1999-12

Design and development of the EER Module for BEARTRAP post mission processing system 2000

Danks, John C.

Monterey, California: Naval Postgraduate School

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THIS THESIS IS UNCLASSIFIED WITH THE REMOVAL OF APPENDIX C

THESIS

DESIGN AND DEVELOPMENTOF THE EER MODULE FOR BEARTRAP POST MISSION PROCESSING SYSTEM 2000

by

John C. Danks

December 1999

Co-Thesis Advisors: Charles W. Therrien
Murali Tummala

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE December 1999	3. REPORT TYPE AND DATES COVERED Master's Thesis

4. TITLE AND SUBTITLE: DESIGN AND DEVELOPMENT OF THE EER MODULE FOR BEARTRAP POST MISSION PROCESSING SYSTEM 2000

5. FUNDING NUMBERS

6. AUTHOR(S) John C. Danks

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Distribution authorized to DoD and DoD Contractors only; software related to intelligence; December 1999. Other requests for this document must be referred to Superintendent, Code 0052, Naval Postgraduate School, Monterey, CA 93943-5000 ~~via the Defense Technical Information Center, 8725 John J. Kingman Rd., STE 0944, Fort Belvior, VA 22060-6218.~~

12b. DISTRIBUTION CODE

13. ABSTRACT *(maximum 200 words)*

This work is intended to support an Extended Echo Ranging (EER) addition to the BEARTRAP post mission Processing System 2000 (S2K). S2K is an analysis tool programmed using Microsoft Visual C++ and residing in a Microsoft Windows NT environment. Both BEARTRAP and EER missions are Anti-Submarine Warfare (ASW) missions and are able to be analyzed on the same hardware system due to the use of the same recording media. This thesis develops a design framework for the S2K EER module, which is the software support needed to perform post mission processing for the EER mission. Two submodules of the design are also developed. First is the Virtual Buoy Repositioning submodule, which uses acoustic data to correct errors in sonobuoy locations that are caused by aircraft navigational errors. Second is the Detection and Classification submodule, which processes the acoustic data to identify signal returns from the target. A preliminary analysis of incoming signals is performed using current techniques and exploring a new technique for signal classification.

14. SUBJECT TERMS Acoustics, BEARTRAP, EER, DSP

15. NUMBER OF PAGES

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18 298-102

DESIGN AND DEVELOPMENT OF THE EER MODULE FOR BEARTRAP POST MISSION PROCESSING SYSTEM 2000

John C. Danks
Lieutenant, United States Navy
B.S., The Ohio State University, 1991

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE
IN
ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 1999

Author: [redacted]
John C. Danks

Approved by: [redacted]
Charles W. Therrien, Co-Thesis Advisor

[redacted]
Murali Tummala, Co-Thesis Advisor

[redacted]
Jeffrey B. Knorr, Jr., Chairman
Department of Electrical and Computer Engineering

ABSTRACT

This work is intended to support an Extended Echo Ranging (EER) addition to the BEARTRAP post mission Processing System 2000 (S2K). S2K is an analysis tool programmed using Microsoft Visual C++ and residing in a Microsoft Windows NT environment. Both BEARTRAP and EER missions are Anti-Submarine Warfare (ASW) missions and are able to be analyzed on the same hardware system due to the use of the same recording media. This thesis develops a design framework for the S2K EER module, which is the software support needed to perform post mission processing for the EER mission. Two submodules of the design are also developed. First is the Virtual Buoy Repositioning submodule, which uses acoustic data to correct errors in sonobuoy locations that are caused by aircraft navigational errors. Second is the Detection and Classification submodule, which processes the acoustic data to identify signal returns from the target. A preliminary analysis of incoming signals is performed using current techniques and exploring a new technique for signal classification.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

AMPO	Advanced Maritime Projects Office
DSTFT	Discrete Short Time Fourier Transform
EER	Extended Echo Ranging
FFT	Fast Fourier Transform
IEER	Improved Extended Echo Ranging
JHU-APL	John Hopkins University-Applied Physics Laboratory
NME	Noise Mean Estimate
PFA	Post Flight Analysis
RF	Radio Frequency
SD	Standard Deviation
TOA	Time of Arrival

I. INTRODUCTION

This thesis concerns post mission analysis of Extended Echo Ranging (EER) missions. This is an anti-submarine warfare mission designed to cover large areas of water using airborne assets. While the aircraft systems are forced to operate within the limitations of the aircraft hardware and its capability to process real-time data, the post mission environment is not confined by these limitations. Current post mission processing systems do not address all of the issues encountered during the task of handling the data. This thesis addresses more of these issues and provides a framework for a complete EER post mission analysis system.

A. THE EER MISSION

EER missions are flown from a variety of airborne platforms. For the purpose of this thesis, only EER missions flown from the Orion P-3 aircraft will be discussed. This is a minor simplification, as the hardware being developed for this software will be capable of inputting the acoustic information from other platforms. Once input to the post mission processing system, the information from the different platforms will be in the same format and no modification of any parts of the proposed EER post mission processing system should be necessary.

1. EER Mission Overview

EER is a type of sonar mission that uses airborne acoustic assets to detect submarines. The mission uses an active source to cover large areas of ocean. The source is explosive and creates a "ping" with a broadband spectrum. Energy from this ping is reflected from the target and received at sonobuoys deployed by the airborne asset. This is based on a simple sonar premise: if enough energy is emitted into the water, a return from a target can be heard.[Ref.1]

A detection event has an associated time delay referenced to the time of the source explosion. Such events are used to generate range arcs, which combined with bearing data

from the listening buoy, provide location information. Several of these range arcs and bearing inputs can be used to further refine the location.

2. SSQ-110 Exploder Buoys

The SSQ-110 buoy is a standard "A" size buoy that provides the "ping" for the EER mission. It contains two explosive packages that act as broadband sources. The use of a broadband source for locating and detecting a target presents several differences when compared to current methods of active sonar using narrowband signals. The narrowband signals can be varied in power, pulse width and frequency while the broadband source has a wide spectrum that is fixed in energy. The foremost difference is the difficulty in differentiating a return from reverberation in the water. Another is the increase in background levels caused by the broadband signal. Since a broadband signal provides no Doppler, there is no ability to determine target parameters based on Doppler shift. Finally, the sound source cannot be optimized for the target type or environmental parameters.[Ref.1]

3. Vertical Line Array Directional Frequency and Recording Buoys

The VLAD buoy [Ref.2] is the type of buoy used for collecting the acoustic data. It performs this task using three main sensor elements: an omnidirectional element and two directional elements that are dipoles. By using the two directional elements, one assigned to the North/South axis and the other to the East/West axis, the incoming signals can be divided azimuthally into four cardiod shaped beams during processing.[Ref.1]

A bearing is also generated based on a magnitude comparison between the North/South and East/West signals. In particular, the ratio of the magnitudes of the two signals should correspond to the tangent of the angle from which the incident signal originates. (The actual process is more complicated, and takes into account noise sources that could skew the magnitude ratio.) Since only two sensor elements are used, the resolution of the resulting bearing is extremely limited.[Ref.1]

A second factor limits the accuracy of the VLAD bearing. When the exploder buoy is activated, the magnetic compass used to measure the orientation of the co-located VLAD buoy is adversely affected. Consequently, bearings from a monostatic return have been reported to be worse than those from a bistatic return.

B. BEARTRAP POST MISSION PROCESSING SYSTEM 2000

BEARTRAP Post Mission Processing System 2000 (S2K) is a system developed to deal with a passive sonar buoy field [Ref.3]. However, the similarities between the materials used to report the missions would allow the same hardware to process both passive sonar and EER missions. The following is a description of S2K intended to provide an understanding of how this system was developed and how it can be applied to EER post mission analysis.

