signal level (MDL) for these sonobuoys, ambient conditions such as sea state, and other factors to set a maximum range for the day for the sonobuoys. For our simulation, we used a transmitted signal-to-noise ratio of 80 dB in a 1 Hz band at a distance 1 yard from the source to simulate each of the target's signal sources. Furthermore, the MDL used for our simulation was set to a reasonable value of 0 dB in a 1 Hz band (this gives +10 dB in a 0.1 Hz band). Given this 80 dB source level, a 0 dB MDL, and assuming a 20 log R signal propagation loss through the water, the maximum detectable range for each of the sonobuoys would be 10,000 yds. To insure a sizable safety margin, a 50% error tolerance was added to this 10,000 yd range, so the maximum allowable range was set at 15,000 yds. Any initial guess estimate whose position values would cause the target to fall outside this 15,000 yd range for any of the observing sensors was then rejected and was never considered again in the MTTA.

The second physical constraint restricted the target's speed to some maximum value which once again was set by the program user. For actual applications, operators usually know what the maximum submerged speeds are for certain classes of targets. If not, the operator could set the upper limit to the maximum known value for any class of targets. For this simulation, the maximum target speed was set to 10 m/sec, which corresponds to roughly 20 knots. Once again, a 50% error tolerance was added to this value so that the maximum allowed speed used by the program was 15 m/sec. This tolerance was

much larger than the speed of the fastest target in the scenario which was 9 m/sec, so this upper limit was considered to be reasonable for our experiments.

With these physical constraints, several of the initial guess estimates could be rejected, so their corresponding intersensor combinations could also be eliminated from further consideration. However, many more combinations passed these two tests than were rejected, so many of the combinations were still under active consideration at this point. If one looks at Figure 5.1, one sees that most of the two-cluster intersensor combinations fall well within the 15,000 yd range of their participating sonobuoys. While the target speed constraint helped reject a few of the remaining combinations, most combinations were still left intact after this test also. Since so many combinations were left, it was decided to perform some statistical tests on the remaining cluster combinations to examine the quality of the initial guess estimates, and to then eliminate all estimates that were found to fit the data poorly in a statistical sense.

5.2 Two Sensor Statistical Compatability Test

After a particular two sensor initial guess estimate has passed the physical constraints test, it is then subjected to a statistical test that measures the consistency of the information contained in the contributing clusters. This test is similar to the test used to determine cluster consistency and is based on the use of indicator variables.

From section 4.2, it will be recalled that the initial guess procedure uses a weighted least squares method to

estimate the target's initial state vector $(x_o, y_o, \dot{x}_o, \dot{y}_o)$. In doing this, it fits a model of the form

$$M_i = B_1 Z_{i1} + B_2 Z_{i2} + B_3 Z_{i3} + B_4 Z_{i4} + \epsilon_i$$

where (using the notation of section 4.1):

$$M_i = \begin{cases} x_1 \sin\beta_1 - y_1 \cos\beta_1 & \text{for sensor 1's bearing data} \\ x_2 \sin\beta_2 - y_2 \cos\beta_2 & \text{for sensor 2's bearing data} \\ c\left[\dfrac{f_o}{f_1}\left(1 + \dfrac{\dot{x}_1 \cos\beta_1 + \dot{y}_1 \sin\beta_1}{c}\right) - 1\right] & \text{For sensor 1 frequency data} \\ c\left[\dfrac{f_o}{f_2}\left(1 + \dfrac{\dot{x}_2 \cos\beta_2 + \dot{y}_2 \sin\beta_2}{c}\right) - 1\right] & \text{For sensor 2 frequency data} \end{cases}$$

$$(B_1, B_2, B_3, B_4) = (x_o, y_o, \dot{x}_o, \dot{y}_o)$$

$$(Z_{i1}, Z_{i2}, Z_{i3}, Z_{i4}) = \begin{cases} (\sin\beta_1, -\cos\beta_1, \Delta t \sin\beta_1, -\Delta t \cos\beta_1) \\ \quad \text{for sensor 1's bearing data} \\ (\sin\beta_2, -\cos\beta_2, \Delta t \sin\beta_2, -\Delta t \cos\beta_2) \\ \quad \text{for sensor 2's bearing data} \\ (0, 0 \cos\beta_1, \sin\beta_1) \quad \text{for sensor 1's} \\ \quad \text{frequency data} \\ (0, 0 \cos\beta_2, \sin\beta_2) \quad \text{for sensor 2's} \\ \quad \text{frequency data} \end{cases}$$

ϵ_i - the measurement error associated with the i^{th} observation.

When using two sensor data to estimate the initial state vector of a particular target, there are only two possibilities:

1) The two clusters contain observations which generate estimates for an actual target, or

2) The two clusters contain information which leads to estimates for a spurious target.

The problem is to separate spurious target estimates from real ones. Figure 5.1 illustrates the potential number of bearing intersections for even a simple example. Because a linear least squares procedure is used, the state vector estimated by the initial guess procedure is a weighted average of all the measurements from both clusters. This will result in producing a great number of the spurious intersections shown in Figure 5.1. The question of separating real from spurious estimates can then be viewed as one of determining whether the two clusters are estimating the same target or not.

One way to do this is to use indicator variables, that is, to formulate the model as above with an additional term Z_{i5};

$$M_i = B_1Z_{i1} + B_2Z_{i2} + B_3Z_{i3} + B_4Z_{i4} + B_5Z_{i5} + \epsilon_i$$

where $Z_{i5} = \begin{cases} 1 & \text{if the observation comes from cluster 1} \\ 0 & \text{if the observation comes from cluster 2.} \end{cases}$

The response model associated with this is

$$E(M_i) = B_1Z_1 + B_2Z_2 + B_3Z_3 + B_4Z_4 + B_5$$
for observations from cluster 1,

$$E(M_i) = B_1Z_1 + B_2Z_2 + B_3Z_3 + B_4Z_4$$
for observations from cluster 2.

If B_5 is significant, then there is a significant difference in the mean response of the system to the data from the two sensors. In essence, B_5 is a measure of the degree of averaging required between the two data sets to generate an initial state vector estimate. A large amount of averaging implies two different targets are producing the estimate, a small amount of averaging implies only one target produced the estimate.

Thus, the two sensor consistency test reduces to a standard linear model test of the significance of the indicator variable Z_5. This corresponds to the hypothesis test

$$H_o: \ B_5 = 0 \qquad \text{versus}$$
$$H_1: \ B_5 \neq 0$$

with test statistic

$$F^* = \frac{SSR\ (Z_5 | Z_1,\ Z_2,\ Z_3,\ Z_4)}{1} \div \frac{SSE\ (Z_1,\ Z_2,\ Z_3,\ Z_4,\ Z_5)}{n_1 + n_2 - 5}$$

where $SSR(Z_5 | Z_1,Z_2,Z_3,Z_4,)$ is the reduction in regression sum of squares caused by including Z_5,

$SSE(Z_1,Z_2,Z_3,Z_4,Z_5)$ is the error sum of squares for the full model,

n_1, n_2 - number of points in clusters 1 and 2, and

F^* - has an F distribution.

The decision rule is:

$$\text{if } F^* \leq F_c\ (1,\ n_1 + n_2 - 5) \text{ accept } H_o$$
$$\text{if } F^* > F_c\ (1,\ n_1 + n_2 - 5) \text{ accept } H_1$$

where

$F_c\ (1,\ n_1 + n_2 - 5)$ is the desired critical point of the F-distribution with 1 and $n_1 + n_2$ -5 degrees of freedom.

Again, because the cost of a type I error is much higher than that of a type II error, the alpha level for this test was set at .005.

5.3 Three Sensor Statistical Compatability Test

Once all two-sensor initial guess estimates have been examined, attention is turned to potential three-sensor initial guess estimates. As with the two-sensor estimates, the three-sensor state estimates are made up of weighted averages

of observations taken from all three clusters. Thus, spurious target estimates which satisfy the physical constraints of the problem often occur.

Figure 5.2 contains a flowchart of the three sensor statistical test. Note that for a particular three-sensor combination to be examined, at least two of the associated two-sensor estimates must have passed the two-sensor test. Once a three-sensor initial guess has been calculated, a nonlinear least squares hypothesis test developed by Gallant (Reference 7) is used to decide whether this estimate is statistically equal to each of the previously computed two-sensor estimates. It is this test which forms the heart of the three-sensor compatability test. If at least two of the associated two-sensor estimates are statistically equal to the three-sensor estimate, then the three-sensor estimate is passed on for further consideration.

The intent of the three-sensor test is to search for consistency among the already computed two-sensor estimates and the current three-sensor estimate. Requiring agreement with at least two of the possible two-sensor estimates prevents one two-sensor estimate from dominating the three-sensor estimate.

The Gallant nonlinear regression test (Reference 7) considers the hypothesis

$$H_o: \vec{X} = \vec{X}_o \quad \text{versus}$$
$$H_1: \vec{X} \neq \vec{X}_o$$

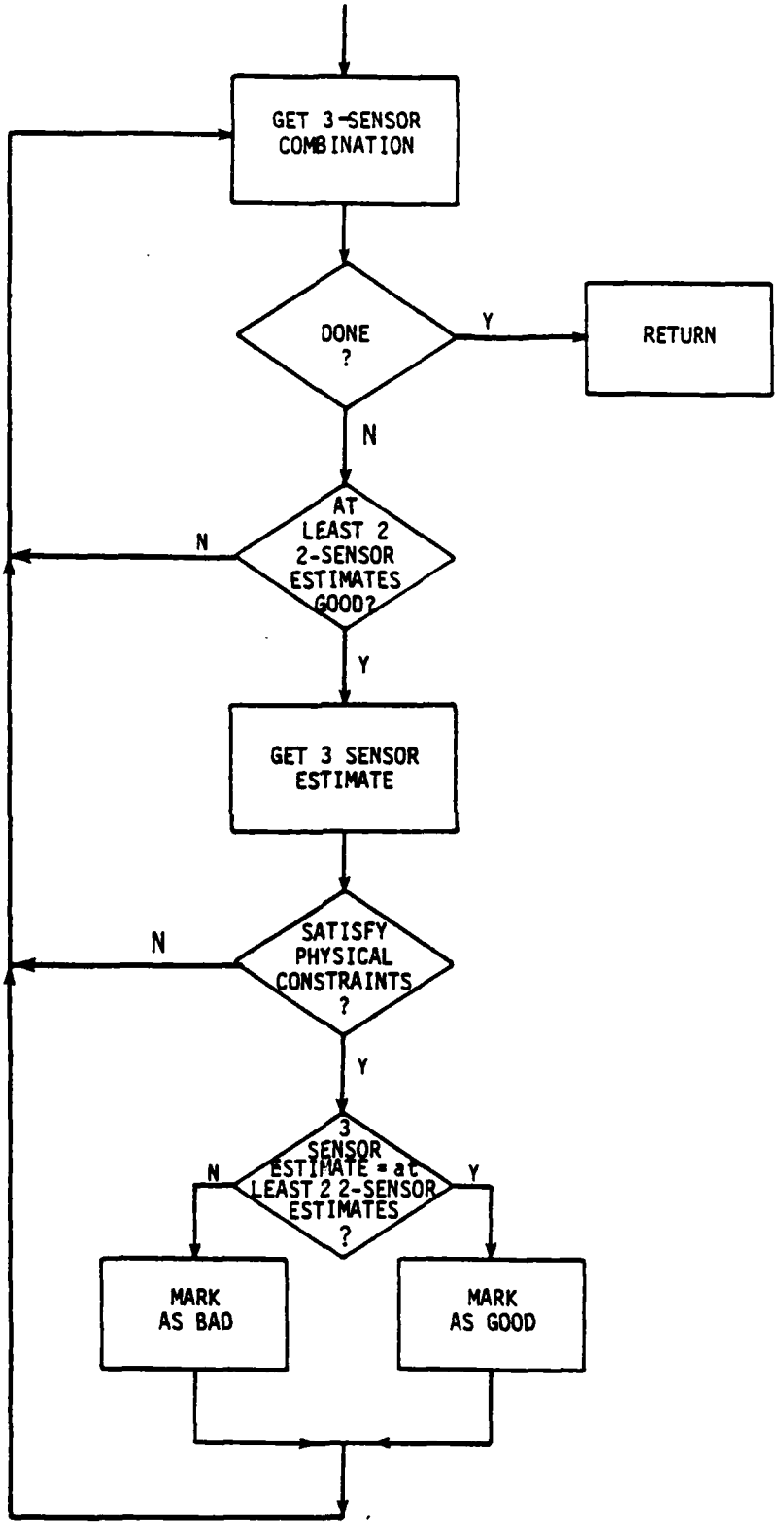


Figure 5.2. Initial Guess Procedure's 3-Sensor Combination Tests

at the α-level of significance where

$\vec{X}_o$ - represents the initial guess, state vector estimate under consideration.

The data, M_i, are measurements assumed to be responses to the input vector $\vec{Z}_i$ linked by the non-linear regression model

$$M_i = f(\vec{Z}_i, \vec{X}_o') + e_i.$$

Under the null hypothesis the quantities

$$M_i - f(\vec{Z}_i, \vec{X}_o') = e_i$$

are called the measurement residuals and are assumed to be independent, normal, random variables with zero mean. If each measurement has an associated positive weight W_i, then the following sum

$$S(\vec{X}_o) = \sum W_i e_i$$

is the weighted residual sum of squares for $\vec{X}_o$.

Then, for the initial guess routine the regression function is:

$$M_i = Z \cdot [(x_o + \dot{x}_o \Delta t_i) \sin \beta_{is} - (y_o + \dot{y}_o \Delta t_i) \cos\beta_{is}] + (1-Z) \cdot [\dot{x}_o \cos \beta_{is} + \dot{y}_o \sin \beta_{is}]$$

where

$(x_o, y_o, \dot{x}_o, \dot{y}_o)$	-	initial state vector of target,
Δt_i	-	$t_i - t_o$,
$\sin \beta_{is}$	-	sine of bearing estimate from sensor s,
$\cos \beta_{is}$	-	cosine of bearing estimate from sensor s,
$Z =$ 1	-	when measurement is crossed-bearing,
$Z =$ 0	-	when measurement is crossed-frequency.

In Gallant's notation, the input vector, $\vec{Z}$, for each measurement is:

$$\vec{Z} = (Z_1, Z_2, Z_3, Z_4) = (Z, \Delta t_i, \sin \beta_{is}, \cos \beta_{is})$$

and the parameter vector is:

$$\vec{X}_o = (x_o, y_o, \dot{x}_o, \dot{y}_o).$$

The test statistic used is

$$T(\vec{X}_o') = \frac{S\left(\vec{X}_o'\right)}{S\left(\vec{X}_o\right)}$$

where $\vec{X}'_o$ - are values for the previously computed two sensor state vector estimates,

$\vec{X}_o$ - the three-sensor state vector estimate.

The critical point is given by

$$C^* = 1 + \frac{pF_\alpha(p,n-p)}{n-p}$$

where

p - number of parameters estimated

n - number of data points

$F_\alpha(p, n-p)$ - α percentage point of the F-distribution with p numerator and n-p denominator degrees of freedom.

Lastly, the decision rule is:

if $T(\vec{X}'_o) \leq c^*$, accept H_o

if $T(X_o) > c^*$, accept H_1.

The test procedure then is:

1) Get the three-sensor data and compute an initial state vector $\vec{X}_o$ with the initial guess procedure,

2) Compute $S(\vec{X}_o)$,

3) For each of the previously computed two-sensor estimates, $\vec{X}'_o$, associated with the three sensors under examination, compute $S(\vec{X}'_o)$ (using the three sensor data),

4) Perform Gallant's test on each of the two-sensor estimates, and

5) If two of the two-sensor estimates pass, then pass the three-sensor estimate as good, otherwise fail it.

The test statistic T is derived by finding the likelihood ratio for the composite hypothesis

$$H_o: \vec{X} = \vec{X}'_o \quad \text{versus}$$
$$H_1: X \neq \vec{X}'_o \quad .$$

The exact distribution of T is not known, but Gallant finds a random variable which is asymptotic to T and whose distribution is known. It is from this asymptotic distribution that C* is determined. Basically, the test examines the ratio of the residual sum of squares for the two-sensor and three-sensor estimates. By definition, the two-sensor estimates will always produce larger values of $S(\vec{X})$ than the three-sensor estimate, so Gallant's test provides a measure of what is too large.

6.0 INTEGER PROGRAMMING AND SCENARIO SELECTION

Up to this point, the MTTA has been concerned primarily with examining multi-sonobuoy data and organizing it into sets of observations that form potential tracks. These potential tracks must now be organized into likely track scenarios. That is, given the current set of observations, determine which set of potential tracks most likely describes the actual events taking place. With even a small number of potential tracks, this becomes a combinatorial problem of fairly large dimensions. To sift through the various potential scenarios to select the optimal set, Tracor has used a linear optimization technique known as integer programming (IP).

The most general integer programming formulation is:

$$\min \quad c'x = \sum_{i=1}^{n} c_i x_i$$

such that

$$X \in S \subset z^n$$

where

- c' is an n-dimensional cost function
- S is the constraint set
- z^n is the set of all n-dimensional integer vectors.