1. Overview of BEARTRAP missions and S2K

The BEARTRAP mission is designed to obtain detailed passive acoustic information on a submerged target. At its inception, BEARTRAP was developed to gather data on Soviet submarines. As the cold war came to an end and the Soviet Union collapsed, BEARTRAP missions were used to obtain data on friendly, cooperative targets. Currently, missions are performed not only on United States submarines but also on those of other countries.

S2K is a Microsoft Windows NT-based application created in Visual C++ that is designed to replace the system currently used for BEARTRAP post mission analysis. This analysis is currently done by several organizations using a diverse set of systems. The plan is to integrate these functions into one facility using one hardware platform that meets the needs of all parties involved. [Ref.2]

2. Orion II

Orion II is a software package used for mission replay by the P-3 community [Ref.4]. To accomplish mission replay, the software has the capability to interpret the electronic log package that is produced during a P-3 mission and display this data in a tactical view. This tactical view is a position plot of the mission that changes over time to replay the mission events such as buoy deployment or target detection.

S2K requires some of the data in the electronic log package to perform post mission analysis. Access to the data logs is provided using automation functions that are a part of the Microsoft Windows environment. These functions allow the two programs to share data. Orion II functions as an input source for S2K, allowing S2K access to these logs. The automation also works in the other direction, allowing Orion II to act as the tactical view for S2K. Though the programs work in tandem to share data, they remain separate; any data that needs to be shared must be specified when the program is written.

3. Incorporating EER

The S2K system is capable of performing EER post mission processing. The airborne platforms that are currently used in EER missions are the same as those used for BEARTRAP. Consequently, the tapes and logs that are produced by EER missions have the same format as those from BEARTRAP missions and can be accessed by the S2K hardware. All that is needed is to add software to S2K with the capability of performing EER mission data analysis.

C. PREPARATION AND ANALYSIS OF THE PROBLEM

Several trips were made to talk to P-3 crews and post flight analysis teams to learn about the EER mission and what functions would be needed and helpful in a post mission processing system (see Appendix A). Some problems were found that are not currently addressed, leading to proposed improvements in the overall process.

One such problem is the validity of buoy-to-tape-track assignment. The identity of the buoy on a tape track may be incorrectly reported. This can be detected and corrected by using acoustic cues from the recorded track and comparing them to the aircraft logs. Currently, there is not a system that performs or supports this task.

Another problem is that the aircraft navigation systems cause a noticeable error in the reported buoy position. The solution proposed here is called "virtual buoy repositioning" and is based on acoustic data. The idea is to use the time of arrival of the ping at the VLAD buoys to adjust their positions and reduce the navigational system errors. This process will improve the target position data determined during post flight analysis.

A final problem is the artificial limitations placed on the post mission analysis software, which is identical to that used on the aircraft. This made the development of advanced tools for detection and target parameter estimation a topic of interest. Although there are other systems that are used for processing, such as Distant Thunder or the software that has been developed at John Hopkins University – Applied Physics Laboratory (JHU-APL), these systems are not generally available. The advantage that S2K has is that the hardware is already deployed at the sites that fly the EER missions. Any tool supplied in S2K will be immediately available to the on-site analysis teams.

D. GOALS OF THESIS

The first goal of this thesis was to design an EER software module for S2K. This included an overview of the process flow and its application to EER. It does not include a full implementation. The design covers the basic format of the S2K EER submodules and how they interrelate.

Two of the functions provided in the module were then explored in detail. The first was the idea of virtual buoy repositioning. The intention was to develop a working algorithm based on a set of initial assumptions based on our site visits and discussion of the buoy position problem.

The next function addressed was the ablity to detect and classify a return. The original intention was to adapt the current aircraft software to the S2K platform. The implementation was to be coded and tested in the MATLAB environment. While this goal was not completely realized, in part due to lack of information from the developers of the system, a number of insights about the problem were developed and are reported in this thesis and the associated classified Appendix C.

E. THESIS OUTLINE

The rest of this thesis has the following organization. Chapter II presents the design of the EER module for S2K with a discussion of the submodules. Chapters III and IV cover the implementation of virtual buoy repositioning and the results of testing the algorithm. Chapter V presents a discussion of the detection code adapted from the aircraft software specifications. The results of testing this code and the analysis of the return signals for classification are presented in the classified Appendix C. Finally, Chapter VI provides conclusions with some ideas for further development.

II. EER MODULE DESIGN

A. INTRODUCTION

The most important step in the development of the EER module is the design phase. Prior to designing the EER module, a detailed study of the EER mission was conducted. As part of this study, several sites conducting EER missions were visited, and a summary of the findings is presented in Appendix B. This chapter presents system level design issues that are an outcome of the study. Each section in this chapter addresses a specific submodule of the EER module. When necessary, specific difficulties in the EER process flow are addressed in tandem with the solutions that the specific submodule provides.

The EER module consists of five submodules that reflect a set of sequential tasks as represented in Figure 2.1. The data input submodule facilitates the communication between the EER module and the other portions of the S2K system. The buoy to track assignment submodule is concerned with checking the validity of what buoys are assigned to which tape track. The buoy repositioning submodule processes the acoustic data from the tape tracks and adjusts the buoy positions reported from Orion II. Detection and classification processing extracts detection events from the acoustic data and provides classification information that determines the validity of the detection event as

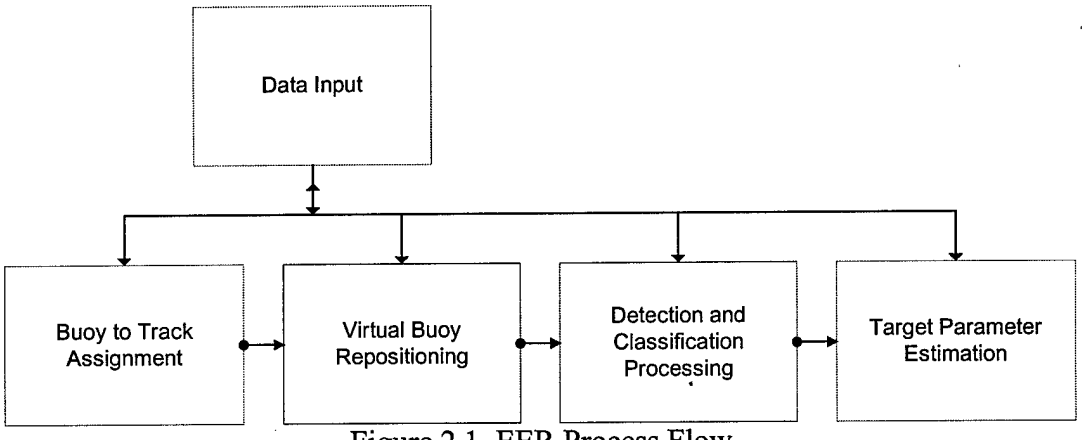

Figure 2.1. EER Process Flow

an actual target. The target parameter estimation submodule uses the detection information to estimate parameters such as target position, direction of motion and target strength. Two of the submodules, buoy repositioning and detection and classification, are discussed with more detail in later chapters. The other three are addressed below.

B. DATA INPUT SUBMODULE

The EER module is intended to be an integral part of the Beartrap S2K software. In this capacity, it will utilize the data input and tracking features of the S2K environment. Of obvious concern is the interface between the S2K parent environment and the child module. One that is less obvious is accessing mission log data from Orion II through the parent. The Data Input submodule will address both of these areas and provide coordination for EER data requests.

1. Time Series Data

The parent software will be relied upon for time series data input, i.e., to read data from tapes and create a digitized time series. The routines to access and store this data reside outside of the EER module. The Data Input submodule will be used as the interface between the storage and retrieval features of the parent program and the child module.

2. Orion II Services

Orion II provides a capability which is essential to processing an EER mission. During a mission, an electronic record of events is generated by the aircraft system. These events reflect many different aspects of what occurs in the aircraft. Some events are as basic as recording the aircraft position at regular intervals. Others are related to specific operator actions. Some of these logs are required by the EER Module. The specifics are supplied with each submodule definition.

Orion II reads the electronic logs from the mission and maintains a database of these events. Of concern to the data input routines is the automation that occurs between

S2K and Orion II. The Data Input submodule will need to request the log events from Orion II using the conduit of the parent program.

3. Sychronization of Temporal Events

The EER module needs the ability to synchronize several frames of reference concerning time. For instance, if a time code is not recorded on the aircraft-generated tapes, the time series data will not have the same reference point in time as the electronic logs. The capability to assign a zero time arbitrarily after digitizing the data would allow the post-flight analysis (PFA) team to synchronize the time series data with other inputs, such as Orion II logs. This functionality may be supplied by S2K's data management system. If it is not, the data input routines will have to provide a method for the operator to perform this synchronization.

The graphical user interface for synchronization is envisioned as a set of horizontal panes each of which represent a time sequence of the various events from the mission. In essence, it is an alignment of tape track events with log events. Initially, only a display of time series data from the tracks will be available. Operator experience is the sole guide for matching the sequences. As processing procedes, further aids are developed such as detection of the ping events for the time series data. Both the raw data and the exploder events provide valuable information to the operator. This information can be displayed in one of the horizontal panes of the user interface as an operator aid.

This parallel interface allows the user an overview of the alignment of the various elements. By adjusting the relative positions of the event streams in the graphical window, the user provides a time offset parameter for each stream relative to the time series data. This time offset parameter can be used to adjust the reference point for each stream. The time associated with any event from the Orion II logs comes from essentially the same source. This means that it is unlikely that a noticeable offset will occur between these events. Consequently, the offset of concern is between the time series data and the Orion II logs.

The adjustment of the "time origin" for any of the logged sequences would be difficult. Such an adjustment would require coordination with Orion II, for the adjustment would occur outside of S2K. The adjustment of the "time origin" for the time series data eliminates this difficulty. By adjusting the time series data only, the process is transparent to external environments such as Orion II.

Time synchronization blurs the boundaries between the data input submodule and the other submodules. As processing continues, more time series events are generated based on the digitized data. For example, times for ping events are derived from energy levels in the acoustic data vice the logs in Orion II. Another example is the generation of detection events. The latter example is poorly suited for use in syncronization as there may not be a corresponding event in the Orion II logs. However, the acoustically derived ping events are an excellent choice for synchronization. The time that a buoy is commanded to explode should be just prior to the energy pulse in the acoustic data. Consequently, the processing in the buoy-to-track assignment submodule should be conducted before the synchronization. This supplies the acoustically derived ping times and a verified buoy-to-track assignment to the synchronization process.

C. BUOY-TO-TRACK ASSIGNMENT SUBMODULE

The system software logs the listening buoys and the associated tracks to which they are assigned. However, errors that may occur in these logs can lead to false asignments. This submodule addresses the false track reporting problem. It is intended to help the operator by providing feedback about the status of buoy-to-track assignments. Proper assignment prevents subsequent errors in the analysis.

Currently, there is no method other than operator action and intervention to verify that the proper assignment occurs. Consider that in a single mission there could be twenty or more ping events. Under these circumstances, it would require twenty instances of operator vigilance to identify possible problems. Once a problem is identified, an interim solution must be developed since there is currently not a system to handle the problem.

The proposed solution is to provide a software analysis tool that uses an energy detector to identify all direct blast receptions at the listening buoys. A comparison of the times of these events yields a distance measurement that can be compared to the distance between reported positions. By generating a list of discrepancies concerning the current assignments and providing this list to the user, a basis for buoy-to-track assignment is developed. The segments of this tool are presented in Figure 2.2.

1. Parallel Input and Preprocessing

Two data inputs are required for this submodule's operation: the time series data from the S2K digitization process and electronic log events for exploder buoy commands retrieved from Orion II. These inputs are obtained and preprocessed in the following ways.

a) Orion II/S2K Buoy Position and Track Assignment

Buoy position data is imported into S2K from Orion II based on mission logs. The data is subsequently stored in the buoy field object of the EER Module. This object tracks the buoy by its Radio Frequency (RF) number. One function that will exist in the buoy field object that is unique to the EER module is buoy pairing. Because each listener buoy is paired with an exploder, the two are often treated as a unit. By referencing the two buoys as one, the duality created in the real world is supported and reinforced in the code.

The buoy position data is used to determine the Euclidean distances between the buoys. These distances are then used to create a set of ordered lists, one for each buoy. The result is that for N buoys, N lists of length $N-1$ are generated. The ith list identifies the other buoys in order from closest to furthest from buoy i.

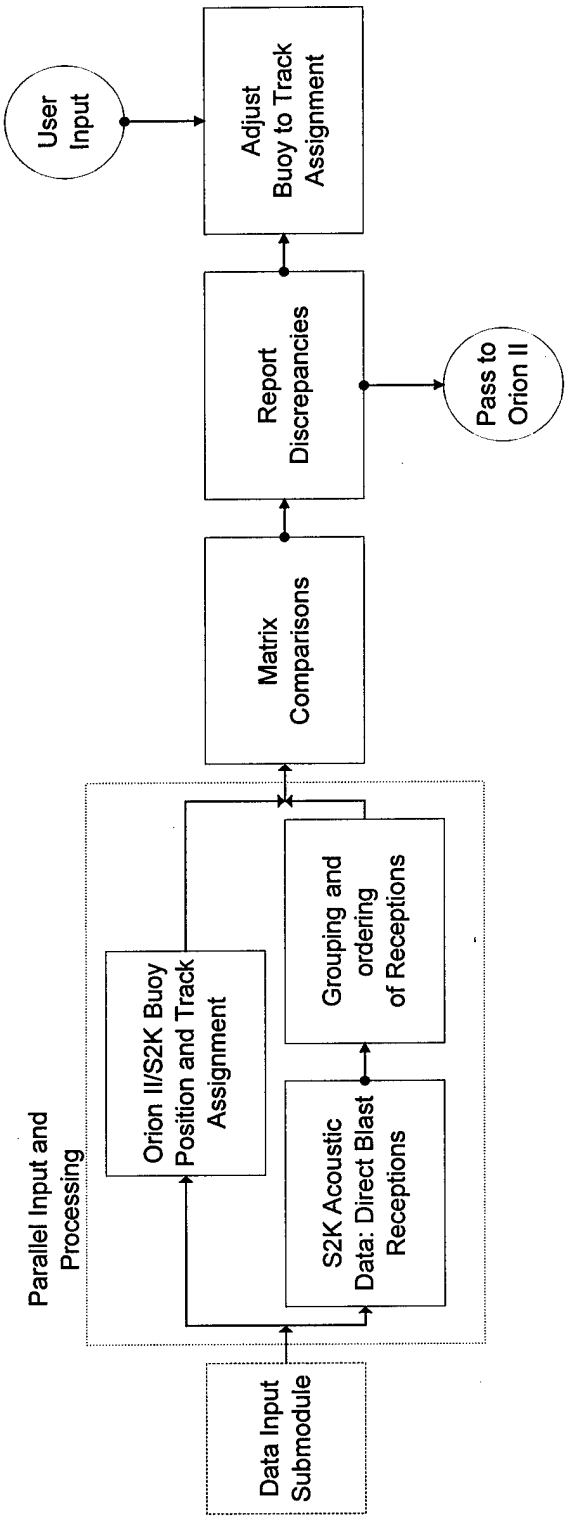

Figure 2.2. Buoy and Track Assignment Process Flow

During the course of the mission, different listener buoys are selected for analysis based on the environment and the source in use. This selection process determines which buoys will be recorded at that time. Consequently, each tape track records the buoy assigned to it by the software for that ping event. This assignment can change from one ping event to another.