Typically, integer programming problems are difficult to solve so a great deal of research has concentrated on identifying and finding efficient solution techniques for certain special classes of this problem. Fortunately, the scenario selection problem falls into a well known problem type called the set partitioning problem.

6.1 Set Partitioning Formulation

Garfinkel and Nemhauser (Reference 17) give the basic formulation of the set partitioning problem. Consider a set $I = \{1, 2, \ldots, m\}$ and let P be a set such that $P = \{P_1, P_2, \ldots, P_n\}$ where $P_j \subseteq I$ for the index set $J = \{1, \ldots, n\}$. Then a subset J^* of J defines a partition of I if

$$\bigcup_{j \in J^*} P_j = I$$

and for $j, k \in J^*$, $j \neq k$ implies

$$P_j \bigcap P_k = \emptyset.$$

Thus, a partition of I consists of a set of disjoint subsets of I such that their union is equal to I.

Let a cost, c_j, be associated with each $j \in J$ so that the total cost associated with a given partition is $\sum_{j \in J^*} c_j$. The set partitioning problem is to then find the partition of I having the minimum cost. Written as an IP, this problem is:

$$\min \sum_{j=1}^{n} c_j x_j$$

$$\text{subject to} \sum_{j=1}^{n} a_{ij} x_j = 1, \quad i = 1, \ldots, m$$

$$x_i = 0, 1; \quad j = 1, \ldots n$$

where $x_j = \begin{cases} 1 \text{ if } j \text{ is in the cover} \\ 0 \text{ otherwise} \end{cases}$

$a_{ij} = \begin{cases} 1 \text{ if } i \in P_j \\ 0 \text{ otherwise.} \end{cases}$

The optimal solution to this problem, if one exists, will yield an x-vector of zero's and 1's such that a 1 in the k^{th} element of x implies P_k is in the partition and a zero in the k^{th} position implies P_k is not in the partition. This problem can be rewritten in vector notation as

$$\min c'x$$

$$\text{subject to } Ax = 1$$

where

1 is a column vector of 1's
$c = (c_1, c_2, \ldots\ldots, c_n)$
$x = (x_1, x_2, \ldots, x_n)$ x_i as above
$A = (a_{ij})_{mxn}$ a_{ij} as above.

Note that A has m rows an n columns. The k^{th} column of A corresponds to P_k. The m^{th} row of column k has a 1 in it if m is in P_k and is zero otherwise.

6.2 Applying Set Partitioning to the Scenario Selection - Constraints

To see that the scenario selection problem can be cast as a 0-1 set partitioning problem let:

$$K = \{K_1, K_2, \ldots, K_{n_1}, K_{n_1+1}, \ldots, K_{n_1+n_2}, K_{n_1+n_2+1}, \ldots, K_{n_1+n_2+\ldots+n_{n-1}}, K_{n_1+n_2+\ldots+n_{n-1}+1}, \ldots, K_{n_1+n_2\ldots+n_n}\}$$

where $K_1, \ldots, K_{n_1}$ represent the n_1 clusters associated with sensor 1

$K_{n_1+1}, \ldots, K_{n_1+n_2}$ represent the n_2 clusters associated with sensor 2

$\vdots$

$K_{n_1+n_2\ldots+n_{n-1}+1}, \ldots, K_{n_1+n_2+\ldots+n_n}$ represent the n_n clusters associated with sensor n;

Then a particular track scenario can be represented by

$$T^* = T_{j_1} \cup T_{j_2} \cup \ldots \cup T_{j_k}$$

with cost

$$\sum_{j=1}^{n_1+n_2+\ldots+n_n} c_j x_j = c_{j_1} + c_{j_2} + \ldots + c_{j_k}$$

where

T_j represents a given track, i.e. a partition of K, $= K_{j_1}, K_{j_2}, \ldots, K_{j_n}$;

x_j a binary variable such that

$$x_j = \begin{cases} 1 & \text{if track } j \text{ is in the scenario} \\ 0 & \text{if track } j \text{ is not in the scenario } j \end{cases}$$

c_j the cost of track j.

Forming the complement set $T_{j_{k+1}}$ as

$$T_{j_{k+1}} = K - T^*$$

we can write K as the union of the T_{j_i}'s for any track scenario, i.e.

$$K = T_{j_1} \cup T_{j_2} \cup \ldots \cup T_{j_k} \cup T_{j_{k+1}}.$$

Lastly, if we allow a particular cluster to appear in one and only one of the T_{j_i}'s for a particular scenario we have that

$$j_i \neq j_k \text{ implies } T_{j_i} \cap T_{j_k} = \emptyset.$$

Therefore, for any track scenario, the T_j's form a partition of K, the set of all clusters. If each cluster in the complement set $T_{j_{k+1}}$ is considered to be a potential track made up of a singleton cluster with zero cost, then the problem may be written as:

$$\min \; c'x$$

subject to

$$Ax + IS = 1$$

where

c = $(c_1, c_2, \ldots, c_n)$ cost vector

x = $(x_1, x_2, \ldots, x_n)$

x_i = $\begin{cases} 1 \text{ if track } T_i \text{ is in scenario} \\ 0 \text{ if track } T_i \text{ is not in scenario} \end{cases}$

A = $(a_{ij})_{mxn}$

a_{ij} = $\begin{cases} 1 \text{ if cluster } K_i \text{ is in track } T_j \\ 0 \text{ if cluster } K_i \text{ is not in track } T_j \end{cases}$

I = m x m identity matrix

S = $(s_1, s_2, \ldots, s_m)$ vector of slack variables having zero cost

s_i = $\begin{cases} 1 \text{ if cluster } K_i \text{ is in complement set} \\ 0 \text{ otherwise} \end{cases}$

1 = column vector of all ones.

6.3 Applying Set Partitioning to Scenario Selection - Cost Function

This section derives the cost function used in the IP formulation of the scenario selection problem. As explained in previous reports (References 2 and 3), Tracor's HTA uses an extended Kalman filter to sequentially track a given target. The filter is initialized by a batch, least-squares algorithm that uses a constant acceleration

motion model for target trajectories. Two measurement types, bearing and Doppler shifted frequency, are used by the measurement model.

The measurement model of the Kalman filter produces a residual stream as data are processed,

$$e_i = Y_i - \hat{Y}_i$$

where Y_i - observed measurement value

$\hat{Y}_i$ - predicted measurement value.

The e_i's are assumed to be independent, normal random variables with zero mean and variance w_e. Thus for a particular track, the likelihood function for the measurement model residual stream can be computed. For track T_i possessing n_i observations, this function becomes:

$$L_i = \frac{e^{-\frac{1}{2}(e_1/w_1)^2}}{\sqrt{2\pi w_1}} \cdot \frac{e^{-\frac{1}{2}(e_2/w_2)^2}}{\sqrt{2\pi w_2}} \cdots \frac{e^{-\frac{1}{2}(e_{n_i}/w_{n_i})^2}}{\sqrt{2\pi w_{n_i}}}$$

$$= \frac{e^{-\frac{1}{2}\sum_{i=1}^{n_i}(e_i/w_i)^2}}{(2\pi)^{n_i/2}(w_1, w_2 \ldots w_{n_i})^{\frac{1}{2}}} .$$

Let $S_j = \{T_{i_1}, T_{i_2}, \ldots, T_{i_{n_i}}\}$ be a track scenario, then the likelihood of S_j is defined as:

$$L(S_i) = L_{i_1} \; L_{i_2} \cdots \; L_{i_{n_j}}$$

where L_{i_k} is the likelihood function for the measurement

model residual stream corresponding to track T_{i_k}. If x_i is a variable such that

$$x_i = \begin{cases} 1 \text{ if } T_i \text{ is in scenario i} \\ 0 \text{ otherwise,} \end{cases}$$

then the above can be written as

$$L(s_i) = L(S) = L_1^{x_1} L_2^{x_2} \ldots L_n^{x_n}$$

where $x_{i_1}=1$, $x_{i_2}=1$, ..., $x_{i_{n_j}}=1$ and all other $x_i = 0$.

In general then,

$$L(S) = L_1^{x_1} * L_2^{x_2} * \ldots * L_n^{x_n}$$

and a reasonable scenario selection candidate would be that set of x_i's which maximizes L(S). This is equivalent to finding that set of x_i's which minimizes

$$-2 \ln L(S) = x_1 (-2 \ln L_1) + x_2 (-2 \ln L_2) + \ldots + x_n (-2 \ln L_n) .$$

Thus, for the scenario selection problem,

$$c_i = -2 \ln L_i = n_i \ln 2\pi + \sum_{i=1}^{n_i} W_i + \sum_{i=1}^{n_i} (e_i/W_i)^2$$

becomes the cost associated with track T_i, and the optimum scenario is the one that maximizes the scenario likelihood function.

6.4 Integer Programming Summary

In summary, it is important to keep in mind the following aspects of the MTTA integer programming formulation. For the given scenario under consideration the members of the component tracks, T_{j_1}, T_{j_2},..., T_{j_k}, are determined by the initial guess routine, the members of the complement set, $T_{j_{k+1}}$, are the clusters that are "left over." The cost c_j, of a particular track T_j is determined by applying the HTA to the data contained in the clusters which comprise T_j.

The solution algorithm is a list based search procedure described in Garfinkel and Nemhauser (Reference 17) and, while there are potentially faster algorithms available, this formulation was felt to be sufficient at the present time. Larger problems will likely force consideration of faster solution algorithms.

Finally, it should be noted that Morefield (Reference 8) has also used integer programming to solve the intersensor correlation problem. He used a set packing algorithm which, with the addition of slack variables, is equivalent to the set partitioning algorithm. However, his rationale for casting the problem in this framework is substantially different, using a Baysian approach to track formation and his cost function is slightly different. More importantly, he uses data types which ignore the passive initialization problem and he works with substantially smaller data sets. Because of this Morefield used neither clustering techniques or initial guess procedures to prune the data, relying on the IP to handle these chores.

7.0 RESULTS

7.1 Introduction

This section presents the results of applying Tracor's MTTA to several different simulated multi-target scenarios. Topics discussed in this section include:

1) Generation of simulated multi-target data,

2) A three target scenario with good geometry, strong signal, and a threshold of 0 dB in a 1 Hz band,

3) A three target scenario with poor geometry, strong signal, and a threshold of 0 dB in a 1 Hz band,

4) A three target scenario with good geometry, moderate signal, and no threshold, and

5) A two target scenario with good geometry strong signal, and a threshold of 0 dB in a 1 Hz band.

The section corresponding to topic (2) contains a detailed discussion that illustrates the performance of the MTTA. Figures and tables are presented which show the results of the data clustering, automatic cluster extraction, initial guess estimation, and integer programming modules.

All scenarios generated were single trial runs, thus the results presented are not Monte Carlo averages and

reflect more variation than do typical Monte Carlo results. All scenarios were of six minute duration and employed three sensors to gather data. Computing was performed on a UNIVAC 1100/61 where the run times per scenario ranged from seven to nine minutes depending on the scenario complexity and the received signal strength. This point should be noted, Tracor's MTTA is currently running in near real time. With only a minimal amount of parallel processing, i.e., separate processors to cluster each buoy's data and two processors to evaluate potential tracks for the IP module, the algorithm could easily be made to run in real time. Because the MTTA uses relatively little storage space (less than 64 K words for the clustering algorithm and less than 30 K words for all other modules) and requires only six to seven significant digits of numerical accuracy, current parallel processing options would be more than adequate for MTTA computations.

7.2 Scenario Data Generation

To study Tracor's MTTA tracking capabilities, multi-target data were needed to test the tracker. Since no real multiple target data sets were readily available, simulated multiple target data were generated to fill this need. Following is a brief discusssion of the type of sonobuoys modeled, the type of trajectories used by the targets, and the target signal strength levels that were used to simulate the multi-target data.

7.2.1 Sensor Locations

For all the geometries investigated, three non-moving passive sensors were used to gather data from the targets. These sensors were deployed in an equilateral,

triangular arrangement that is known as a tri-tac pattern. The baseline distance between each of the sonobuoys was fixed at 7000 m. This arrangement was felt to be sufficient to insure that each of the targets could be observed by all the sensors.

7.2.2 Target Trajectories

Three different multi-target geometries were simulated for the current study. For each geometry, all the targets followed unique, constant velocity, constant heading trajectories for six minute durations. These trajectories were specifically designed to prevent track intersections during this six minute time span. The three geometries used are summarized in Tables 7.I-III. These scenarios were used to determine if certain geometric and dynamic combinations could be formed that might prevent the MTTA from tracking all the targets. As will be discussed, the scenarios described in Tables 7.I and 7.III proved to be favorable observation geometries whereas the scenario described in Table 7.II proved to be an unfavorable multi-target geometry with the MTTA failing to track one of the three targets.

7.2.3 Simulated DIFAR Multi-Target Data

After the three scenarios described above were created, simulated sonobuoy measurement data for the targets were needed so that the MTTA could attempt to reconstruct the tracks for all the targets. To generate the data for individual targets, a DIFAR simulator (References 2 and 3) was used to generate frequency and bearing measurements. This simulator used a peak picking method to estimate the frequency from a normalized frequency spectrum and an arctangent processor to estimate the bearing. For this simulation, a data

TABLE 7.I

DESCRIPTION OF THE 3 TARGET TRACKS FOR SCENARIO 1

Buoy Positions

Buoy #	X(m)	Y(m)	V(m/s)
1	-3500	1000	0
2	0	7062	0
3	3500	1000	0

Initial Target Tracks (t_o=0s.)

Target #	X_o(m)	Y_o(m)	V_o(m/s)	Course Heading (°)
1	-3500	-750	6	45
2	2500	500	9	90
3	-500	4500	4	300

Final Target Tracks (t_f=360s)

Target #	X_f(m)	Y_f(m)	V_f(m/s)	Course Heading (°)
1	-1473	777	6	45
2	2500	3740	9	90
3	220	3253	4	300

TABLE 7.II

DESCRIPTION OF THE 3 TARGET TRACKS FOR SCENARIO 2

Buoy Positions

Buoy #	X(m)	Y(m)	V(m/s)
1	-3500	1000	0
2	0	7062	0
3	3500	1000	0

Initial Target Tracks (t_o=0s.)

Target #	X_o(m)	Y_o(m)	V_o(m/s)	Course Heading (°)
1	-3000	-750	6	15
2	2500	500	9	90
3	-500	4500	4	300

Final Target Tracks (t_f=360s)

Target #	X_o(m)	Y_o(m)	V_o(m/s)	Course Heading (°)
1	-914	-191	6	15
2	2500	3740	9	90
3	220	3253	4	300

TABLE 7.III

DESCRIPTION OF THE 2 TARGET
TRACK FOR SCENARIO 4

Buoy Positions

Buoy #	X(m)	Y(m)	V(m/s)
1	-3500	1000	0
2	0	7062	0
3	3500	1000	0

Initial Target Tracks (t_o=0s.)

Target #	X_o(m)	Y_o(m)	V_o(m/s)	Course Heading (°)
1	-1000	-500	6	0
2	-3000	7000	6	300

Final Target Tracks (t_f=360s)

Target #	X_f(m)	Y_f(m)	V_f(m/s)	Course Heading (°)
1	1160	-500	6	0
2	-1920	5129	6	300

update rate of 10 sec. was used to make the measurement, and since the accuracy of the frequency estimates is inversely proportional to the update rate, the frequency measurements were limited to a 0.1 Hz resolution. As described before (References 2 and 3), this data generation program is a fairly realistic model which outputs non-Gaussian measurements for the individual targets.

To form the multi-target data needed for the current investigation, individual target data sets were generated for each of the targets present in a given scenario. Then, the individual target data sets were merged for each of the sonobuoys to produce what was referred to last year as multiple linetracker data (Reference 3). Certain weaknesses were acknowledged previously in simulating the data in this fashion, but overall, this type of multiple target data was felt to be more than adequate for testing the MTTA.

For all but one case, each target transmitted a tone whose SNR level as measured 1 yard from the source was simulated as 80 dB in a 1 Hz band. For these cases, the DIFAR simulator used a threshold of 0 dB in a 1 Hz band that had to be met or exceeded before any measurement estimates were output. If one uses a 20 log R approximation for the propagation loss, where R is the magnitude of the distance from the source to the receiver, an 80 dB source level without any fluctuation could be heard 10,000 yards away and just meet the 0 dB threshold criterion. However, the DIFAR simulator introduces some random noise terms to model random fluctuations in both the target signal source level and the ambient noise level, so the absolute maximum observation range for the sonobuoys cannot be set.

One other case was simulated which used a lower SNR value for the target source level, but eliminated the thresholding criterion so that measurement estimates were made for all time updates. Usually, threshold levels are set so that most target measurements exceed the level but no, or almost no, random noise measurements are introduced into the data set. Unfortunately, such thresholding techniques can eliminate actual measurements in order to insure that no noise is included in the data. For this particular case, the threshold was dropped to see if the sensors could pick up a weaker signal than that used in the other simulations. The target source level was lowered by 3 dB to 77 dB in a 1 Hz band. This 3 dB loss effectively decreased by approximately 30% the range at which the signal could be heard. To compensate for this loss in range of the detection system, the threshold was eliminated so that measurement updates were always made. The clustering algorithms were then used to pick the true measurements from the random noise found in the data. The results from eliminating the measurement threshold and then clustering the data to separate true target data from random noise are presented in a subsequent subsection.