A method for identifying and tagging the time series data is needed and S2K supplies this functionality outside of the EER module. This initial buoy-to-track assignment data is based the electronic aircraft logs. Orion II is the gateway between the logs and S2K. The EER module will directly access the S2K data matrix.

b) S2K Acoustic Data: Direct Blast Reception

The EER module will also access the digitized data streams through the S2K system. This data will be processed by an energy detector to determine the direct blast reception times for all recorded pings. As the direct blast reception is orders of magnitude larger than decaying reverb from earlier pings or any other sound source, this is a straightforward task. Each time the threshold of the energy detector is exceeded, a new reception object will be created in the program environment which contains all necessary buoy, track and time data to uniquely identify it.

c) Grouping and Ordering of Receptions

Once the data is processed to generate reception objects, these objects are assigned to specific ping command events from the electronic logs. This assignment is based on proximity of the reception objects in time to the command events. Note that the time between command events is on the order of minutes while the proximity of direct blast receptions from the same ping is on the order of seconds. Consequently, the possibility of associating a reception object to the wrong command event is low if the time series data hase been properly synchronized with the Orion II logs. A command

event object is created that contains all data pertinent to the event: the identifier for the triggered buoy, a list of the associated reception events, and storage for the list of discrepancies created by the following processes.

The reception objects for each event are then sorted according to time. The buoy that gains initial reception of the direct blast should be the listener collocated with the exploder source buoy. The first reception is traced back to its orignal tape track. The buoy assignment for the track at that time is retrieved from S2K. This assignment is checked against the buoy identifier for the command event. If the identifiers do not match, the problem is entered into the discrepancy list.

For the next step, the following assumptions are made about the acoustic environment: straight line propagation and constant sound velocity. Under these circumstances, the time that the explosion takes to travel from the source to the listener buoys corresponds directly to a distance. The actual calculation of the distance is not necessary; note that ordering by reception time corresponds to ordering by distance from the source buoy.

2. Matrix Comparisons

The orderings generated in the previous step are then compared with the orderings generated from the buoy positions during preprocessing. By comparing these two orderings, a list of discrepancies is generated which is added to the command event object discrepancy list. A confidence factor is then associated with each discrepancy based on measurement variances and buoy distance proximity. Proximity is a major concern. Depending on the geometry of the buoy field, some inter-buoy distances may have similar values. If several of these pairings exist inside the set of buoys recorded for a specific ping, small variances in measurements may cause a false discrepancy to be reported.

3. Report Discrepancies

The user is presented with each discrepancy list next to the confidence measures and a best choice buoy to track assignment that minimizes the number of discrepancies. By comparing the discrepancy list and confidence measures to a representation of the reported buoy positions supplied by Orion II, the operator can then come to an informed conclusion. The operator's choices and interpretations can be entered by altering the best choice assignment list. Choices to discard or accept the best choice list without change are also available.

4. Adjust Buoy-to-Track Assignment

The buoy-to-track assignment as understood by S2K and Orion II is updated according to the user's instructions. This must be updated in both programs to prevent confusion at a later point.

D. TARGET PARAMETER ESTIMATION SUBMODULE

Once a target has been detected, the next step is to gather as much data about the platform as possible. The target parameter estimation submodule is the vehicle designed for this task. This submodule also generates hard copy reports concerning the results. These reports are intended for the aircrew and are aimed at improving air crew training methods by providing feedback. When reviewed alongside processing results from the Detection and Classification submodule, the aircrew can identify output characteristics that indicate a target. These characteristics are similar to those output by the aircraft software. This provides the aircrew with detection characteristics in a familiar format that is referenced to one of their own missions which can then be used as a training aid.

The submodule is broken down into four segments for further explanation. These segments are shown in Figure 2.3 and described in detail below.

1. Range Arc Calculation

We can fix a "time origin" for the ping event by assuming that the paired exploder/listener buoys are co-located. This point in time is when the direct blast is received at the co-located listener buoy. A detection during this event creates a time of arrival (TOA) datum. This datum is the difference between the time of the detect and the "time origin". Using the environmental data, a set of positions that correspond to the TOA datum can be produced. If straight path sound propagation and constant sound velocity are assumed, the result is a locus of points that form an elliptical range arc. The foci of the ellipse are the two buoys of interest: exploder and listener. In the case of a monostatic detection, the foci are coincident and the simple case of a circle is the result. The range arc is output to Orion II for display purposes. This artifact identifies when and where detections occur during Orion mission replay.[Ref.1]

2. Position Estimation

The next step is to estimate target position and pass this position to Orion II for display. Target position is estimated using geometry based on the range arcs and DIFAR bearing data. The following cases cover the geometries concerned and the influence of bearing data in this localization process. Note that range arcs cannot be coincident; this would require the buoys associated with two range arcs and the TOA datum to be identical. Coincident range arcs can only occur if the same range arc is recorded twice by the software.

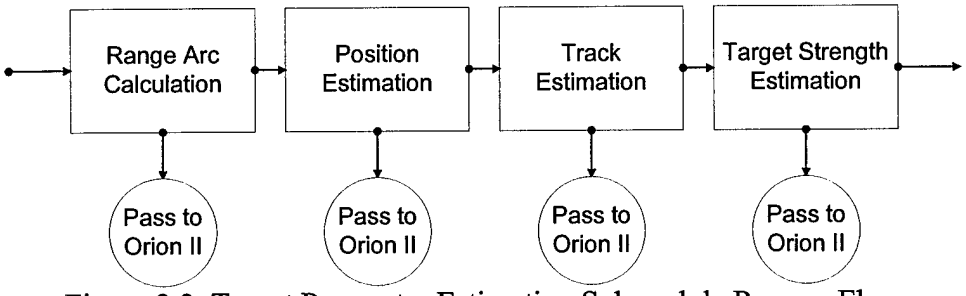

Figure 2.3. Target Parameter Estimation Submodule Process Flow

a) Two Range Arcs with No Intersection

This case needs to be considered to account for possible false detects and errors in measurements. It is an issue concerning robustness of the process. If this case is ignored and an intersection is assumed, invalid results are produced. Such a situation would result in a run time error.

b) Two Range Arcs with Two Intersections

This is a likely case. It occurs when the target range is large compared to the buoy spacing. The result is two ellipses whose foci are so closely spaced that they are essentially circular. The slight variation in centering of these two shapes causes the two intersections.

c) Two Range Arcs with Four Intersections

This case occurs when the target range is small compared to the buoy spacings. Though it is rare in proctice, this is still considered as a possibility.

d) More than Two Range Arcs

The case of more than two returns begins to limit the points to consider. By identifying locations of intersection clusters, many of the above cases can be narrowed down to specific locations. It is possible that given three detections, a unique fix is generated. One method to use for this calculation is grouping points by proximity. This can be accomplished by a neural network designed to detect up to four point clusters. An average position can then be derived for each cluster. This average can be weighted based on the variances in the individual measurements. Ultimately, the choices will be presented to the operator for adjustment or acceptance using the Orion II tactical display.

e) DIFAR Bearing Data

As discussed above, range data alone can be imprecise. Multiple choices for a fix location may exist; in the case of a single detect, it leaves an infinite number of points to consider. A second source of data that can narrow the choices is the DIFAR bearing. This bearing tends to have a large amount of error. Consequently this will affect the accuracy of any postion gained from it. However, in the case of a single detect per event, the bearing will give a hybrid crossfix with the range and thus an approximate location. When the range ellipses cross at multiple points, the bearing can be used to eliminate some of the points from consideration.

Caution should still be exercised when approaching the bearing data in the case of a single detect per ping in the monostatic mode. The listener buoy has just been exposed to a high pressure blast after the ping. Consequently, its bearing may be adversely affected and have even less accuracy than reported by the technical documents. While no studies have been made of this problem, operators have reported that the bearing accuracy is degraded.

In order to recognize these possible inaccuracies, a quantitative measurement error is reported with each position calculation. The reported error depends on the variances in the TOA and bearing measurements. A second element is a comparison of the sources for the fix; for instance, if three lines of position are entered into a fix and there is only one crossing point, that increases the expected certainty of that fix. If there are three crossing points and they are widely spaced, the fix is considered poor. A quantitative measurement can be made of this spread and used in the error measurement of the fix.

3. Track Estimation/Target Motion Analysis

By fixing the position and providing an uncertainty measurement, a probability distribution concerning position is formed. However, by taking several of these fixes, one

can develop a larger picture of position. Depending on the number and accuracy of the fix points, two options are available.

The first possibility is to estimate the path of target motion. For this, the position data must have a low error and contain enough points to represent the actual target motion, including possible maneuvers. In this case, the EMST tracker in S2K can use this input to generate a track and send this track to Orion II. Unfortunately, large errors in measurement and a low number of detections are usually the case for EER missions. The use of the EMST tracker [Ref.5] will be precluded in most cases.

The second option is generation of an advanced area of probable location. By producing best curve fits for the data and projecting this out over time, a time varying area can be generated that accounts for target motion as well as position uncertainties. This provides a general analysis of target motion that is beneficial for intelligence gathering and for planning follow-up missions. This construct will be passed to the Orion II display.

4. Target Strength Estimation

The target strength can be estimated from the sonar equation using the known values of source level, detection energy and environmental parameters. If a good track is developed, a target strength estimate is related to the aspect angle of the target. Aspect related values are of the most use for further analysis of the EER process as well as intelligence gathering.

THIS PAGE INTENTIONALLY LEFT BLANK

III. VIRTUAL BUOY REPOSITIONING SUBMODULE

The goal of the Virtual Buoy Repositioning submodule is to use acoustic data gathered during the mission to adjust the recorded buoy positions. By adjusting position, the accuracy of subsequent steps in EER will be increased. For instance, accuracy of target postion estimates relies directly on proper buoy positioning and is especially sensitive to buoy positions at long ranges.

A. POSITION ERROR CONCERNS

1. The Dual Frame of Reference

There are two frames of reference used for position reporting: a tactical reference and a geographic reference. The tactical frame of reference centers around an arbitrary coordinate point entered at the start of the mission. Whether or not this point has any relation to the "real world" is up to the system operator. The system will use this point with subsequent "dead reckoning" to provide position information regardless of its relevance. The geographic frame of reference takes position data from the navigation system. The existence of these two frames of reference can cause confusion. Since the mission is executed in the tactical frame of reference, the following discussion addresses the points from this frame of reference only. This ensures an end effect that can be compared to the mission results.

2. Sources of Error

All positions in the tactical frame of reference are subject to the increasing errors caused by the dead reckoning system. These errors are corrected by periodic operator updates to the system. The goal is to have accurate relative position of the mission objects, which include the aircraft, sonobuoys, and estimated target trajectories. By confirming aircraft position compared to a known point, usually a sonobuoy, the dead reckoning system can be restarted. Frequent navigational updates to the tactical frame of

reference minimize the relative errors between recorded positions. However, only the initially entered coordinate point has an effect on how these positions relate to the "real world."

The sources of position error are thus based on the dead reckoning capability of the aircraft systems and the frequency of the system updates. The largest contributors to the errors generated between updates are turns made by the aircraft. As turn rate increases, the error increases. A constant velocity for the aircraft induces little error.

The aircraft is operated with the knowledge of these principles. The aircrew's intention is to designate lines of buoys and to maintain steady flight while deploying the line. These so-called "linear groups" tend to have low relative position errors. However, since patterns do not necessarily place all sonobuoys on the same line, turns during field deployment are unavoidable. This leads to errors between the groupings that grow during the deployment of the field.

B. BUOY REPOSITIONING OVERVIEW

The following is a general discussion of the steps involved in Virtual Buoy Repositioning. The process is based on the flow diagram in Figure 3.1. In order to better understand the process flow, a brief discussion of the algorithms involved is included with each step. A detailed discussion of the pertinent algorithms is supplied in the next section.

1. Group Buoys and Designate Anchor Buoy

The first step in the process allows the user to group the buoys. The idea of grouping buoys is based on the sources of error. Recall that buoys dropped in a straight line while maintaining constant flight parameters tend to have less relative error. The assignment of buoys to groups allows the algorithm to exploit this relationship in later steps.

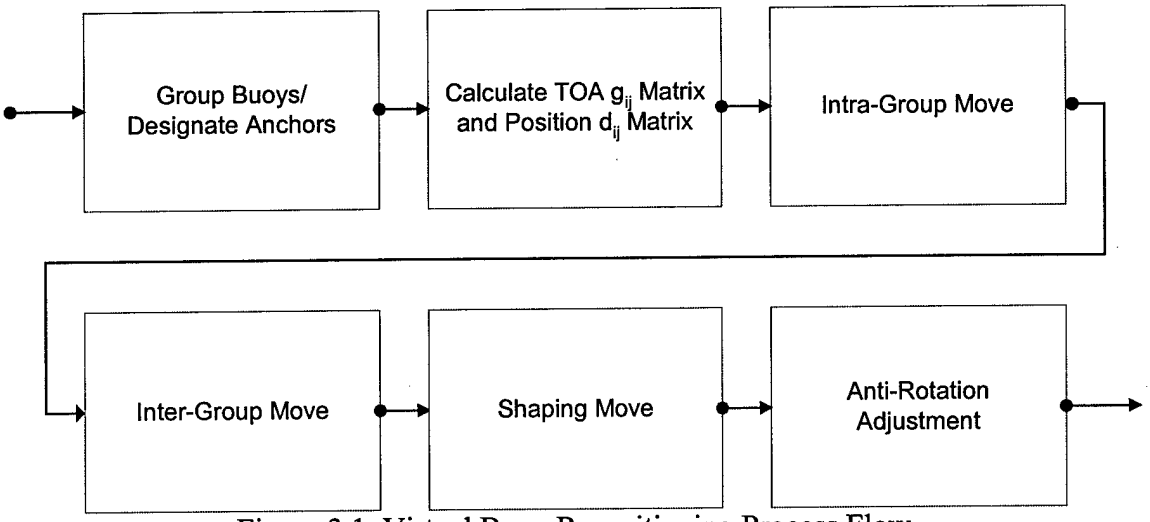

Figure 3.1. Virtual Buoy Repositioning Process Flow

The user is also required to designate an "anchor buoy." This buoy acts as a fixed reference point for the virtual repositioning process. This fixed point is necessary to limit rotation and translation of the buoy field. That is, without such a fixed reference point there would be an infinite number of equivalent solutions to the buoy repositioning problem caused by translations and rotations which are consistent with the given inter-buoy distance measurements.