7.3 Three Target Scenario, Good Geometry, Strong Signal

7.3.1 Introduction

This section contains a discussion of the results obtained by applying the MTTA to a three target scenario with good observation geometry in a low noise environment. It is very likely that no multi-target algorithm will be able to handle all possible observation geometries, so this report

contains examples of the MTTA's performance for both good and bad observation geometries. Furthermore, real ocean ambient noise levels and target source levels vary drastically from one situation to the next, so detected SNR's vary greatly for different encounters. This variation in detected SNR's seriously affects the quality of the measurements which in turn greatly affects tracking accuracy. By running a three target scenario under various degrees of signal degradation, it was hoped that a better idea of algorithm performance could be obtained.

Figure 7.1 contains a plot of this scenario. The plus signs represent the buoys, the solid lines the actual trajectories, and the dashed lines MTTA's estimated trajectories. Detailed results of the MTTA's solutions from each module are presented next.

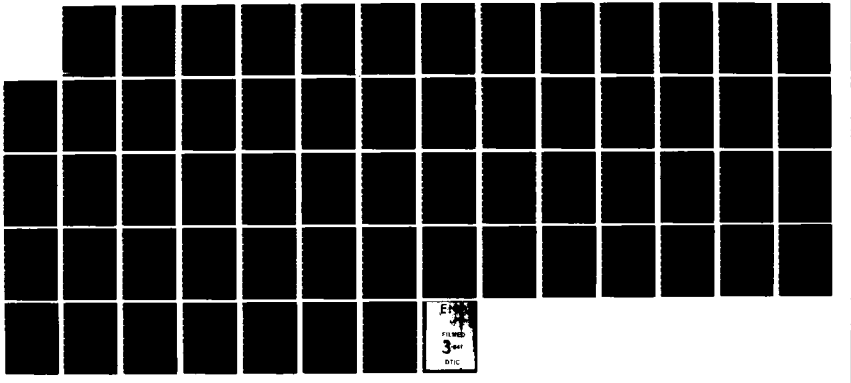
7.3.2 Data Clustering Results - Buoys 1 and 3

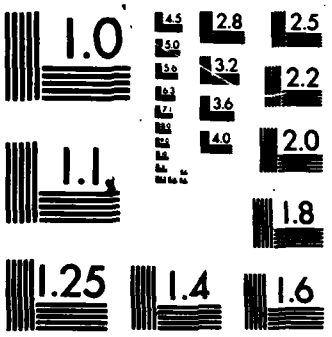
Table 7.IV contains the frequency, bearing sine, and bearing cosine measurements generated by buoy 1, while Table 7.V contains the measurements generated by buoy 3 of the scenario. The column headed 'TARGET' indicates whether the measurement came from target 1, target 2, or target 3.

Tables 7.VI and 7.VII present the clusters generated by Ling's algorithm (for buoys 1 and 3, respectively), the associated cluster level or node number (see section 3.2), and the object numbers for the members of each cluster. Examination of these tables illustrates several points:

AD-A137 564 APPLICATIONS OF CLUSTER ANALYSES AND INTEGER PROGRAMMING TO MULTIPLE TARG..(U) TRACOR INC AUSTIN TX APPLIED SCIENCES GROUP D COOPER ET AL. 21 MAR 83 2/2

UNCLASSIFIED TRACOR-T83-AU-9552-U N00014-82-C-0044 F/G 12/1 NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

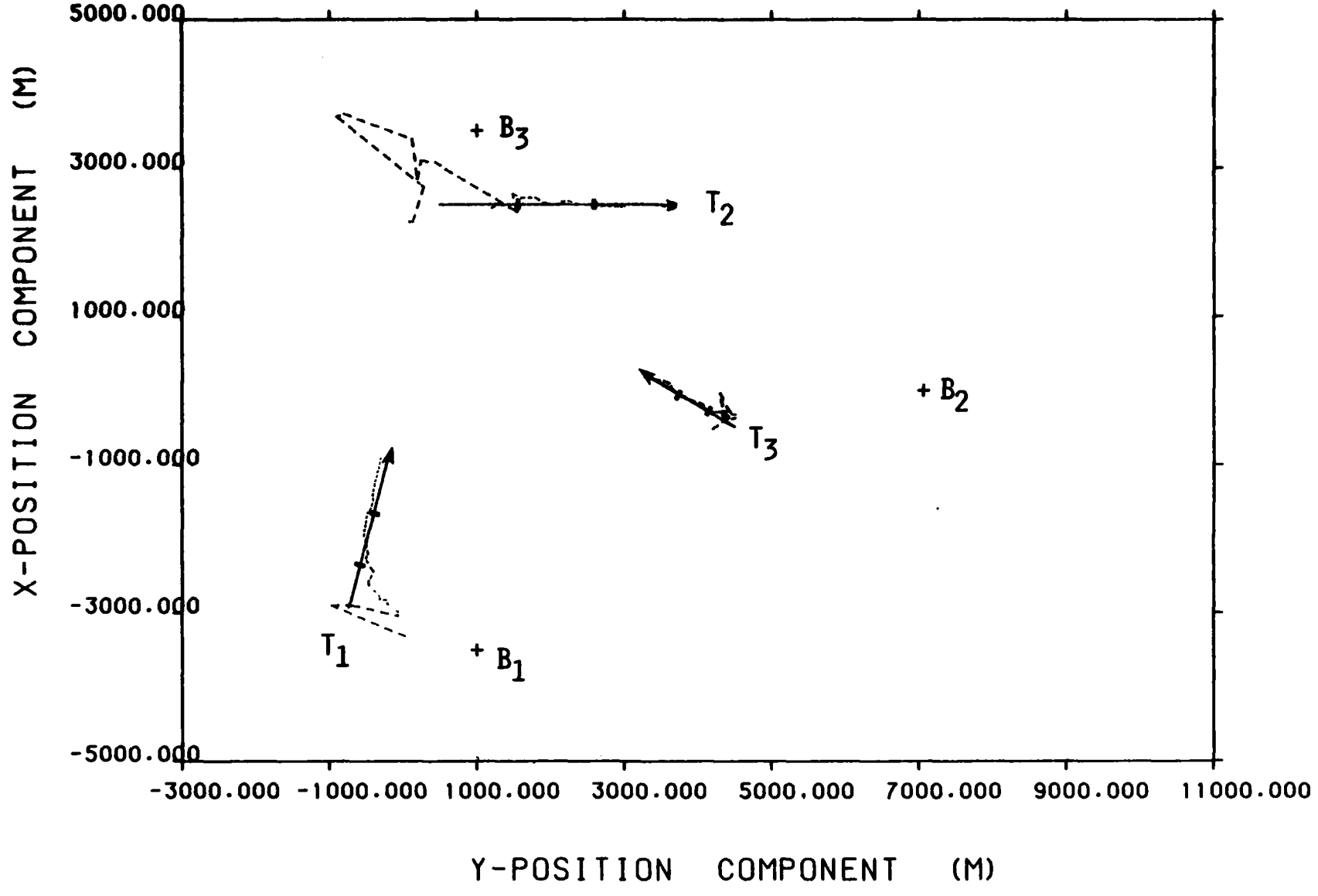


Figure 7.1. Plot of the MTTA's Estimated Tracks vs. the True Tracks for the 3-Target, Strong Signal and Good Geometry Scenario

TABLE 7.IV

SIMULATED MULTI-TARGET DIFAR DATA FOR
BUOY 1 OF SCENARIO 1

SAMPLE NO.	TIME	FREQUENCY	COS(β)	SIN(β)	TARGET NO.
1	15.0000	150.000	.308343	-.951275	1
2	15.0000	150.100	.655623	.755089	3
3	25.0000	149.900	.348433	-.937334	1
4	25.0000	150.100	.642860	.765984	3
5	35.0000	149.900	.391800	-.920050	1
6	35.0000	150.000	.970826	-.239784	2
7	35.0000	150.100	.778779	.627298	3
8	45.0000	149.900	.370085	-.928999	1
9	55.0000	149.900	.328874	-.944374	1
10	55.0000	150.000	.996433	-.643933-001	2
11	55.0000	150.100	.661244	.750171	3
12	65.0000	149.900	.447230	-.894419	1
13	65.0000	150.100	.638200	.769871	3
14	75.0000	150.100	.556584	.830792	3
15	85.0000	149.800	.560186	-.828367	1
16	85.0000	150.000	.999073	-.430587-001	2
17	85.0000	150.100	.741068	.671430	3
18	95.0000	149.800	.529447	-.848343	1
19	95.0000	150.000	.997932	-.642841-001	2
20	95.0000	150.100	.698291	.715814	3
21	105.000	149.900	.989046	.147611	2
22	115.000	149.800	.756462	-.654037	1
23	115.000	149.900	.999826	-.186446-001	2
24	115.000	150.100	.712585	.701586	3
25	125.000	149.800	.743048	-.669238	1
26	125.000	149.900	.975569	.219694	2
27	125.000	150.100	.672624	.739984	3
28	135.000	149.800	.617491	-.786578	1
29	135.000	149.900	.975051	.221983	2
30	135.000	150.100	.799014	.601313	3
31	145.000	149.800	.682499	-.730887	1
32	145.000	149.900	.993101	.117262	2
33	155.000	149.700	.671624	-.740892	1

(CONTINUED)

TABLE 7.IV (CONCLUDED)

SAMPLE NO.	TIME	FREQUENCY	COS(β)	SIN(β)	TARGET NO.
34	155.000	149.900	.908245	.418439	2
35	165.000	149.700	.610771	-.791807	1
36	165.000	149.900	.999421	.340218-001	2
37	175.000	149.700	.701946	-.712230	1
38	175.000	149.900	.987672	.156538	2
39	175.000	150.100	.819604	.572931	3
40	185.000	149.700	.689152	-.724617	1
41	185.000	150.100	.777887	.628405	3
42	195.000	149.900	.999920	-.126506-001	2
43	195.000	150.100	.808951	.587876	3
44	205.000	149.700	.745327	-.666699	1
45	205.000	149.800	.973447	.228911	2
46	205.000	150.100	.797952	.602721	3
47	215.000	149.700	.761986	-.647593	1
48	215.000	149.800	.963391	.268099	2
49	215.000	150.100	.790408	.612580	3
50	225.000	149.700	.777883	-.628409	1
51	235.000	149.700	.792367	-.610045	1
52	235.000	149.800	.935300	.353854	2
53	235.000	150.100	.759726	.650248	3
54	245.000	149.600	.617720	-.575616	1
55	245.000	150.000	.696261	.717769	3
56	255.000	149.600	.845389	-.534151	1
57	255.000	150.000	.813719	.581259	3
58	265.000	149.600	.817620	-.575758	1
59	265.000	150.000	.775536	.631704	3
60	275.000	149.600	.866447	-.495783	1
61	295.000	149.600	.863743	-.503933	1
62	295.000	149.700	.917841	.396947	2
63	295.000	150.000	.796430	.604730	3
64	305.000	149.600	.884648	-.466260	1
65	315.000	149.600	.872472	-.488667	1
66	325.000	149.600	.841562	-.540160	1
67	325.000	149.700	.948582	.316532	2
68	325.000	150.000	.821892	.569646	3
69	335.000	149.600	.894065	-.447937	1
70	335.000	149.700	.966681	.255982	2
71	335.000	150.000	.771498	.636231	3
72	345.000	149.600	.908669	-.417516	1
73	345.000	150.000	.911435	.411444	3
74	355.000	149.600	.919773	-.392450	1
75	355.000	150.000	.812330	.583198	3

TABLE 7.V

SIMULATED MULTI-TARGET DIFAR DATA FOR
BUOY 3 OF SCENARIO 1

SAMPLE NO.	TIME	FREQUENCY	COS(β)	SIN(β)	TARGET NO.
1	5.00000	150.600	-.957011	-.290051	1
2	5.00000	150.400	-.881346	-.472472	2
3	15.0000	150.600	-.991340	-.131323	1
4	15.0000	150.300	-.936695	-.350145	2
5	25.0000	150.600	-.996409	.846668-001	1
6	25.0000	150.200	-.969837	-.243756	2
7	25.0000	150.300	-.790688	.612219	3
8	35.0000	150.600	-.988933	-.148360	1
9	35.0000	150.100	-.988085	-.153907	2
10	45.0000	150.600	-.938766	-.344555	1
11	45.0000	150.100	-.996927	-.783859-001	2
12	45.0000	150.300	-.707389	.706824	3
13	55.0000	150.000	-.999433	-.336628-001	2
14	55.0000	150.300	-.744818	.667267	3
15	65.0000	150.000	-.998722	.505450-001	2
16	65.0000	150.300	-.720066	.693885	3
17	75.0000	149.900	-.988110	.153750	2
18	75.0000	150.300	-.797587	.616209	3
19	85.0000	150.600	-.934025	-.357209	1
20	85.0000	149.800	-.969759	.244063	2
21	95.0000	149.800	-.959518	.281646	2
22	105.000	150.600	-.985134	-.171788	1
23	105.000	149.600	-.922599	.385761	2
24	115.000	149.600	-.866289	.499543	2
25	115.000	150.300	-.815628	.578576	3
26	125.000	150.600	-.820882	-.571098	1
27	125.000	149.600	-.840743	.541434	2
28	135.000	150.600	-.966779	-.255615	1
29	135.000	149.500	-.793701	.612719	2
30	145.000	150.300	-.669386	.742915	3
31	155.000	150.600	-.896914	-.442408	1
32	155.000	149.500	-.777783	.628533	2
33	155.000	150.300	-.731440	.681906	3

(CONTINUED)

TABLE 7.V (CONTINUED)

SAMPLE NO.	TIME	FREQUENCY	COS(β)	SIN(β)	TARGET NO.
34	165.000	150.600	-.969606	-.244673	1
35	165.000	149.400	-.727551	.686053	2
36	165.000	150.300	-.797280	.603609	3
37	175.000	150.600	-.915253	-.402879	1
38	175.000	149.400	-.738751	.673978	2
39	175.000	150.300	-.645460	.763794	3
40	185.000	150.600	-.961503	-.274796	1
41	185.000	149.400	-.601320	.799008	2
42	185.000	150.300	-.718704	.695316	3
43	195.000	149.300	-.631385	.775470	2
44	195.000	150.300	-.712829	.701744	3
45	205.000	149.300	-.603445	.797404	2
46	205.000	150.300	-.847868	.530207	3
47	215.000	150.600	-.959561	-.281502	1
48	215.000	149.300	-.603835	.797110	2
49	225.000	149.300	-.548741	.835993	2
50	225.000	150.300	-.823967	.566638	3
51	235.000	150.600	-.976378	-.216070	1
52	235.000	149.300	-.533599	.845737	2
53	235.000	150.300	-.704284	.709998	3
54	245.000	150.600	-.981613	-.190884	1
55	245.000	150.300	-.768649	.639677	3
56	255.000	149.300	-.492151	.870510	2
57	255.000	150.300	-.733882	.679277	3
58	265.000	150.600	-.948240	-.317555	1
59	265.000	150.300	-.776910	.627136	3
60	275.000	149.300	-.463732	.885976	2
61	285.000	149.300	-.404698	.914451	2
62	285.000	150.300	-.791906	.623521	3
63	295.000	149.300	-.469664	.882845	2
64	295.000	150.300	-.750253	.661151	3
65	305.000	150.600	-.996470	.552907-001	1
66	305.000	149.200	-.390199	.920731	2
67	305.000	150.300	-.761725	.647900	3
68	315.000	149.200	-.389568	.920999	2
69	315.000	150.300	-.708016	.706196	3
70	325.000	150.600	-.982243	-.187611	1
71	325.000	150.300	-.862839	.505479	3
72	335.000	150.600	-.966961	-.254923	1
73	335.000	149.200	-.379682	.925117	2
74	335.000	150.300	-.777935	.628344	3
75	345.000	149.200	-.435891	.900000	2
76	345.000	150.300	-.906276	.422473	3

(CONTINUED)

TABLE 7.V (CONCLUDED)

SAMPLE NO.	TIME	FREQUENCY	COS(β)	SIN(β)	TARGET NO.
77	355.000	150.600	-.976356	-.216167	1
78	355.000	149.200	-.265759	.964039	2
79	355.000	150.300	-.821147	.570723	3

TABLE 7.VI

CLUSTERING OUTPUT FOR BUOY 1 OF SCENARIO 1

CLUSTER	NODE	SIZE	OBJECTS CLUSTERED
1	1	2	28 29
2	2	2	46 49
3	3	2	2 4
4	4	2	37 40
5	5	3	43 46 49
6	6	2	22 25
7	7	2	16 19
8	8	2	50 51
9	9	2	5 8
10	10	3	47 50 51
11	11	4	44 47 50 51
12	12	2	64 65
13	13	2	72 74
14	14	2	54 58
15	15	2	11 13
16	16	2	60 61
17	17	3	69 72 74
18	18	4	60 61 64 65
19	19	4	39 43 46 49

(CONTINUED)

TABLE 7.VI (CONTINUED)

CLUSTER	NODE	SIZE	OBJECTS CLUSTERED
20	20	7	20 24
21	21	2	15 18
22	22	3	3 5 8
23	23	2	45 48
24	26	5	39 41 43 46 49
25	27	4	3 5 8 9
26	29	3	54 56 58
27	30	2	68 75
28	32	3	33 37 40
29	35	7	60 61 64 65 69 72 74
30	38	4	2 4 11 13
31	39	10	54 56 58 60 61 64 65 69 72 74
32	41	3	20 24 27
33	43	6	39 41 43 46 49 53
34	46	2	59 63
35	49	11	54 56 58 60 61 64 65 66 69 72 74
36	50	2	32 38
37	51	4	17 20 24 27
38	52	3	10 16 19
39	54	3	57 59 63
40	55	2	67 70
41	57	5	57 59 63 68 75

(CONTINUED)

TABLE VI (CONTINUED)

CLUSTER	NODE	SIZE	OBJECTS CLUSTERED
42	65	3	21 32 38
43	66	6	57 59 63 68 71 75
44	68	7	30 39 41 43 46 49 53
45	72	5	3 5 6 9 12
46	73	7	33 37 40 44 47 50 51
47	74	5	21 26 29 32 38
48	75	8	33 35 37 40 44 47 50 51
49	77	8	2 4 11 13 17 20 24 27
50	81	2	28 31
51	85	6	21 26 29 32 36 38
52	91	4	24 25 28 31
53	95	3	45 48 52
54	98	7	21 23 26 29 32 36 38
55	100	3	62 67 70
56	102	9	2 4 7 11 13 17 20 24 27
57	103	10	2 4 7 11 13 14 17 20 24 27
58	113	6	15 18 22 25 28 31
59	130	7	55 57 59 63 68 71 75
60	134	17	2 4 7 11 13 14 17 20 24 27 30 39 41 43 46 49 53
61	168	4	6 10 16 19
62	198	8	55 57 59 63 68 71 73 75
63	217	14	15 18 22 25 28 31 33 35 37 40 44 47 50 51

(CONTINUED)

TABLE 7.VI (CONCLUDED)

CLUSTER	NODE	SIZE	OBJECTS CLUSTERED
64	221	6	1 3 5 8 9 12
65	222	25	15 18 22 25 28 31 33 35 37 40 44 47 50 51 54 56 58 60 61 64 65 66 69 72 74
66	225	11	6 10 16 19 21 23 26 29 32 36 38
67	226	25	2 4 7 11 13 14 17 20 24 27 30 39 41 43 46 49 53 55 57 59 63 68 71 73 75
68	230	12	6 10 16 19 21 23 26 29 32 34 36 38
69	234	13	6 10 16 19 21 23 26 29 32 34 36 38 42
70	248	16	6 10 16 19 21 23 26 29 32 34 36 38 42 45 48 52
71	260	31	1 3 5 8 9 12 15 18 22 25 28 31 33 35 37 40 44 47 50 51 54 56 58 60 61 64 65 66 69 72 74
72	265	19	6 10 16 19 21 23 26 29 32 34 36 38 42 45 48 52 62 67 70
73	455	44	2 4 6 7 10 11 13 14 16 17 19 20 21 23 24 26 27 29 30 32 38 36 38 39 41 42 43 45 46 48 49 52 53 55 57 59 62 63 67 68 70 71 73 75
74	864	75	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75

TABLE 7.VII

CLUSTERING OUTPUT FOR BUOY 3
OF SCENARIO 1

CLUSTER	NODE	SIZE	OBJECTS CLUSTERED
1	1	2	45 46
2	2	2	66 68
3	3	2	42 44
4	4	2	35 38
5	5	2	64 67
6	6	2	49 52
7	7	2	51 54
8	8	2	59 62
9	9	2	60 63
10	10	3	66 68 73
11	11	3	55 59 62
12	12	2	5 8
13	13	2	12 16
14	14	2	29 32
15	15	3	43 45 48
16	16	3	12 14 16
17	17	2	20 21
18	18	2	34 40
19	19	5	55 59 62 64 67

(CONTINUED)

TABLE 7.VII (CONTINUED)

CLUSTER	NODE	SIZE	OBJECTS CLUSTERED
20	20	3	56 60 63
21	22	3	34 40 47
22	23	4	28 34 40 47
23	25	2	24 27
24	27	2	72 77
25	28	3	33 42 44
26	29	2	53 57
27	30	2	46 50
28	31	2	31 37
29	32	7	53 55 57 59 62 64 67
30	33	8	53 55 57 59 62 64 67 74
31	35	3	70 72 77
32	36	5	49 52 56 60 63
33	37	2	36 39
34	38	4	66 68 73 75
35	40	6	49 52 56 60 61 63
36	41	11	33 42 44 53 55 57 59 62 64 67 74
37	42	2	10 19
38	43	9	43 45 46 49 52 56 60 61 63
39	45	12	33 42 44 53 55 57 59 62 64 67 69 74
40	54	4	12 14 16 18
41	56	6	28 34 40 47 51 54

(CONTINUED)

TABLE 7.VII (CONTINUED)

CLUSTER	NODE	SIZE	OBJECTS CLUSTERED
42	57	2	9 11
43	59	13	33 42 44 53 55 57 59 62 64 67 69 74 79
44	61	5	12 14 16 18 25
45	64	6	7 12 14 16 18 25
46	67	2	13 15
47	71	3	1 10 19
48	72	15	30 33 39 42 44 53 55 57 59 62 64 67 69 74 79
49	74	7	7 12 14 16 18 25 36
50	75	7	28 34 40 47 51 54 58
51	76	16	30 33 39 42 44 53 55 57 59 62 64 67 69 71 74 79
52	83	18	30 33 39 42 44 46 50 53 55 57 59 62 64 67 69 71 74 79
53	85	8	22 28 34 40 47 51 54 58
54	86	19	30 33 39 42 44 46 50 53 55 57 59 62 64 67 69 71 74 76 79
55	92	26	7 12 14 16 18 25 30 33 36 39 42 44 46 50 53 55 57 59 62 64 67 69 71 74 76 79
56	107	10	3 8 22 28 34 40 47 51 54 58
57	117	5	66 68 73 75 76
58	121	3	23 24 27
59	127	13	3 8 22 28 34 40 47 51 54 58 70 72 77
60	133	16	1 3 8 10 19 22 28 34 40 47 51 54 58 70 72 77
61	135	18	1 3 8 10 19 22 28 31 34 37 40 47 51 54 58 70 72 77
62	171	19	1 3 8 10 19 22 26 28 31 34 37 40 47 51 54 58 70 72 77

(CONTINUED)

TABLE 7.VII (CONCLUDED)

CLUSTER	NODE	SIZE	OBJECTS CLUSTERED
63	195	3	35 38 41
64	250	12	35 38 41 43 45 48 49 52 56 60 61 63
65	253	17	35 38 41 43 45 48 49 52 56 60 61 63 66 68 73 75 76
66	260	4	9 11 13 15
67	270	19	29 32 35 38 41 43 45 48 49 52 56 60 61 63 66 68 73 75 78
68	281	20	1 3 5 8 10 19 22 26 28 31 34 37 40 47 51 54 56 70 72 77
69	288	22	23 24 27 29 32 35 38 41 43 45 46 49 52 56 60 61 63 66 68 73 75 78
70	297	5	6 9 11 13 15
71	298	3	17 20 21
72	310	8	6 9 11 13 15 17 20 21
73	317	9	4 6 9 11 13 15 17 20 21
74	345	10	2 4 6 9 11 13 15 17 20 21
75	356	21	1 3 5 8 10 19 22 26 26 31 34 37 40 47 51 54 58 65 70 72 77
76	587	32	2 4 6 9 11 13 15 17 20 21 23 24 27 29 32 35 38 41 43 45 48 49 52 56 60 61 63 66 68 73 75 76
77	606	53	1 2 3 4 5 6 8 9 10 11 13 15 17 19 20 21 22 23 24 26 27 28 29 31 32 34 35 37 38 40 41 43 45 47 48 49 51 52 54 56 58 60 61 63 65 66 68 70 72 73 75 77 78
78	784	79	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79

1) The hierarchical nature of the data organization imposed by the algorithm;

2) The varying lengths of time that clusters remain isolated before becoming subsets of other larger clusters; and

3) All clusters are eventually merged to form a cluster containing the whole set of data.

Note, that for buoy 3, cluster 75 contains exclusively all the points generated by target 1, cluster 76 does the same for target 2, and cluster 55 contains all the points associated with target 3. Similarly for buoy 1, cluster 71 contains only target 1 data, cluster 72 contains target 2 data, and cluster 67 contains all the data from target 3. Thus, the clustering algorithm has clearly separated the target data, now the MTTA must seek to automatically identify those clusters which are most likely to contain the desired individual target data.

Tables 7.VIII and 7.IX contain the isolation indices and survival function values for all of the clusters generated (see sections 3.2 and 3.3). Keeping only those clusters whose survival function values are less than 10^{-4}, it can be seen that not only do all the candidate clusters pass the isolation test, but so do several others. Thus, additional cluster extraction techniques are needed to pick out the appropriate data. This is discussed in the next section.

TABLE 7.VIII

SURVIVAL FUNCTION AND ISOLATION INDEX VALUES FOR CLUSTERS FROM BUOY 1 OF SCENARIO 1

			ISOLATION INDEX
CLUSTER NO. =	1	SURVIVAL FUNCTION = .1830669-001	73
CLUSTER NO. =	2	SURVIVAL FUNCTION = .8501675	3
CLUSTER NO. =	3	SURVIVAL FUNCTION = .1487049	35
CLUSTER NO. =	4	SURVIVAL FUNCTION = .2180120	28
CLUSTER NO. =	5	SURVIVAL FUNCTION = .3200111	14
CLUSTER NO. =	6	SURVIVAL FUNCTION = .9302080-002	85
CLUSTER NO. =	7	SURVIVAL FUNCTION = .8555628-001	45
CLUSTER NO. =	8	SURVIVAL FUNCTION = .8972366	2
CLUSTER NO. =	9	SURVIVAL FUNCTION = .4933486	13
CLUSTER NO. =	10	SURVIVAL FUNCTION = .9218807	1
CLUSTER NO. =	11	SURVIVAL FUNCTION = .1112133-002	62
CLUSTER NO. =	12	SURVIVAL FUNCTION = .7217808	6
CLUSTER NO. =	13	SURVIVAL FUNCTION = .8046436	4
CLUSTER NO. =	14	SURVIVAL FUNCTION = .4417269	15
CLUSTER NO. =	15	SURVIVAL FUNCTION = .2850254	23
CLUSTER NO. =	16	SURVIVAL FUNCTION = .8969467	2
CLUSTER NO. =	17	SURVIVAL FUNCTION = .2293015	18
CLUSTER NO. =	18	SURVIVAL FUNCTION = .1566442	17
CLUSTER NO. =	19	SURVIVAL FUNCTION = .4666636	7
CLUSTER NO. =	20	SURVIVAL FUNCTION = .3173511	21
CLUSTER NO. =	21	SURVIVAL FUNCTION = .6106839-002	92

(CONTINUED)

TABLE 7.VIII (CONTINUED)

			ISOLATION INDEX
CLUSTER NO. =	22	SURVIVAL FUNCTION = .6644110	5
CLUSTER NO. =	23	SURVIVAL FUNCTION = .1872693-001	72
CLUSTER NO. =	24	SURVIVAL FUNCTION = .9803829-001	17
CLUSTER NO. =	25	SURVIVAL FUNCTION = .7076964-002	45
CLUSTER NO. =	26	SURVIVAL FUNCTION = .4401400	10
CLUSTER NO. =	27	SURVIVAL FUNCTION = .2269791	27
CLUSTER NO. =	28	SURVIVAL FUNCTION = .3375559-001	41
CLUSTER NO. =	29	SURVIVAL FUNCTION = .4659114	4
CLUSTER NO. =	30	SURVIVAL FUNCTION = .1351269-001	39
CLUSTER NO. =	31	SURVIVAL FUNCTION = .6603140-001	10
CLUSTER NO. =	32	SURVIVAL FUNCTION = .4384878	10
CLUSTER NO. =	33	SURVIVAL FUNCTION = .1611562-001	25
CLUSTER NO. =	34	SURVIVAL FUNCTION = .6437461	8
CLUSTER NO. =	35	SURVIVAL FUNCTION = .2384186-006	173
CLUSTER NO. =	36	SURVIVAL FUNCTION = .4368393	15
CLUSTER NO. =	37	SURVIVAL FUNCTION = .5632228-001	26
CLUSTER NO. =	38	SURVIVAL FUNCTION = .5559623-004	116
CLUSTER NO. =	39	SURVIVAL FUNCTION = .7801826	3
CLUSTER NO. =	40	SURVIVAL FUNCTION = .8179311-001	45
CLUSTER NO. =	41	SURVIVAL FUNCTION = .2886278	9
CLUSTER NO. =	42	SURVIVAL FUNCTION = .4729810	9
CLUSTER NO. =	43	SURVIVAL FUNCTION = .2165139-004	64

(CONTINUED)

TABLE 7.VIII (CONTINUED)

			ISOLATION INDEX
CLUSTER NO. = 44	SURVIVAL FUNCTION =	.2652407-005	66
CLUSTER NO. = 45	SUPVIVAL FUNCTION =	.4619360-006	149
CLUSTER NO. = 46	SURVIVAL FUNCTION =	.6786491	2
CLUSTER NO. = 47	SURVIVAL FUNCTION =	.2166149	11
CLUSTER NO. = 48	SURVIVAL FUNCTION =	.2682209-006	142
CLUSTER NO. = 49	SURVIVAL FUNCTION =	.3830850-002	25
CLUSTER NO. = 50	SURVIVAL FUNCTION =	.5722709	10
CLUSTER NO. = 51	SUPVIVAL FUNCTION =	.1132821	13
CLUSTER NO. = 52	SURVIVAL FUNCTION =	.8452478-001	22
CLUSTER NO. = 53	SURVIVAL FUNCTION =	.2264977-005	153
CLUSTER NO. = 54	SURVIVAL FUNCTION =	.3129244-006	127
CLUSTER NO. = 55	SURVIVAL FUNCTION =	.1102686-005	165
CLUSTER NO. = 56	SUPVIVAL FUNCTION =	.7777778	1
CLUSTER NO. = 57	SUPVIVAL FUNCTION =	.1671453-003	31
CLUSTER NO. = 58	SURVIVAL FUNCTION =	.4172325-006	104
CLUSTER NO. = 59	SURVIVAL FUNCTION =	.1415610-005	68
CLUSTER NO. = 60	SURVIVAL FUNCTION =	.1192093-006	92
CLUSTER NO. = 61	SUPVIVAL FUNCTION =	.1293644-002	57
CLUSTER NO. = 62	SUPVIVAL FUNCTION =	.1404643-002	28
CLUSTER NO. = 63	SUPVIVAL FUNCTION =	.1309161	5
CLUSTER NO. = 64	SUPVIVAL FUNCTION =	.4649013-003	43

(CONTINUED)

TABLE 7.VIII (CONCLUDED)

				ISOLATION INDEX
CLUSTER NO. =	65	SURVIVAL FUNCTION =	.1341105-006	42
CLUSTER NO. =	66	SURVIVAL FUNCTION =	.1985213	5
CLUSTER NO. =	67	SURVIVAL FUNCTION =	.8940697-007	229
CLUSTER NO. =	68	SURVIVAL FUNCTION =	.2439254	4
CLUSTER NO. =	69	SURVIVAL FUNCTION =	.4708424-002	14
CLUSTER NO. =	70	SURVIVAL FUNCTION =	.3411472-003	17
CLUSTER NO. =	71	SURVIVAL FUNCTION =	.1043081-006	600
CLUSTER NO. =	72	SURVIVAL FUNCTION =	.1043081-006	190
CLUSTER NO. =	73	SURVIVAL FUNCTION =	.1043081-006	409

TABLE 7.IX

SURVIVAL FUNCTION AND ISOLATION INDEX VALUES FOR CLUSTERS FROM BUOY 3 OF SCENARIO 1

			ISOLATION INDEX
CLUSTER NO. =	1	SURVIVAL FUNCTION = .4869150	14
CLUSTER NO. =	2	SURVIVAL FUNCTION = .6630117	8
CLUSTER NO. =	3	SURVIVAL FUNCTION = .2757257	25
CLUSTER NO. =	4	SURVIVAL FUNCTION = .4042685-004	191
CLUSTER NO. =	5	SURVIVAL FUNCTION = .4864466	14
CLUSTER NO. =	6	SURVIVAL FUNCTION = .2124872	30
CLUSTER NO. =	7	SURVIVAL FUNCTION = .7896106-001	49
CLUSTER NO. =	8	SURVIVAL FUNCTION = .8570225	3
CLUSTER NO. =	9	SURVIVAL FUNCTION = .5674055	11
CLUSTER NO. =	10	SURVIVAL FUNCTION = .1141836	28
CLUSTER NO. =	11	SURVIVAL FUNCTION = .5389733	8
CLUSTER NO. =	12	SURVIVAL FUNCTION = .6948277-002	95
CLUSTER NO. =	13	SURVIVAL FUNCTION = .8568013	3
CLUSTER NO. =	14	SURVIVAL FUNCTION = .1609325-005	262
CLUSTER NO. =	15	SURVIVAL FUNCTION = .1137623	28