2. Calculate Inter-Buoy Distance Matricies

There are two sets of inter-buoy distances relevant to this problem. The first distance is calculated based on the TOA events from the previous submodule. The time difference can be directly translated to a distance based on a propagation model. Constant velocity and straight line propagation for sound are assumed here. The distance so calculated between the i^{th} and j^{th} buoy is designated g_{ij} and is entered as the element in the i^{th} row and the j^{th} column of what is referred to as the TOA distance matrix. These distances are referred to in the process as the "true" inter-buoy distances.

The second set of distances is calculated using the reported tactical buoy positions from the Orion II input. These distances are designated d_{ij} and entered into another

matrix. These distances are the starting values for the repositioning algorithms. Note that, as the process continues, this matrix is updated whenever buoys are moved. That is, the individual d_{ij} represent updated inter-buoy distances throughout the process.

3. Intra–group Movement

The first step in the process uses the Shaping Algorithm to correct the buoy positions within the individual groups. The goal of this algorithm is to obtain the desired inter-buoy spacing. The term "shaping" is used because the buoy spacings have a direct correspondence to the geometric shape of the field. This procedure is concerned only with relative buoy positions. It has the effect of removing the generally smaller errors between buoys of the same linear group and increases the effectiveness of subsequent repositioning steps.

The concerns of rotation and translation for the whole field are not a consideration for the individual groups. Any rotation or translation that occurs here is treated in later steps. Consequently, the individual groups that do not contain the anchor buoy are temporarily anchored to an arbitrary member buoy without any significant adverse effect.

4. Inter–group Movement

This step is used to lessen rotation of the buoy field during a field-wide shaping procedure. The grouping provided by the user is the key for this inter-group algorithm. The algorithm is designed to move the groups relative to each other in order to minimize the average distance errors. This method decreases position error while inducing negligible field rotation. This also tends to lower the rotation that will be introduced in the subsequent shaping steps. The positions provided in this step are saved in an auxillary matrix for use in Step 6.

5. Field Reshaping

This step uses the Shaping Algorithm described in the Intra-Group Movement step but applies it to the whole field instead of a subgroup of the buoys. This brings the buoy field as close to its final shape as possible. It must be emphasized that this step only adjusts the *relative* position of the buoys. This limitation causes rotational and translational effects that are addressed in other steps.

6. Anti–Rotation Adjustment

The goal of anti-rotation adjustment is to remove as much rotational error from the field as possible. The algorithm to do this uses the positions saved in the auxillary matrix prior to the field reshaping. The current positions are compared to these to obtain an average position error. While some of this error is associated with the relative position adjustment, there is a portion that is due to rotation alone. The intention is that rotating the reshaped field around the anchor buoy to minimize the average postion error will minimize the rotational effect on the field. The error can only reach zero if there is no error in the shape of the field and no translational errors.

C. BUOY REPOSITIONING IMPLEMENTATION

1. Simplifying Assumptions

To investigate the idea of virtual buoy repositioning, several assumptions are made to create a coherent set of algorithms:

1) Sound velocity is constant in water.
2) Propagation of sound is in a straight line (i.e., no reflection or refraction).
3) There is no relative drift in the buoy field during mission.
4) All Buoys are pinged at least once.
5) Data is recorded for all buoys (i.e., no missing data).

The environmental assumptions, numbers 1 through 3, are the least limiting. Considering the size of the buoy field, the distances calculated based on constant velocity and straight line propagation of sound will still tend to be accurate. Many current methods for target analysis rely on similar assumptions. In a deep water environment, the assumption of a unique drift vector for all buoys will also be accurate. However, in littoral environments or in areas, such as the Gulf Stream, this assumption may be more difficult to justify, and operators should be cautioned.

The data availability assumptions, numbers 4 and 5, are perhaps the weakest point of the process. For instance, a change in ping sequence by the aircrew based on new information or a shift to passive prosecution may cause some buoys not to be pinged and thus limit the EER data set. Also, if more buoys are deployed than there are tracks on the tape deck, not all buoys can be recorded.

2. Description of Algorithms

a) Shaping Algorithm

The Shaping Algorithm is based on an algorithm by Fukunaga [Ref.3] which concerns mapping points of N-dimensional space to points in a two-dimensional space while attempting to preserve their inter-point distances. The usefulness of this idea to buoy repositioning is apparent when one considers the goal. We are moving data points with distances d_{ij} to conform to an assumed "true" set of distances, g_{ij}. This corresponds to a special case of Fukunaga's mapping where the source and destination space are both two-dimensional, and provides a method to iteratively change buoy postions and obtain the desired spacing.

The method is based on minimization of a measure of normalized accumulated error, specifically the following error criterion:

$$\varepsilon = \left(\sum_{j=2}^{N} \sum_{i<j}^{N} g_{ij} \right)^{-1} \sum_{j=2}^{N} \sum_{i<j}^{N} \frac{\left(g_{ij} - d_{ij}\right)^2}{g_{ij}}. \tag{3.1}$$

The nomalization factor used is the summation of all the elements of the TOA distance matrix. The second summation is the measured error term that accounts for the difference between g_{ij} and d_{ij}. Because the matrix is symmetric and the diagonal elements are zero by definition, only the g_{ij} terms below the diagonal are used in the summations. [Ref.3]

First, let us observe that each d_{ij} is a function of the assumed coordinates (x_i,y_i) and (x_j,y_j) for the buoys i and j. Thus the error ε is implicitly a function of all the coordinates (x_k,y_k) which are not fixed. Let (x_k,y_k) denote the position for the k^{th} buoy. Then a steepest descent type of algorithm can be used to try to minimize this error. At each iteration, the coordinates are changed as follows:

$$x_k(l+1) = x_k(l) - \alpha \frac{\partial \mathcal{E}}{\partial x_k}$$
$$y_k(l+1) = y_k(l) - \alpha \frac{\partial \mathcal{E}}{\partial y_k} \qquad (3.2)$$

where α is a step size parameter. This algorithm, while not *guaranteed* to minimize ε, is reasonable in that it moves each coordinate in a direction that tends to reduce the error.

In order to apply the iteration to minimize the error, the derivative of ε is taken with respect to x_k, a coordinate element corresponding to the k^{th} buoy. This requires the use of the chain rule, as d_{ij} depends on x_k as shown in Equation 3.3:

$$\frac{\partial \mathcal{E}}{\partial x_k} = -2\left(\sum_{j=2}^{N}\sum_{i<j}^{N} g_{ij}\right)^{-1} \sum_{j<k}^{N}\left[\frac{\left(g_{kj} - d_{kj}\right)}{\left(g_{kj}\, d_{kj}\right)}\right]\left[x_k - x_j\right] \qquad (3.3)$$

where $d_{kj} = \sqrt{\left(x_k - x_j\right)^2 + \left(y_k - y_j\right)^2}$.

Substituting this result into Equation 3.2 yields the following:

$$x_k(l+1) = x_k(l) + 2\alpha\left(\sum_{j=2}^{N}\sum_{i<j}^{N} g_{ij}\right)^{-1} \sum_{j<k}^{N}\left[\frac{\left(g_{kj} - d_{kj}\right)}{\left(g_{kj}\, d_{kj}\right)}\right]\left[x_k - x_j\right]. \qquad (3.4)$$

This process needs to be applied to the y_k as well. The formulas are identical except for the substitution of y for x.

The algorithm uses Equation 3.4 to adjust the buoy positions as shown in the following pseudocode:

For counter1 = 1 to (number of repetitions)

```
    For counter2 = 1 to (number of buoys)
        For counter3 = 1 to 2
            y_k = y_k -2α(δε/δ y_k); ⇦Equation 3.4
        End loop3
    End loop2
    Recalculate d_ij;
    If (average distance moved)<(minimum position error)
        Break from loop1;
    End if
End loop1
```

Note that this process has two ways to end: either the maximum number of repetitions is reached or the average distance that a buoy is moved falls below a minimum threshold.