(CONTINUED)

TABLE 7.IX (CONTINUED)

			ISOLATION INDEX
CLUSTER NO. =	16	SURVIVAL FUNCTION = .5202636-001	38
CLUSTER NO. =	17	SURVIVAL FUNCTION = .1236796-005	281
CLUSTER NO. =	18	SURVIVAL FUNCTION = .8134705	4
CLUSTER NO. =	19	SURVIVAL FUNCTION = .1868028	13
CLUSTER NO. =	20	SURVIVAL FUNCTION = .2989062	16
CLUSTER NO. =	21	SURVIVAL FUNCTION = .9254659	1
CLUSTER NO. =	22	SURVIVAL FUNCTION = .3250551-001	33
CLUSTER NO. =	23	SURVIVAL FUNCTION = .6443247-002	96
CLUSTER NO. =	24	SURVIVAL FUNCTION = .6607238	8
CLUSTER NO. =	25	SURVIVAL FUNCTION = .3638257	13
CLUSTER NO. =	26	SURVIVAL FUNCTION = .8560889	3
CLUSTER NO. =	27	SURVIVAL FUNCTION = .6270218-001	53
CLUSTER NO. =	28	SURVIVAL FUNCTION = .4153132-002	104
CLUSTER NO. =	29	SURVIVAL FUNCTION = .8346999	1
CLUSTER NO. =	30	SURVIVAL FUNCTION = .1916807	8
CLUSTER NO. =	31	SURVIVAL FUNCTION = .6958670-003	92
CLUSTER NO. =	32	SURVIVAL FUNCTION = .5954256	4
CLUSTER NO. =	33	SURVIVAL FUNCTION = .1608063	35
CLUSTER NO. =	34	SURVIVAL FUNCTION = .2457201-003	79
CLUSTER NO. =	35	SURVIVAL FUNCTION = .6270485	3
CLUSTER NO. =	36	SURVIVAL FUNCTION = .3229120	4
CLUSTER NO. =	37	SURVIVAL FUNCTION = .2197462	29

(CONTINUED)

TABLE 7.IX (CONTINUED)

			VARIABLES CLUSTERED
CLUSTER NO. =	38	SURVIVAL FUNCTION = .2980232-006	207
CLUSTER NO. =	39	SURVIVAL FUNCTION = .1332875-001	14
CLUSTER NO. =	40	SURVIVAL FUNCTION = .4812570	7
CLUSTER NO. =	41	SURVIVAL FUNCTION = .5071954-001	19
CLUSTER NO. =	42	SURVIVAL FUNCTION = .1750886-004	203
CLUSTER NO. =	43	SURVIVAL FUNCTION = .1288421-001	13
CLUSTER NO. =	44	SURVIVAL FUNCTION = .6755485	3
CLUSTER NO. =	45	SURVIVAL FUNCTION = .2078058	10
CLUSTER NO. =	46	SURVIVAL FUNCTION = .2908707-004	193
CLUSTER NO. =	47	SURVIVAL FUNCTION = .7163716-002	62
CLUSTER NO. =	48	SURVIVAL FUNCTION = .9947246-001	6
CLUSTER NO. =	49	SURVIVAL FUNCTION = .3642718-001	18
CLUSTER NO. =	50	SURVIVAL FUNCTION = .1591002	10
CLUSTER NO. =	51	SURVIVAL FUNCTION = .1291833	5
CLUSTER NO. =	52	SURVIVAL FUNCTION = .2543985	3
CLUSTER NO. =	53	SURVIVAL FUNCTION = .9631157-002	22
CLUSTER NO. =	54	SURVIVAL FUNCTION = .5627866-001	6
CLUSTER NO. =	55	SURVIVAL FUNCTION = .1341105-006	692
CLUSTER NO. =	56	SURVIVAL FUNCTION = .4996697-002	20
CLUSTER NO. =	57	SURVIVAL FUNCTION = .5066395-006	136
CLUSTER NO. =	58	SURVIVAL FUNCTION = .1639178-005	167

(CONTINUED)

TABLE 7.IX (CONCLUDED)

			ISOLATION INDEX
CLUSTER NO. =	59	SURVIVAL FUNCTION = .1273446	6
CLUSTER NO. =	60	SURVIVAL FUNCTION = .4329841	2
CLUSTER NO. =	61	SURVIVAL FUNCTION = .1490116-006	36
CLUSTER NO. =	62	SURVIVAL FUNCTION = .1341105-006	110
CLUSTER NO. =	63	SURVIVAL FUNCTION = .1034689-001	55
CLUSTER NO. =	64	SURVIVAL FUNCTION = .3669095	3
CLUSTER NO. =	65	SURVIVAL FUNCTION = .2089143-004	23
CLUSTER NO. =	66	SURVIVAL FUNCTION = .1516947-001	37
CLUSTER NO. =	67	SURVIVAL FUNCTION = .1882628-002	12
CLUSTER NO. =	68	SURVIVAL FUNCTION = .1192093-006	75
CLUSTER NO. =	69	SURVIVAL FUNCTION = .1043081-006	299
CLUSTER NO. =	70	SURVIVAL FUNCTION = .1559601	13
CLUSTER NO. =	71	SURVIVAL FUNCTION = .3577756	12
CLUSTER NO. =	72	SURVIVAL FUNCTION = .2003570	7
CLUSTER NO. =	73	SURVIVAL FUNCTION = .6867349-003	28
CLUSTER NO. =	74	SURVIVAL FUNCTION = .2533197-006	242
CLUSTER NO. =	75	SURVIVAL FUNCTION = .1043081-006	250
CLUSTER NO. =	76	SURVIVAL FUNCTION = .8940697-007	19
CLUSTER NO. =	77	SURVIVAL FUNCTION = .8940697-007	178

7.3.3 Cluster Extraction Results -- Buoy 3 and Buoy 1

A cut-off point of 10^{-4} was used for the survival function to determine which clusters were passed to the extraction module for further processing. For buoy 1, the following cluster numbers survived this test: 35, 38, 43, 44, 45, 48, 53, 54, 55, 58, 59, 60, 65, 67, 71, 72, and 73. For buoy 3, these cluster numbers were chosen: 4, 14, 17, 38, 42, 46, 55, 57, 58, 61, 62, 65, 68, 59, 74, 75, 76, and 77.

The polarization test (see section 3.4) eliminated from further consideration clusters 38, 53, and 55 from buoy 1 and clusters 4, 14, 17, 42, 46, 58, and 74 from buoy 3. The regression test and the exclusion inclusion rules resulted in cluster 67, 71, and 72 being extracted for buoy 1 and clusters 55, 69, and 75 being extracted for buoy 2. Thus for buoy 1, the clustering and extraction processes were able to completely separate the data into three clusters containing only the measurements belonging to a given target. For buoy 3 the same processing techniques produced two of the required data clusters but not the third, selecting cluster 69 rather than 76. From Table 7.VII, it can be seen that cluster 76 contains clusters 69 and 74 as subsets. The polarization test rejected cluster 74 as a potential candidate and the regression test rejected the consistency hypothesis of 69 and 74. Thus, only the last two-thirds of the data for target 2 which was contained in cluster 69 was selected for further consideration.

7.3.4 Initial Guess Results

Table 7.X gives a listing of the clusters extracted for each buoy along with an identification of which

TABLE 7.X

BUOY, CLUSTER, AND TARGET NUMBER
CORRESPONDENCE TABLE

Buoy Number	Cluster Number	Target Number
1	1	3
	2	1
	3	2
2	1	3
	2	1
	3	2
3	1	3
	2	2
	3	1

target is represented by each cluster. Allowing a track to be composed of both two-cluster and three-cluster intersensor combinations, there are then

$$(3*3 + 3*3 + 3*3) + (3*3*3) = 54$$

potential target tracks. Only twelve of these potential inter-sensor combinations correspond to actual tracks. Table 7.XI lists the results from the initial guess algorithm. From this table it can be seen that:

1) Of the twelve cluster combinations corresponding to actual tracks, only 1 was rejected (marked by the asterisk),

2) Of the 42 cluster combinations corresponding to pseudo tracks, 29 (69%) were rejected,

3) Five of the three-cluster combinations were not even considered because at least two of the associated two-cluster combinations had been rejected, and

4) From Figure 5.1, several of the pseudo estimates can be seen to correspond quite closely to the illustrated line-of-sight crossed-bearing intersections.

Once the potential track set has been pruned by the initial guess procedure, the resultant state vector estimates and their associated cluster combinations are then passed to the integer programming module.

TABLE 7.XI

INITIAL GUESS AND CONSTRAINTS RESULTS FOR SCENARIO 1

Buoy 1 (Cluster/ Target)	Buoy 2 (Cluster/ Target)	Buoy 3 (Cluster/ Target)	Accepted or Rejected	X (m)	X (m)	$\dot{X}$ (m)	$\dot{Y}$ (m)
1/3	1/3	--	A	-598.	4386.	2.05	-3.6
1/3	2/1	--	R	-2613.	1715.	-5.51	4.07
1/3	3/2	--	A	423.	5194.	-7.98	6.0
2/1	1/3	--	A	-1999.	-3021.	-3.2	-2.5
2/1	2/1	--	A	-2992.	-571.	5.53	1.17
2/1	3/2	--	R	-3356.	599.	1.53	3.72
3/2	1/3	--	A	-927.	1139.	-.52	-3.24
3/2	2/1	--	A	-2557.	749.3	.52	3.24
*3/2	3/2	--	R	2668.	20.	.74	9.62
1/3	--	1/3	A	-521.	4638.	1.6	-2.87
1/3	--	2/2	A	-1086.	3523.	-3.3	1.57
1/3	--	3/1	R	-4835.	-30.	7.33	7.42
2/1	--	1/3	A	-6789	9906.	.13	-2.05
2/1	--	2/2	A	-5540.	6538.	-6.72	-4.68
2/1	--	3/1	A	-3186.	-197.	6.42	1.45
3/2	--	1/3	R	3382.	1230.	.15	-4.43
3/2	--	2/2	A	2638.	762.	.36	8.68
3/2	--	3/1	R	1980.	115.	3.1	11.7
--	1/3	1/3	A	-392.	4760	1.2	-3.5
--	1/3	2/2	R	23.	7180.	-.54	-3.24
--	1/3	3/1	A	-1232.	368.	7.29	-4.45
--	2/1	1/3	A	-137.	3835.	5.4	1.25
--	2/1	2/2	A	1630.	1916.	-2.62	4.6
--	2/1	3/1	A	-1337.	57.	6.5	.92

(CONTINUED)

Buoy 1 (Cluster/ Target)	Buoy 2 (Cluster/ Target)	Buoy 3 (Cluster/ Target)	Accepted or Rejected	X (m)	Y (m)	$\dot{X}$ (m)	$\dot{Y}$ (m)
--	3/2	1/3	R	1890.	1415.	12.8	12.6
--	3/2	2/2	A	2407.	730.	-.46	9.15
--	3/2	3/1	R	2423.	708.	4.35	10.93
1/3	1/3	1/3	A	-347.	4626	1.59	-3.43
1/3	1/3	2/2	R	-598.	4386.	2.05	-3.6
1/3	1/3	3/1	R	-3487.	903.	6.34	-5.25
1/3	2/1	1/3	R	-432.	4635.	2.01	-1.43
1/3	2/1	2/2	R	-2613.	1715.	-5.51	4.07
1/3	2/1	3/1	R	--	--	--	--
1/3	3/2	1/3	A	-289.	4552.	-1.35	2.18
1/3	3/2	2/2	R	423.	5194.	-7.98	6.0
1/3	3/2	3/1	R	-4243.	-1162.	-2.39	6.68
2/1	1/3	1/3	R	-5198.	6430.	.07	-2.5
2/1	1/3	2/2	R	-1999.	-3021.	-3.2	-2.49
2/1	1/3	3/1	R	-3410.	-180.	5.52	-1.93
2/1	2/1	1/3	R	-6352.	9184.	5.19	1.08
2/1	2/1	2/2	R	-2879.	-752.	2.58	1.26
2/1	2/1	3/1	A	-3180.	-198.	6.33	1.34
2/1	3/2	1/3	R	--	--	--	--
2/1	3/2	2/2	R	-3356.	599.	1.53	3.72
2/1	3/2	3/1	R	--	--	--	--
3/2	1/3	1/3	R	-103.	4184.	.3	-3.56
3/2	1/3	2/2	R	-927.	1139.	-.52	-3.24
3/2	1/3	3/1	R	-1185.	363.	5.28	-3.65
3/2	2/1	1/3	R	1700.	2431.	2.5	.13

(CONTINUED)

TABLE 7.XI (CONCLUDED)

Buoy 1 (Cluster/ Target)	Buoy 2 (Cluster/ Target	Buoy 3 (Cluster/ Target)	Accepted or Rejected	X (x)	Y (m)	$\dot{X}$ (m)	$\dot{Y}$ (m)
3/2	2/1	2/2	A	1817.	1453.	-.18	4.29
3/2	2/1	3/1	A	-422.	171.	3.97	3.62
3/2	3/2	1/3	R	--	--	--	--
3/2	3/2	2/2	A	2668.	20.	.74	9.62
3/2	3/2	3/1	R	--	--	--	--

7.3.5 Integer Programming Results

Figure 7.2 gives the integer programming problem to be solved for this scenario (see Section 6). Note that the number of potential tracks represented in the objective function (i.e., the number of columns) is fewer than the number passed by the initial guess module. This happens because those tracks with positive costs, as computed by the HTA, need not be considered since they would never be part of the solution to this minimization problem. This can significantly reduce the amount of time spent finding an optimum since the solution algorithm spends a lot of time arranging and sorting the columns.

The solution found by the algorithm is

$$X_{13} = X_{14} = X_{16} = 1$$
$$X_1 = X_2 = \ldots X_{12} = X_{15} = S_1 = S_2 = \ldots S_4 = 0.$$

Using Figure 7.2 and Table 7.X this solution is seen to correctly correspond to:

X_{13} - three sensor data for target 3 (1, 1, 1),
X_{14} - three sensor data for target 1 (2, 2, 3),
X_{16} - three sensor data for target 2 (3, 3, 2).

7.3.6 Conclusion

The MTTA correctly sorted the raw sensor data, correctly eliminated about 70% of the possible pseudo-tracks, and then identified the actual tracks. For this scenario, Figure 7.1 illustrates the estimated tracks generated by

$$\min x = -404x_1 - 207x_2 - 62x_3 - 297x_4 - 244x_5 - 161x_6 - 324x_7 - 347x_8 - 398x_9 - 320x_{10} - 211x_{11} - 184x_{12} - 559x_{13} - 408x_{14} - 17x_{15} - 412x_{16}$$

Subject to:

$$\begin{array}{ll} & \text{Cluster 1} \\ \text{Buoy 1} & \text{Cluster 2} \\ & \text{Cluster 3} \\ & \text{Cluster 1} \\ \text{Buoy 2} & \text{Cluster 2} \\ & \text{Cluster 3} \\ & \text{Cluster 1} \\ \text{Buoy 3} & \text{Cluster 2} \\ & \text{Cluster 3} \end{array} \left\{ \begin{bmatrix} 1&1&0&0&0&0&1&0&0&0&0&0&1&0&0&0 \\ 0&0&1&1&0&0&0&1&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&1&1&0&0&0&0&0&0&0&1&1 \\ 1&0&1&0&1&0&0&0&1&1&0&0&1&0&0&0 \\ 0&0&0&1&0&1&0&0&0&0&1&1&0&1&1&0 \\ 0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&1 \\ 0&0&0&0&0&0&1&0&1&0&1&0&1&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1 \\ 0&0&0&0&0&0&0&1&0&1&0&1&0&1&1&0 \end{bmatrix} \bullet \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \\ x_{11} \\ x_{12} \\ x_{13} \\ x_{14} \\ x_{15} \\ x_{16} \end{bmatrix} + \begin{bmatrix} 1&0&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&0&1 \end{bmatrix} \bullet \begin{bmatrix} S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \\ S_6 \\ S_7 \\ S_8 \\ S_9 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

Figure 7.2. Integer Programming Equation for Scenario 1

the MTTA and the actual trajectories followed by the targets. The "tic" marks in each true trajectory line represent two minute intervals. All the estimated trajectories have come fairly close to the actual tracks before the first tic mark. The early severe fluctuations are a direct result of the batch initializer's updating scheme. After an initial state vector estimate is obtained for t=0, a Kalman filter is used to update the state to the current time. The filter is applied with a large initial covariance matrix which allows it to search over a wide area to update the state estimate. Typically, early in the track, only one buoy supplies the first three or four measurements, and this limited observation geometry coupled with the large covariance matrix results in state updates which are substantially different from the actual ones. In all cases, the initial state vector estimates at t=0, and the updated estimates passed to the sequential filter after initialization were quite close to the true values. In any event, the tracking accuracy is a result of HTA characteristics, not MTTA performance, given that MTTA has correctly solved the intersensor data matching problem.

The large fluctuations observed for target 2 in Figure 7.1 are due to the fact that for buoy 3, the cluster extraction algorithm did not put the first third of the T_2 (target 2) data set with the last two-thirds. Thus, buoy 2 by itself supplied the first few measurements for target 2. This failure of the extraction algorithm can be traced to the linearity assumption of the testing mechanism; target 2 went through CPA relative to buoy 3 resulting in severe nonlinear frequency and bearing changes. The first part of the T_2 data set contained the Doppler compression data up through CPA, the last part consisted of the Doppler expansion data for T_2 that was found after CPA. The cluster extraction algorithm rightly viewed these two data streams as different processes, and concluded that they did not go together. It is felt that this

problem can be remedied by including quadratic terms in the extraction's regression model to compensate for the nonlinear frequency and bearing changes encountered during CPA.

7.4 Three Target Scenario with Bad Geometry Low Noise, 0 dB threshold in 1 Hz Band

Figure 7.3. shows the scenario discussed in this section. Table 7.II lists the true initial and final conditions for each of the targets in this scenario. The trajectories simulated are the same as those modeled for the scenario discussed in section 7.3 except that target 1 has a forty-five degree course heading instead of the fifteen degree heading that was used previously.

From Figure 7.3, it can be seen that the MTTA provides reasonable tracks for targets 1 and 2, but no track at all is generated for target 3. Tables 7.XII and 7.XIII list the data and the clusters, respectively, selected for buoy 1, while Tables 7.XIV and 7.XV list the same information for buoy 3. Buoy 2 sees all targets perfectly. From Tables 7.XII and 7.XIII, it can be seen that buoy 1 sees target 3 perfectly, however, it gets targets 1 and 2 confused and splices the last ten points from target 1 onto the target 2 observation set. A similar phenomena occurs for buoy 3 when the first 10 observations from target 2 are interspersed among those of target 1. Buoy 3 also sees target 3 perfectly. Looking at the data, it is clear that the measurements become quite similar during these times and the clustering algorithm simply cannot distinguish between the right and wrong data sets. However, when the power levels of the various signals are examined, a clear difference is exposed. The choice of attribute variables used in this study ignores signal power and consequently

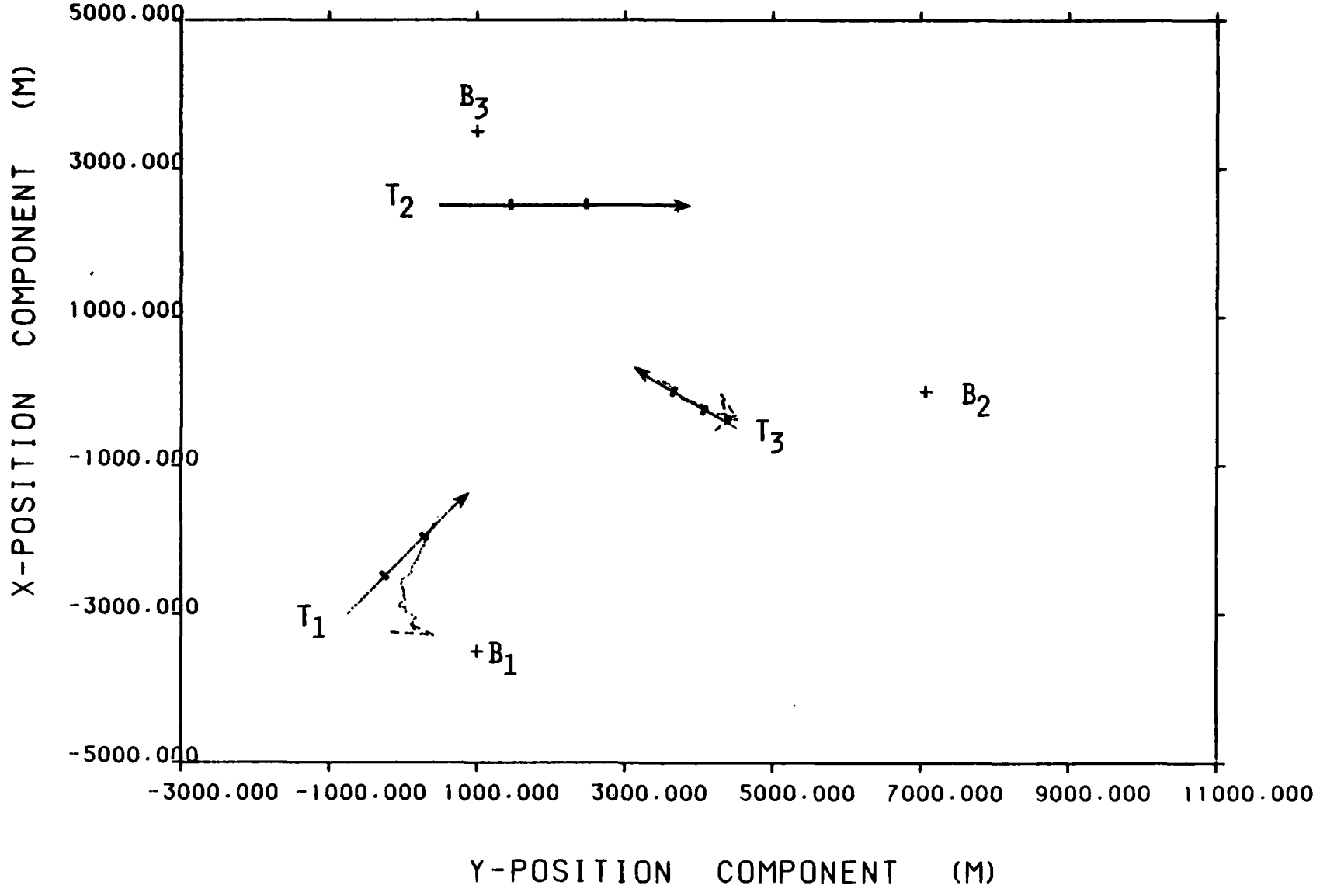


Figure 7.3. Plot of the MTTA's Estimated Tracks vs. the True Tracks for the 3-Target Strong Signal, and Poor Geometry Scenario

TABLE 7.XII

SIMULATED MULTI-TARGET DIFAR DATA FOR
BUOY 1 OF SCENARIO 2

SAMPLE NO.	TIME	FREQUENCY	COS(β)	SIN (β)	TARGET NO.
1	[illegible]	[illegible]	.247612	-.969695	1
2	[illegible]	[illegible]	[illegible]	[illegible]	1
3	[illegible]	[illegible]	[illegible]	[illegible]	3
4	[illegible]	[illegible]	[illegible]	[illegible]	1
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	3
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	1
7	[illegible]	[illegible]	[illegible]	[illegible]	2
8	[illegible]	[illegible]	[illegible]	[illegible]	3
9	[illegible]	[illegible]	[illegible]	[illegible]	1
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	1
11	[illegible]	[illegible]	[illegible]	[illegible]	2
12	[illegible]	[illegible]	[illegible]	[illegible]	3
13	[illegible]	[illegible]	[illegible]	[illegible]	1
14	[illegible]	[illegible]	[illegible]	[illegible]	3
15	[illegible]	[illegible]	[illegible]	[illegible]	1
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	3
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	1
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	2
19	[illegible]	[illegible]	.741749	[illegible]	3
20	[illegible]	[illegible]	[illegible]	[illegible]	1
21	[illegible]	[illegible]	[illegible]	[illegible]	2
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	3
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	1
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	2
26	[illegible]	[illegible]	[illegible]	[illegible]	1
27	[illegible]	[illegible]	[illegible]	[illegible]	2
28	[illegible]	[illegible]	[illegible]	[illegible]	3
29	[illegible]	[illegible]	[illegible]	[illegible]	1
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	2
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	1
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	2
33	[illegible]	[illegible]	.734941	[illegible]	1
34	[illegible]	[illegible]	[illegible]	[illegible]	2
35	[illegible]	[illegible]	.671464	[illegible]	1

(CONTINUED)

TABLE 7.XII (CONCLUDED)

SAMPLE NO.	TIME	FREQUENCY	COS(β)	SIN(β)	TARGET NO.
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	2
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	3
[illegible]	[illegible]	[illegible]	[illegible]	-[illegible]	1
44	[illegible]	[illegible]	.777987	[illegible]	3
[illegible]	[illegible]	[illegible]	[illegible]	-[illegible]	1
[illegible]	[illegible]	[illegible]	[illegible]	-[illegible]	2
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	3
[illegible]	[illegible]	[illegible]	[illegible]	-[illegible]	1
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	2
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	3
[illegible]	[illegible]	[illegible]	[illegible]	-[illegible]	1
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	2
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	3
[illegible]	[illegible]	[illegible]	[illegible]	-[illegible]	1
[illegible]	[illegible]	[illegible]	[illegible]	-[illegible]	1
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	2
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	3
[illegible]	[illegible]	[illegible]	[illegible]	-[illegible]	1
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	3
[illegible]	[illegible]	[illegible]	[illegible]	-[illegible]	1
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	3
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	3
[illegible]	[illegible]	[illegible]	[illegible]	-[illegible]	1
64	[illegible]	[illegible]	[illegible]	-[illegible]	1
[illegible]	[illegible]	[illegible]	[illegible]	-[illegible]	1
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	2
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	3
[illegible]	[illegible]	[illegible]	[illegible]	-[illegible]	1
[illegible]	[illegible]	[illegible]	[illegible]	-[illegible]	1
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	2
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	3
[illegible]	[illegible]	[illegible]	[illegible]	-[illegible]	1
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	2
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	3
[illegible]	[illegible]	[illegible]	[illegible]	-[illegible]	1
[illegible]	[illegible]	[illegible]	[illegible]	.411449	3
77	[illegible]	[illegible]	[illegible]	-[illegible]	1
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	3

TABLE 7.XIII

CLUSTERING OUTPUT FOR BUOY 1
OF SCENARIO 1

CLUSTER	NODE	SIZE	OBJECTS CLUSTERED
1	1	2	75 77
2	2	2	29 32
3	3	2	23 25
4	4	2	48 51
5	5	3	72 75 77
6	6	2	50 53
7	7	2	9 10
8	8	2	3 5
9	9	3	47 50 53
10	10	2	15 17
11	11	3	20 23 25
12	12	2	18 21
13	13	3	48 51 54
14	14	2	28 34
15	15	2	12 14
16	17	4	45 48 51 54
17	18	4	42 47 50 53
18	19	2	22 27
19	20	3	9 10 13

(CONTINUED)

TABLE 7.XIII (CONTINUED)

CLUSTER	NODE	SIZE	OBJECTS CLUSTERED
20	23	2	49 52
21	24	2	1 2
22	26	5	62 64 67 70 73
23	29	4	68 72 75 77
24	30	2	58 63
25	31	2	73 78
26	34	3	28 31 34
27	35	4	3 5 12 14
28	36	5	68 69 72 75 77
29	37	3	22 27 30
30	39	6	42 44 47 50 53 57
31	40	2	82 87
32	43	6	65 68 69 72 75 77
33	46	2	36 40
34	47	2	35 41
35	48	4	19 22 27 30
36	49	3	11 18 21
37	51	4	6 9 10 13
38	52	3	61 62 67

(CONTINUED)

TABLE 7.XIII (CONTINUED)

CLUSTER	NODE	SIZE	OBJECTS CLUSTERED
39	53	2	70 73
40	54	5	61 62 67 71 78
41	61	2	55 60
42	64	3	24 35 41
43	65	6	61 62 67 71 74 78
44	66	7	33 42 44 47 50 53 57
45	67	4	56 58 60 63
46	69	5	43 45 48 51 54
47	71	5	24 29 32 35 41
48	72	5	4 6 9 10 13
49	75	5	15 17 20 23 25
50	77	8	3 5 12 14 19 22 27 30
51	83	3	36 38 40
52	84	6	24 29 32 35 39 41
53	90	3	49 52 56
54	92	7	24 26 29 32 35 39 41
55	93	3	66 70 73
56	95	9	3 5 8 12 14 19 22 27 30
57	96	10	3 5 8 12 14 16 19 22 27 30
58	115	7	59 61 62 67 71 74 78
59	116	6	26 31 34 36 38 40
60	119	17	3 5 8 12 14 16 19 22 27 30 33 42 44 47 50 53 57

(CONTINUED)

TABLE 7.XIII (CONCLUDED)

CLUSTER	NODE	SIZE	OBJECTS CLUSTERED
61	147	4	7 11 18 21
62	151	5	55 58 60 63 64
63	173	8	59 61 62 67 71 74 76 78
64	177	6	46 55 58 60 63 64
65	188	7	1 2 4 6 9 10 13
66	190	11	29 31 34 36 38 40 43 45 48 51 54
67	193	12	46 55 58 60 63 64 65 68 69 72 75 77
68	196	16	15 17 20 23 25 28 31 34 36 38 40 43 45 48 51 54
69	198	11	7 11 18 21 24 26 29 32 35 39 41
70	199	25	3 5 8 12 14 16 19 22 27 30 33 42 44 47 50 53 57 59 61 62 67 71 74 76 78
71	204	12	7 11 18 21 24 26 29 32 35 37 39 41
72	205	23	1 2 4 6 9 10 13 15 17 20 23 25 28 31 34 36 38 40 43 45 48 51 54
73	208	24	7 11 18 21 24 26 29 32 35 37 39 41 46 55 58 60 63 64 65 68 69 72 75 77
74	225	27	7 11 18 21 24 26 29 32 35 37 39 41 46 49 52 55 56 58 60 63 64 65 68 69 72 75 77
75	229	50	1 2 4 6 7 9 10 11 13 15 17 18 20 21 23 24 25 26 28 29 31 32 34 35 36 37 38 39 40 41 43 45 46 48 49 51 52 54 55 56 58 60 63 64 65 68 69 72 75 77
76	245	53	1 2 4 6 7 9 10 11 13 15 17 18 20 21 23 24 25 26 28 29 31 32 34 35 36 37 38 39 40 41 43 45 46 48 49 51 52 54 55 56 58 60 63 64 65 66 68 69 70 72 73 75 77
77	437	78	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78

TABLE 7.XIV

SIMULATED MULTI-TARGET DIFAR DATA FOR
BUOY 3 OF SCENARIO 2

SAMPLE NO.	TIME	FREQUENCY	COS(β)	SIN(β)	TARGET NO.
1	[illegible]	150.407	-.691746	-.472479	2
2	[illegible]	150.500	-.799414	-.[illegible]	1
3	[illegible]	150.700	-.736695	-.[illegible]	2
4	[illegible]	150.[illegible]	-.[illegible]	-.[illegible]	1
5	[illegible]	150.[illegible]	-.[illegible]	-.[illegible]	2
6	[illegible]	150.[illegible]	-.[illegible]	.[illegible]	3
7	[illegible]	150.[illegible]	-.[illegible]	-.[illegible]	2
8	[illegible]	150.[illegible]	-.[illegible]	-.[illegible]	1
9	[illegible]	150.[illegible]	-.[illegible]	-.[illegible]-001	2
10	[illegible]	150.[illegible]	-.[illegible]	.[illegible]	3
11	[illegible]	150.[illegible]	-.[illegible]	-.[illegible]	1
12	[illegible]	150.[illegible]	-.[illegible]	-.[illegible]-001	2
13	[illegible]	150.[illegible]	-.[illegible]	.[illegible]	3
14	[illegible]	150.[illegible]	-.[illegible]	.[illegible]	1
15	[illegible]	150.[illegible]	-.[illegible]	-.[illegible]-001	2
16	[illegible]	150.[illegible]	-.[illegible]	-.[illegible]	3
17	[illegible]	150.[illegible]	-.[illegible]	.[illegible]	1
18	[illegible]	150.[illegible]	-.[illegible]	.[illegible]-001	2
19	[illegible]	150.[illegible]	-.[illegible]	.[illegible]	3
20	[illegible]	150.[illegible]	-.[illegible]	-.[illegible]	1
21	[illegible]	150.[illegible]	-.[illegible]	-.[illegible]	2
22	[illegible]	150.[illegible]	-.[illegible]	-.[illegible]-001	1
23	[illegible]	150.[illegible]	-.[illegible]	.[illegible]	2
24	[illegible]	150.[illegible]	-.[illegible]	-.[illegible]-001	1
25	[illegible]	150.[illegible]	-.[illegible]	.[illegible]	2
26	[illegible]	150.[illegible]	-.[illegible]	.[illegible]	3
27	[illegible]	150.[illegible]	-.[illegible]	-.[illegible]-001	1
28	[illegible]	150.[illegible]	-.[illegible]	.[illegible]	2
29	[illegible]	150.[illegible]	-.[illegible]	-.[illegible]	1
30	[illegible]	150.[illegible]	-.[illegible]	.[illegible]	2
31	[illegible]	150.[illegible]	-.[illegible]	.[illegible]	3
32	[illegible]	150.[illegible]	-.[illegible]	-.[illegible]-001	1
33	[illegible]	149.500	-.777737	.[illegible]	2
34	[illegible]	150.[illegible]	-.731443	.[illegible]	3
35	[illegible]	150.[illegible]	-.[illegible]	.[illegible]	2
36	[illegible]	150.[illegible]	-.[illegible]	.[illegible]	3
37	[illegible]	150.[illegible]	-.[illegible]	.[illegible]-001	1
38	[illegible]	149.[illegible]	-.[illegible]	.[illegible]	2
39	[illegible]	150.[illegible]	-.[illegible]	.[illegible]	3
40	[illegible]	150.[illegible]	-.[illegible]	-.[illegible]	1