The design of this algorithm does not specify the buoys that are considered. By passing only the buoys of concern with their distances, subgroups inside the field can be adjusted. The algorithm implementation gives this flexibility so that the same code can be used in both the intra-group step and the reshaping step.

b) Inter-Group Algorithm

The error measurement used for position correction is the average error vector:

$$\bar{v} = \frac{1}{N}\sum_{i=1}^{N} v_i \tag{3.5}$$

where N is the size of the group of buoys to be moved. The error vector v_i between buoy i and j is shown in Figure 3.2. The length of the error vector is the difference between g_{ij} and d_{ij} and it points in the direction from buoy j to buoy i. The error vector between buoy j and all buoy i's in the group are found and then averaged to form the average error vector.

Each iteration designates what will be called a "hold group." This is the group that will be used as the reference to calculate average error vectors during the iteration. The error measurement used to designate the hold group is the sum of absolute distance errors, a measurement that is unique for each buoy j:

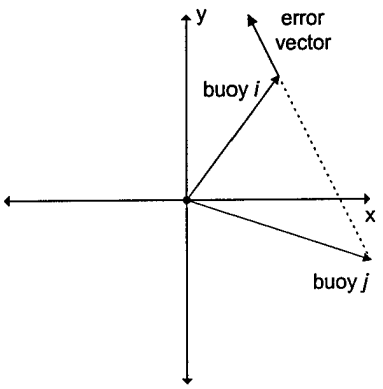

Figure 3.2. Definition of Error Vector

$$\varepsilon_{ij}^{abs} = \sum_{i \neq j} \left| g_{ij} - d_{ij} \right| . \tag{3.6}$$

The algorithm that applies these error measurements is illustrated using the following psuedocode:

```
For counter1 = 1 to (number of repetitions)
    Calculate ε_j^abs for all buoys; ⇦Equation 3.6
    Sort buoys by |ε_j^abs|;

    Do: move up the list of buoys
        Until: buoy chosen is not in the last hold group
    End Do
    If (counter1=multiple of 5) or (counter1=(multiple of 5)+1)
        Set hold group to anchor group;
    End If

    For counter2 = 1 to (number of groups)
        If (counter2 ≠ hold group)
            For counter3 = (cycle through hold group buoys)
                Calculate v̄ for group (counter2) using (buoy counter3); ⇦Equation 3.5
                Subtract error vector to positons of all buoys in group (counter2);
                Recalculate d_ij;
            End loop3
        End If
    End loop2

    If (average distance moved)<(minimum position error)
        Break from loop1;
    End if
End loop1
```

The first step in the iteration is to choose the hold group. The choice is based on three factors. The first factor is chosing a buoy with a low value of ε_j^{abs}. This is

an indication that the buoy is in proper position relative to the rest of the field. By using the group that contains this buoy as the hold group, a "best choice" starting point is established. The second factor is to choose different hold groups each time through the process. Using the same group twice in a row has little or no effect; the error has already been reduced close to zero on the first pass. The third factor is to force the use of the anchor group during the process. The anchor group is defined as the group containing the anchor buoy. This is done by choosing the anchor group as the hold group whenever counter1 is less than or equal to one in modulo 5 arithmetic. While this uses the anchor group as the hold group for two consecutive repetitions, the emphasized bias towards the anchor group as the true basis for postion justifies this choice.

Next, the buoys in the hold group are used to define average error vectors for the other groups, which are termed "move groups." The move groups' buoys then have their positions adjusted. This is done in a sequential fashion by using loops. Each buoy in the hold group is considered separately; the average error vector it generates is applied to the move group associated with counter2 prior to proceding to the next buoy in the hold group. This sequential process does cause the buoys to move through a series of points that surround the minimum error point that would be achieved if all hold group buoys were considered at once. However, the sequential process converges faster. When the anchor group is used consecutively, the distance of this locus of points is reduced in magnitude further justifying factor three in the last paragraph. Once the move group has its postion adjusted, d_{ij} must be updated for the next pass.

Note that this algorithm allows two methods of exit. The first method is by a limitation in the number of iterations. This prevents continuation of the algorithm if any oscillations or divergence occurs. The second method is based on the average movement of buoys falling below a minimum error level. This allows the algorithm to stop if the reduction of the error metric falls below a minimum level. At this point, the algorithm has reached a designated point of diminishing return; the error reduction is no longer worth the calculation required.

c) *Anti-Rotation Adjustment Algorithm*

Anti-Rotation Adjustment is the least complex algorithm of the set. Its error measurement is based on a position error measurement between the positions before and after the Field Reshaping step. The error criterion that is minimized is:

$$\varepsilon_{pos} = \frac{\sum_{i=1}^{N}\left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2\right]^{1/2}}{N} \tag{3.7}$$

where N represents the number of buoys in the field, and x and y are buoy position coordinates. The circumflex denotes the buoy postion prior to the Field Reshaping.

The algorithm can be represented by the following pseudocode:

```
Translate buoy field so anchor buoy is at origin;
Set initial θ;
Calculate ε_pos;
Set new ε_pos=0;
While new ε_pos -ε_pos >(minimum difference)
    Rotate field by θ;
    Calculate new ε_pos;
    If (new ε_pos )>= ε_pos
        Do not update buoy positions;
        θ = -θ/2;
    Else
        Update buoy postions;
        Store new ε_pos as ε_pos;
    End if
End while loop
```

This is a basic approach to find the minimum point for ε_{pos} based on the difference between successive error calculations. When the difference in the value of ε_{pos} between successive iterations falls below a minimum difference, the algorithm exits the iterative loop. The internal steps of the iteration test whether the error metric is increasing or decreasing according to the following description of the pseudocode.

The first step is to calculate the error at the current field position. Then the field is rotated by the incremental angle θ, and these new buoy positions are stored in a temporary matrix. The new positions are then used to calculate a new error value. If the

error is lower for the rotated field, the field buoy positions are updated to the rotated positions. In the case where the error increases, the incremental angle is halved and the direction of rotation is reversed. There are only two times that this can happen. One is at the beginning of the algorithm and corresponds to an initial direction of rotation away from a minimum. The other time is when the field is rotated past a minimum error point. When in proximity to a minimum point, the algorithm may need to reverse more than once to continue iterating towards the minimum. This is essentially an oscillation across the minimum point until the step size is less than the distance to the minimum.

It must be understood that this algorithm finds the first local minimum it encounters and then exits. In the case of rotating a two buoy field where one buoy is the center of rotation and the other moves, there can be only one minimum. However, when considering a larger field of buoys, there may exist several local minima. It is believed that the minima are close enough together in terms of both rotation angle and error value that finding a local minimum is sufficient.

IV. VIRTUAL BUOY REPOSITIONING TESTING AND RESULTS

This chapter covers the development of the Virtual Buoy Repositioning submodule. First, an overview of the testing procedure is given. This is followed by a discussion of the measurements of error. Then the results are presented: first the Shaping Algorithm results and then the Process Flow results. This order will demonstrate the rotation and translation effects caused by the Shaping Algorithm and allow a comparison to the full Process Flow results.

A. OVERVIEW OF TESTING

1. Creating the Buoy Field

The first step in the testing procedure was to create a hypothetical buoy field. The field contained sixteen buoys arranged in a grid. Nominal buoy spacing was given by a square unit cell that was ten thousand yards, or five nautical miles, on a side. The buoys were then grouped into linear groups of fours: one through four in the first group, five through eight in the next, and so on. The resulting field, including the buoy numbering, is shown in Figure 4.1. These postitions were used to represent the *reported* positions that would be obtained from the Orion II software.