(CONTINUED)

TABLE 7.XIV (CONCLUDED)

SAMPLE NO.	TIME	FREQUENCY	COS(β)	SIN(β)	TARGET NO.
42	1[illegible]	140.400	-.[illegible]	.7[illegible]	2
43	1[illegible]	160.700	-.71[illegible]	.6[illegible]	3
43	1[illegible]	140.700	-.[illegible]	.7[illegible]	2
44	145.700	160.700	-.712439	.701744	3
45	205.100	180.400	-.991726	-.1[illegible]	1
46	205.100	140.700	-.6[illegible]	.7074[illegible]	2
47	205.700	160.700	-.[illegible]	.6[illegible]	3
48	205.[illegible]	140.700	-.[illegible]	.7[illegible]	2
49	205.[illegible]	180.400	-.[illegible]	-.3[illegible]	1
50	225.100	140.700	-.[illegible]741	.6[illegible]	2
51	225.700	160.700	-.[illegible]	.6[illegible]	3
52	245.100	140.700	-.[illegible]99	.[illegible]	2
53	245.700	160.700	-.[illegible]	.7[illegible]	3
54	245.[illegible]	180.400	-.[illegible]	-.1[illegible]	1
55	245.[illegible]	160.700	-.7[illegible]	.6[illegible]	3
56	265.100	140.700	-.[illegible]151	.6[illegible]	2
57	265.700	160.700	-.7[illegible]	.6[illegible]	3
58	265.700	180.400	-.[illegible]	-.[illegible]	1
59	[illegible]	160.700	-.[illegible]	.6[illegible]	3
60	[illegible]	180.400	-.[illegible]	-.1[illegible]	1
61	[illegible]	140.700	-.[illegible]	.[illegible]	2
62	285.100	140.700	-.[illegible]	.[illegible]61	2
63	285.700	160.700	-.7[illegible]	.6[illegible]	3
64	285.[illegible]	180.400	-.[illegible]	.1[illegible]	1
65	[illegible]	140.700	-.[illegible]	.6[illegible]	2
66	[illegible]	160.700	-.7[illegible]	.6[illegible]	3
67	[illegible]	180.400	-.[illegible]	-.[illegible]	1
68	305.100	140.700	-.[illegible]	.[illegible]	2
69	305.700	160.700	-.741[illegible]	.6[illegible]	3
70	325.[illegible]	180.400	-.[illegible]	.1[illegible]	1
71	[illegible]	140.700	-.[illegible]	.6[illegible]	2
72	[illegible]	160.700	-.7[illegible]	.7[illegible]	3
73	[illegible]	160.700	-.[illegible]	.6[illegible]	3
74	345.[illegible]	180.400	-.[illegible]	-.3[illegible]	1
75	345.[illegible]	140.700	-.[illegible]	.[illegible]	2
76	345.700	160.700	-.777[illegible]	.6[illegible]	3
77	345.700	180.400	-.[illegible]	-.1[illegible]	1
7[illegible]	[illegible]	140.[illegible]	-.[illegible]	.[illegible]	2
7[illegible]	[illegible]	160.700	-.[illegible]	.6[illegible]	3
[illegible]	[illegible]	180.400	-.[illegible]	-.1[illegible]	1
[illegible]	365.[illegible]	140.700	-.[illegible]	.[illegible]	2
[illegible]	385.700	160.700	-.[illegible]	.6[illegible]	3

TABLE 7.XV

CLUSTERING OUTPUT FOR BUOY 3
OF SCENARIO 2

CLUSTER	NODE	SIZE	OBJECTS CLUSTERED
1	1	2	46 49
2	2	2	68 71
3	3	2	42 44
4	4	2	35 38
5	5	2	66 69
6	6	2	30 32
7	7	2	24 27
8	8	2	59 63
9	9	2	61 65
10	10	3	68 71 75
11	11	3	55 59 63
12	12	3	22 24 27
13	13	2	10 16
14	14	2	30 33
15	15	3	43 46 48
16	16	3	10 13 16
17	17	2	20 21
18	18	5	55 59 63 66 69
19	19	3	56 61 65
20	22	2	16 17
21	24	3	11 14 17

(CONTINUED)

TABLE 7.XV (CONTINUED)

CLUSTER	NODE	SIZE	OBJECTS CLUSTERED
22	25	2	29 [illegible]
23	27	3	34 42 44
24	28	2	53 57
25	29	2	47 51
26	30	7	53 55 57 59 63 66 69
27	31	8	53 55 57 59 63 66 69 76
28	33	2	76 77
29	34	5	50 52 56 61 65
30	35	4	22 24 27 32
31	36	2	31 39
32	37	4	68 71 75 78
33	39	2	58 67
34	40	6	50 52 56 61 62 65
35	41	11	34 42 44 53 55 57 59 63 66 69 76
36	42	4	8 11 14 17
37	43	9	43 46 48 50 52 56 61 62 65
38	45	12	34 42 44 53 55 57 59 63 66 69 72 76
39	53	4	10 13 16 19
40	55	4	58 67 70 77
41	56	2	7 9
42	57	2	46 54
43	59	13	34 42 44 53 55 57 59 63 66 69 72 76 82
44	61	5	10 13 16 19 26

(CONTINUED)

TABLE 7.XV (CONTINUED)

CLUSTER	NODE	SIZE	OBJECTS CLUSTERED
45	64	8	4 6 10 13 16 19 26 36
46	66	5	2 8 11 14 17
47	68	3	55 54 60
48	69	2	12 15
49	73	15	31 34 39 42 44 53 55 57 59 63 66 69 72 76 82
50	76	7	6 10 13 16 19 26 36
51	78	5	58 67 70 74 77
52	79	2	29 40
53	81	16	31 34 39 42 44 53 55 57 59 63 66 69 72 73 76 82
54	82	6	22 24 27 29 32 40
55	83	6	2 4 8 11 14 17
56	88	18	31 34 39 42 44 47 51 53 55 57 59 63 66 69 72 73 76 82
57	90	19	31 34 39 42 44 47 51 53 55 57 59 63 66 69 72 73 76 79 82
58	95	7	22 24 27 29 32 37 40
59	96	26	6 10 13 16 19 26 31 34 36 39 42 44 47 51 53 55 57 59 63 66 69 72 73 76 79 82
60	101	8	45 54 58 60 67 70 74 77
61	120	9	45 49 54 58 60 67 70 74 77
62	127	5	68 71 75 78 81
63	131	3	21 25 28
64	145	10	45 49 54 58 60 67 70 74 77 80
65	179	15	2 4 8 11 14 17 22 24 27 29 32 37 40
66	200	3	35 38 41

(CONTINUED)

TABLE 7.XV (CONCLUDED)

CLUSTER	NODE	SIZE	OBJECTS CLUSTERED
67	255	12	35 38 41 43 46 48 50 52 56 61 62 65
68	257	17	35 38 41 43 46 48 50 52 56 61 62 65 68 71 75 78 81
69	264	4	7 9 12 15
70	278	19	30 33 35 38 41 43 46 48 50 52 56 61 62 65 68 71 75 78 81
71	288	11	45 49 54 58 60 64 67 70 74 77 80
72	291	22	23 25 28 30 33 35 38 41 43 46 48 50 52 56 61 62 65 68 71 75 78 81
73	299	5	5 7 9 12 15
74	300	3	18 20 21
75	301	24	2 4 8 11 14 17 22 24 27 29 32 37 40 45 49 54 58 60 64 67 70 74 77 80
76	310	8	8 7 9 12 15 18 20 21
77	317	9	3 5 7 9 12 15 18 20 21
78	321	25	1 2 4 8 11 14 17 22 24 27 29 32 37 40 45 49 54 58 60 64 67 70 74 77 80
79	349	34	1 2 3 4 5 7 8 9 11 12 14 15 17 18 20 21 22 24 27 29 32 37 40 45 49 54 58 60 64 67 70 74 77 80
80	525	60	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 24 26 27 29 31 32 34 36 37 39 40 42 44 45 47 49 51 53 54 55 57 58 59 60 63 64 66 67 69 70 72 73 74 76 77 79 80 82
81	635	82	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82

treats all measurements as if they were received at a common power level. However, the propagation losses for a near target should be much smaller than those for a distant target, so clearly the received signal powers should vary according to the proximity of the target to the receiver. Other studies currently under way at Tracor indicate the power levels can be clustered by applying an intensity index to the power level and then clustering with this index. It is felt that application of a similar methodology to DIFAR data could improve the ability of the algorithm to separate data from two targets that are close in bearing and frequency, but differ in power levels.

7.5 Three Target Scenario with Good Geometry, Moderate Noise, No Threshold

Geometrically, this scenario is the same as the one discussed in section 7.3. The difference between the two scenarios is in the simulated source power levels. For the first scenario, there was an 80 dB source level, but for this scenario, the source level was reduced to 77 dB. Figure 7.4 provides graphical comparison of the target true tracks versus the MTTA's estimates of these tracks.

This particular scenario was simulated to test the following two data processing capabilities of the MTTA:

1) The ability of the clustering and automatic cluster extraction algorithms to separate signals from random, background noise, and

2) The ability of the MTTA's HTA and IP to pick the correct targets and to adequately track them with noisy data gathered from moderate strength target signals.

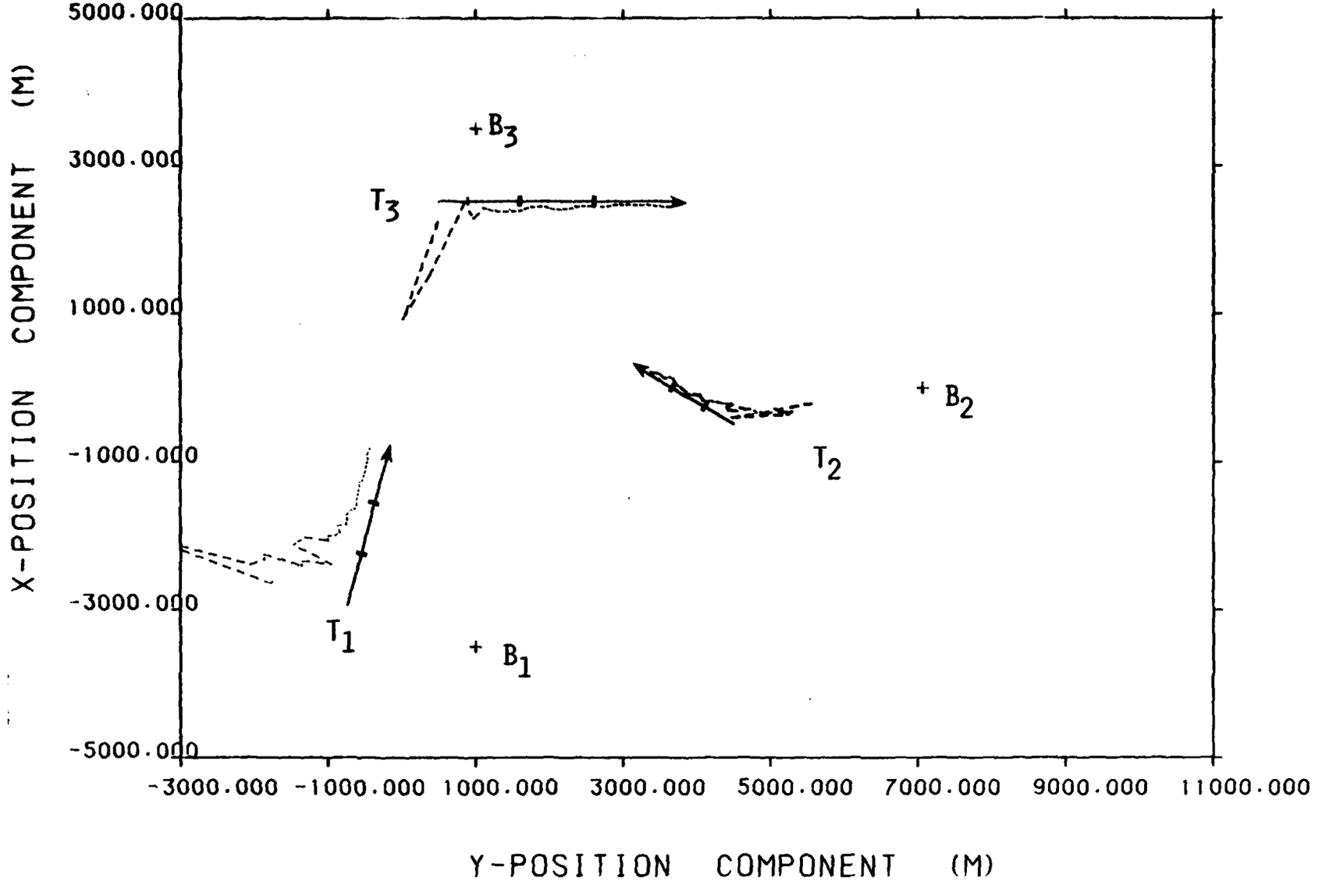


Figure 7.4. Plot of the MTTA's Estimated Tracks vs. the True Tracks for the 3-Target, Weak Signal and Good Geometry Scenario

As stated previously, the 3 dB drop in the simulated target signal strength resulted in an approximate 30% decrease in the range over which the target could be readily detected. For this scenario, each of the targets could be well detected by one (i.e, the closest) sonobuoy, each could be fairly well detected by a second sonobuoy, but the targets could just be marginally detected by the sonobuoy farthest from the signal source. Particularly if a 0 dB threshold was used for detection by the sonobuoys, each target would be nearly unobservable for one sonobuoy in the pattern. To compensate for any possible loss of data, it was decided to drop the threshold criterion and to let the clustering algorithm sort the true signals from the random noise.

Overall, the MTTA performed very well in processing the noisy data. For the most part, random noise was eliminated from the true signals by the clustering and automatic extraction processes. Most of the signal clusters contained a couple of noise points that were similar to the true measurement set, but that did not really belong with the true data. However, the DIFAR simulator computes a measurement standard deviation that is a function of the estimated SNR of the received signal. For the spurious noise samples, the estimated SNR values were very small (i.e., less than 0 dB), so a large variance was computed for the frequency and bearing estimates. The HTA uses the measurement variance to weight the measurement before generating a least squares update of the target track. When large variances are provided with measurement data, the tracker essentially weights the data so

that they can cause little or no change in the track update. The net effect is that when only a few poor measurements are provided to the HTA with variances that indicate the data should carry minimal weight, the HTA can continue to track the target without seriously degrading the track updates. The results from this scenario, presented in Figure 7.4, illustrate this point well.

From the figure, it can be seen that after some fluctuations during the track initialization phase, excellent track estimates were obtained for targets 2 and 3. For target 1, somewhat poorer estimates of the trajectory were obtained. Part of the problem with this estimate is that target 1 is approaching buoy 3 nearly head-on from a very long distance. Because the target is heading nearly directly at the buoy, very little bearing change would be observed even for a well received signal. Since the signal is only of moderate strength and is fairly distant from the sonobuoy receiver, the propagation losses are substantial enough to cause the signal to be poorly received. Since the signal is so poorly detected, serious fluctuations result in the bearing measurements which may in turn appear to the tracker to be a significant bearing rate. Hence, even though buoy 3 does observe target 1 and the clustering and cluster extraction processes adequately sort the data, the measurements from buoy 3 are still of such poor quality that they adversely affect the tracking solutions from the MTTA for target 1. Nonetheless, these tracking solutions for all the targets would probably be adequate for real world encounters.

In conclusion, the MTTA seems to have passed the two tests that this scenario was designed to examine. First,

it did an adequate job of separating noise from true signals. Second, the MTTA did an adequate job of tracking the targets with the poorer quality data generated for this examination. However, once again, bad observational geometry adversely affected the MTTA's estimates, perhaps even more so than the degraded quality of the data.

7.6 Two Target Scenario with Good Geometry, Low Noise, and 0 dB in a 1 Hz Band Threshold

The last case considered by this study was a relatively simple, two target scenario. The same tri-tac buoy pattern was used to observe the targets, and favorable geometries and low background noise levels were used to simulate this scenario. Figure 7.5 contains a plot of the true target trajectories versus the MTTA's estimates of the tracks for this two-target scenario. While the tracking solutions presented in the plots show good results, one interesting point was encountered in generating these results that needs further discussion. For target 2, the MTTA picked only a 2-cluster, intersensor combination to track the target rather than the correct 3-cluster combination that would have been preferred. The remaining third cluster was left unused, and the MTTA did a good job of tracking the target without the data from this cluster. One must question why the third cluster describing target 2 was excluded, however, and so the results were more closely examined to find the cause of this exclusion.

For this scenario, each sonobuoy should have generated two different data clusters, one cluster to represent

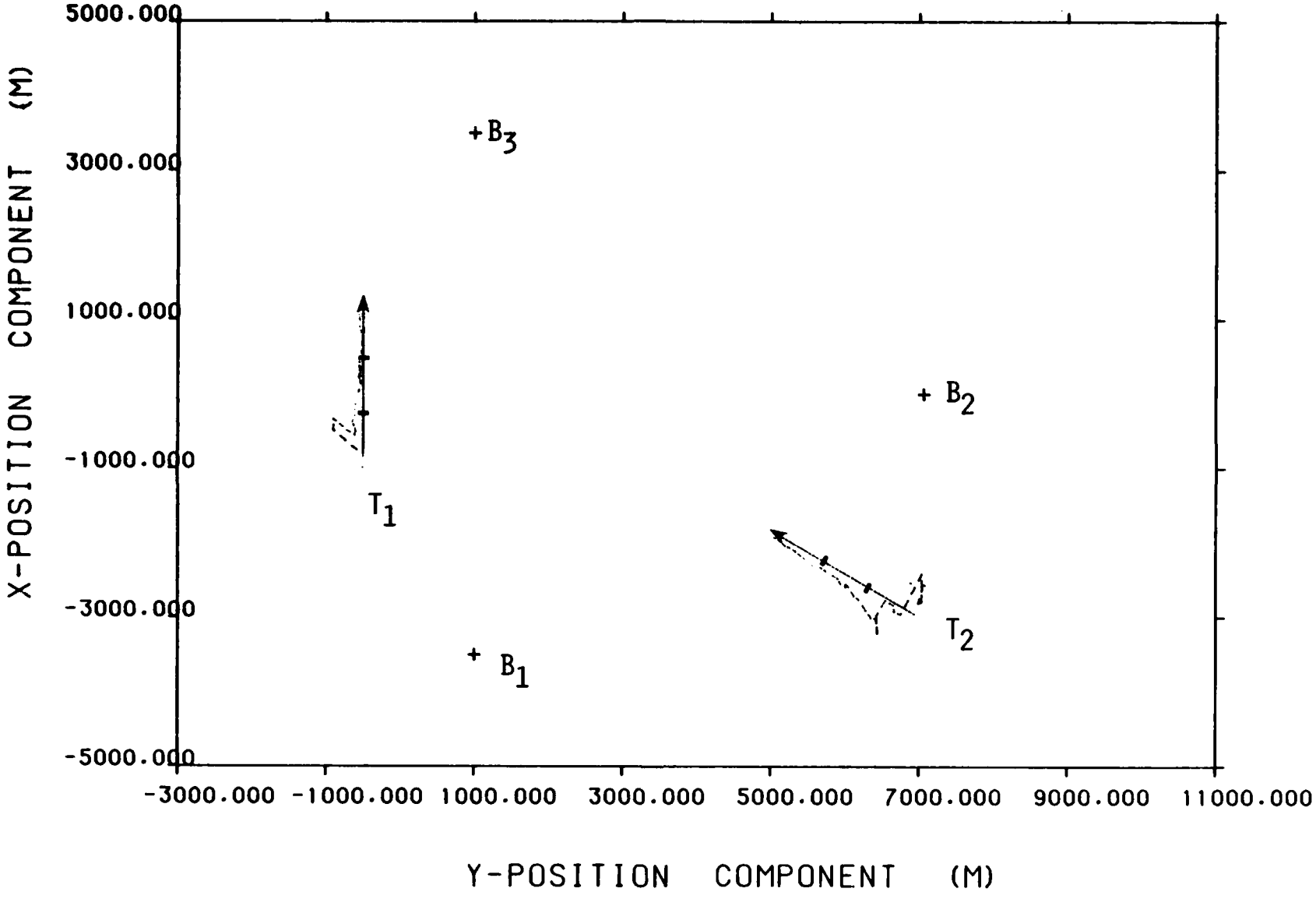


Figure 7.5. Plot of the MTTA's Estimated Tracks vs. the True Tracks for the 2-Target, Strong Signal and Good Geometry Scenario

each of the two targets. The clustering and extraction algorithms did a perfect job of sorting the data into individual target data sets for each of the sonobuoys. Next, the data clusters were passed to the initial guess (IG) algorithm to eliminate unlikely cluster combinations and to keep the rest. Since the data were simulated for this scenario, it was known that the correct cluster combination for target 1 should have been (1, 2, 1), meaning the first cluster from sensor 1, the second cluster from sensor 2, and the first cluster from sensor 3 should be combined to track target 1. For target 2, the correct three-cluster combination should have been (2, 1, 2). The IG correctly passed all of the correct 2-cluster combinations and the single 3-cluster combination for target 1. It also passed all of the correct 2-cluster combinations for target 2, but the IG rejected the one 3-cluster combination that should have been passed to represent target 2. Apparently, the 3-cluster statistical test found too much statistical difference between this estimate and the 2-sensor combinations that make-up this cluster triplet. Because of this rejection by the IG, the preferred 3-cluster combination was rejected before the MTTA's IP could even consider it. Without this cluster triplet, the IP could only pick the single 2-sensor combination that minimized the cost function and leave the remaining cluster unaccounted for. This proved to be exactly what happened when the MTTA picked optimal solutions of (1, 2, 1) for target 1 and (2, 1, 0) for target 2 and left the second cluster for sensor 3 unused.

In conclusion, the MTTA once again performed well in tracking the two targets in this scenario as can be seen in Figure 7.5. Unfortunately, the MTTA's IG module prematurely eliminated the cluster combination that would have been preferred to track target 2. This one case is the sole

example in all the multi-target scenarios studied where the IG failed to pass the preferred 3-sensor solution. It should be remembered, however, that any statistically based test is going to make a wrong decision part of the time, even though hypothesis tests are constructed to minimize type I errors. The result is that, occasionally, good initial guess estimates are going to be eliminated purely by chance.

8.0 SUMMARY AND RECOMMENDATIONS

8.1 Project Summary

The scenarios discussed in Section 7.0 have shown that in its present form the MTTA possesses the ability, with no required operator interventions, to properly sort passive sonobuoy data into individual target data sets, correlate those data sets across sensors, and select the correct target track scenario from the hundreds of potential tracks. Additionally, in its current configuration, the algorithm has performed in near real time and, with minimal processing alterations, could operate substantially faster than real time.

Simulation results showed the importance that scenario geometry played in correctly identifying appropriate tracks. Under certain circumstances, the clustering algorithm can become confused over target data and group together points from different targets having substantially different power levels. However, even in the worst case studied, two of the three possible tracks were estimated fairly well. Results based on good geometry but moderate to high background noise levels indicate that the MTTA is not as sensitive to noise levels as it is to bad geometry. This implies that the clustering and extraction algorithms are confused less by measurements corrupted by noise than by similar measurements coming from different targets. Remedies to these problems will be discussed in the next section.

8.2 Recommendations for Future Work

The recommendations for future work contained in this report fall into two natural categories: enhancements to

correct the deficiencies observed in the MTTA under the current contract, and extensions required to make the MTTA able to perform for long periods of time.

Under the current contract, the following deficiencies were detected in the MTTA:

1) Confusion of targets under certain geometries, even when the power levels of the signals were different, and

2) Incorrect cluster extraction when targets were undergoing CPA.

To remedy the first problem, it is felt that the estimated power level can be converted to an intensity index, and this index can then be used as an added attribute to sort the data by clustering techniques. The reason for converting the power level to index form is that typically the raw power-level estimates vary too much to be useful, but indexing such as is done for visual displays has proven to be useful in other studies. The second problem can be corrected by adopting a curvilinear model instead of a linear one for the extraction regression equations. This should permit the model to more closely approximate the nonlinear measurement time histories observed during CPA.

Proposed extensions to the MTTA are essentially based on the nature of the multi-target tracking problem itself. Basically, the problem can be divided into two parts. The first part consists of recognizing new targets and initializing their tracks, and the second part consists of updating tracks for established targets. Tracor's MTTA, as it

presently stands, is essentially a solution to the first part of the problem. It can take raw data, automatically cluster to recognize targets, and then estimate the associated tracks for each of the targets.

Figure 8.1 represents Tracor's conception of the completed multitarget algorithm. The current MTTA is located in the "search for new targets..." box. The boxes surrounding it contain data management functions. They maintain sensor data pools, place elements in the current track table, and remove points forming a trajectory from the sensor data pools. There is no theoretical development associated with the implementation of these functions, they amount to the lines of code necessary to set up the appropriate data structures and module linkages.

The thrust of the theoretical development will be concentrated on determining if a given data point is associated with one or more current tracks and then performing the appropriate update. There are several potential candidate strategies for attacking this problem including:

1) Nearest neighbor and gating approaches of Sea (Reference 18) and Singer and Sea (Reference 19).

2) A posteriori analysis of track density by Sittler (Reference 20) and Stein and Blackman (Reference 21),

3) Probability data association scheme used by Bar-Shalom and Tse (Reference 22), and

4) Cluster-based data association schemes.

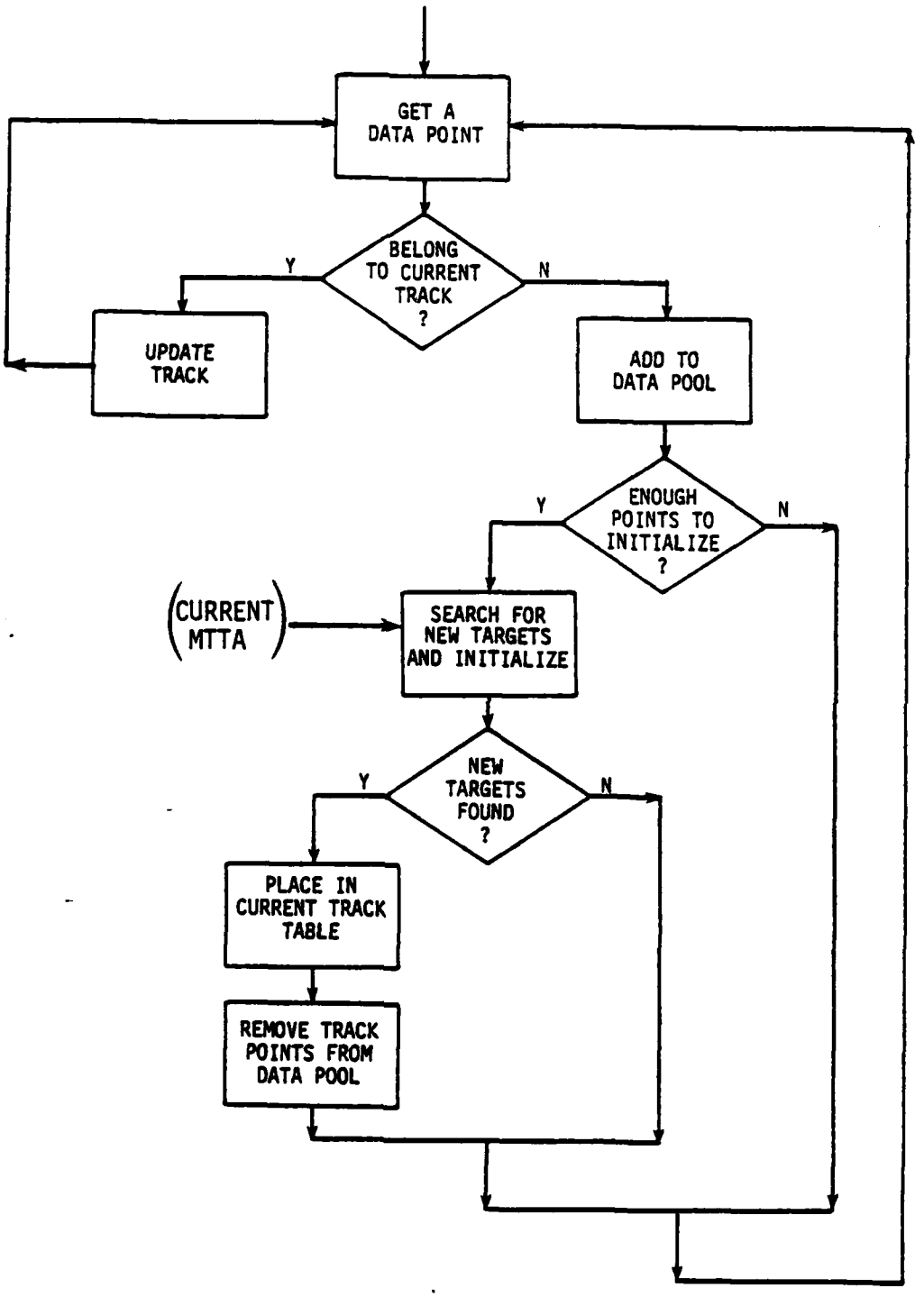


Figure 8.1. Logical Flowchart of Tracor's Proposed Final MTTA

At this point, the thinking is that track determination and state update will be handled by a combination of (1) and (4) in conjunction with the existing sequential filter used by the HTA. Recent work (Reference 23) indicates that (1) is among the most widely used of all measurement classification schemes. Coupled with a gating mechanism at the clustering stage, it seems like a very promising approach. Additionally, to handle the problem of intersecting targets, it may very well be appropriate to allow a given measurement to update more than one track but with a substantially reduced measurement weight for each track. This is similar in philosophy to use of the B_k overlapping cluster algorithm to sort data.

In summary, it is felt that the most difficult part of the multi-target tracking problem, target recognition and track initialization, has largely been solved. The literature contains a broad range of strategies for updating tracks once initialization has occurred, but up until now, there has been little work done with the problem of multi-target track initializations. It is believed that Tracor's MTTA provides the basis for solving this total problem, and it is recommended that its development be continued.

9.0 REFERENCES

1) Reeder, Hugh, Final Report: A Maximum Likelihood Procedure for Air ASW Programs (U), Contract N60921-79-C-0123, 15 May 1980.

2) Corser, Glenn and Wilson, Thomas, Final Report: Hybrid Passive Tracking Algorithms, Contract N00014-78-C-0670, 31 October 1980.

3) Cooper, Don and Corser, Glenn and Wilson, Thomas, Final Report: Hybrid Tracking Algorithm Improvements and Cluster Analysis Methods, Contract N00014-78-C-0670, 26 February 1982.

4) Alam, Kursheed and Mitra, Amitava, "Polarization Test for the Multinomial Distribution," J. Am. Stat. Assoc. Vol 76, 1981.

5) Draper, N.R. and Smith, H., Applied Regression Analysis, John Wiley, New York, 1966.

6) Neter, John and Wasserman, William, Applied Linear Statistical Models, Richard D. Irwin, Homewood, Ill., 1974.

7) Gallant, A.R., "The Power of the Likelihood Ratio Test of Location in Nonlinear Regression Models," J. Am. Stat. Assoc. Vol. 70, 1975.

8) Morefield, Charles L., "Application of 0-1 Integer Programming to Multi-Target Tracking Problems," IEEE Trans. on Automatic Control, Vol. AC-22, No. 3, 1977.

9) Jardine, Nicholas and Sibson, Robin, Mathematical Taxonomy, John Wiley and Sons Ltd., London, 1971.

10) Marshall, Kim and Romesburg, Charles H., "CLUSTAR and CLUSTID-Programs for Hierarchical Cluster Analysis," The American Statistician, Vol. 34, No. 3, August 1980.

11) Ling, Robert F., "On the Theory and Construction of k-Clusters," The Computer Journal, Vol. 15, No. 4, November 1972.

12) Ling, Robert F., "A Probability Theory of Cluster Analysis," J. Am. Stat. Assoc., Vol. 68, No. 341, March 1973.

13) Suich, Ronald and Derringer, George, "Is the Regression Equation Adequate? -- One Criterion," Technometrics, Vol. 19, 1977.

14) Ellerton, Roger, "Is the Regression Equation Adequate? -- A Generalization," Technometrics, Vol. 20, 1978.

15) Bard, Yonathan, Nonlinear Parameter Estimation, Academic Press, New York, 1974.

16) Young, Hugh D., Statistical Treatment of Experimental Data, McGraw-Hill, New York, 1962.

17) Garfinkel, Robert S. and Nemhauser, George L., Integer Programming, John Wiley, New York, 1972.

18) Sea, R.G., "Optimal Correlation of Sensor Data with Tracks in Surveillance Systems," Proc. 6th Hawaii Int'l Conf. on Sys. Sci. IEEE, 1973.

19) Singer, R.A., and Sea, R.G., "A New Filter for Optimal Tracking in Dense Multi-Target Environments," Proc. 9th Ann. Conf. on Cir. and Sys. Th., 1971.

20) Sittler, R.W., "An Optimal Data Association Problem in Surveillance Theory," IEEE Trans. Mil. Elec., Vol MIL-8, 1964.

21) Stein, J.J. and Blackman, S.S. "Generalized Correlation of Multi-Target Track Data," IEEE Trans. Aero. and Elec. Sys., Vol. AES-11, No. 6, 1975.

22) Bar-Shalom, Y. and Tse, E., "Tracking in a Cluttered Environment with Probabilistic Data Association," Automatica, Vol. II, 1975.

23) Goodman, Irwin R. and Wiener, Howard L. and Willman, Warren W., Naval Ocean-Surveillance Correlation Handbook, 1979, (Draft), Naval Research Lab, Systems Research Branch, Washington.

END

FILMED

3-84

DTIC