The next step in the testing process was to create a set of true postions for the field. This was accomplished by generating error vectors and adding them to the reported buoy positions. The set of error vectors was created using a white gaussian noise process of zero mean with a standard deviation (SD) of two thousand feet for each of the two coordinate axes. This represents the relatively small error generated by the aircraft during level flight. Next, each group has successive error vectors with SD equal to two thousand feet added to each of the member buoys. This represented additional errors inserted by the aircraft turning between successive groups. Buoy #1, the chosen anchor buoy, was left errorless. This allows us to compare results from the Monte Carlo trials more easily.

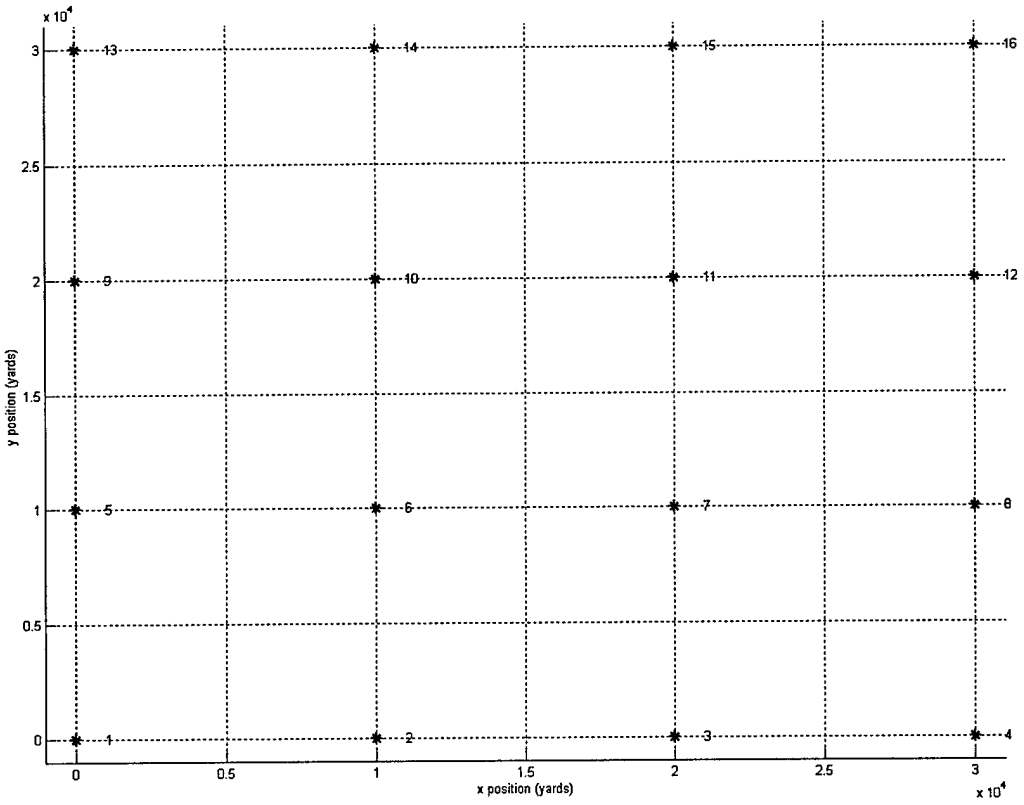

Figure 4.1. Reported Buoy Positions

The process described above may seem backward; it may seem to the reader that the reported postions are supposed to be the ones in error, not the actual positions. However, one must consider the frame of reference. The aircrew is watching a display based on the tactical frame of reference and attempting to place the buoys in proper relative positions. Consequently, the observable results are often close to perfection. If the actual positions were observable, they would be skewed from this perfect representation. In other words, the reported positions of the buoys make the buoy field look like a perfect geometric shape when in fact errors in the navigation system make this an untrue version of reality. The error process described above attempts to create a similar picture by forcing the reported positions into a perfect grid and then adding error to obtain the actual postions.

2. Calculating the Distances

The generation of the d_{ij} matrix is straightforward; Euclidean distance measurements are used. However, the generation of g_{ij} for the testing procedure requires more explanation. In actual use of the algorithm, the g_{ij} would be derived from TOA measurements. For purposes of testing with the positions of the buoys depicted in Figure 4.1, the g_{ij} can be computed directly from the grid.

For a set of data runs, the limitations of assuming errorless g_{ij} were explored. It is unrealistic to expect that the TOA distance measurements will contain no error. The source of these measurements, an estimation of the direct blast reception times, will have some associated error. The effects of the environment changing the sound velocity and propagation path will contribute to TOA error as well. Consequently, gaussian errors were added to the g_{ij} during the calculation step for a secondary set of data runs. Several different levels of noise power were considered in an attempt to quantify the effect that error in the TOA distance measurements will have on the results.

3. Using the Process

The next step was to apply the Virtual Buoy Repositioning process described in Chapter III to the buoy field. This was performed within the Matlab environment. A list of the files used is provided in Appendix C. Each data run consisted of two hundred Monte Carlo trials under the same conditions of error for position and g_{ij}. The average results were considered for each buoy and for the overall field when calculating error metrics.

B. ERROR MEASUREMENTS FOR TESTING

The measurements used to represent the effectiveness of the algorithms are different from the error metrics used inside the algorithms. This is made possible by the method of development and testing, which provides access to the actual buoy positions.

1. Average Distance Error

This error measurement is calculated using the formula:

$$\varepsilon_d = \frac{2}{N(N-1)} \sum_{j=2}^{N} \sum_{i<j} \left(g_{ij} - d_{ij} \right) \tag{4.1}$$

where N is the total number of buoys and the distance terms are as defined in the last chapter. The information provided by the average error between g_{ij} and d_{ij} is a measurement of how closely the algorithm is approximating the true shape of the buoy field. The direct relationship between the inter-buoy distances and the relative positions of the buoys is what makes this measurement useful. However, this measurement is invariant to rotation and translation errors.

2. Average Position Error

The Average Position Error measurement is calculated as:

$$\varepsilon_p = \frac{2}{N(N-1)} \sum_{j=2}^{N} \sum_{i<j} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right]. \tag{4.2}$$

The circumflex notes the reported position updated by the algorithm while the other is the true buoy position. The information provided by ε_p is a measurement of how well the algorithm approximates the true buoy positions. This is the ultimate test of design usefulness and reflects all possible errors: field rotation and translation as well as shape.

C. FIELD RESHAPING RESULTS

A typical plot of the field before (shown by x) and after (shown by +) reshape with actual positions (shown by o) superimposed is shown in Figure 4.2. These results are given to emphasize one critical issue: rotation of the buoy field. The error in the inter-buoy distances was on the order of 10^{-11} yards, while ε_p had an average value of 443.6 yards across two hundered trials. Other methods are necessary to lower the value of average position error.

D. PROCESS FLOW RESULTS WITH NO DISTANCE ERROR

This section presents the processing results when the elements g_{ij} are errorless. The error metrics were calculated after each step of the flowchart in Figure 3.1. A data run with only the Field Reshaping step considered was generated for a comparison. The progression of this process is represented by the error values in Table 4.1 and Table 4.2. These tables reflect the Average Distance error measurements and the Average Position error measurements, respectively.

The first result is that the final position errors are less than the error produced by using the Field Reshape alone, as seen in Table 4.2. This justifies the claim that the current process reduces the field rotation that reshaping imparts. The second result is that the distance error decreases over all the steps except the last one. This is to be expected as the Anti-Rotation algorithm has no effect on the relative spacing of the buoys; it only

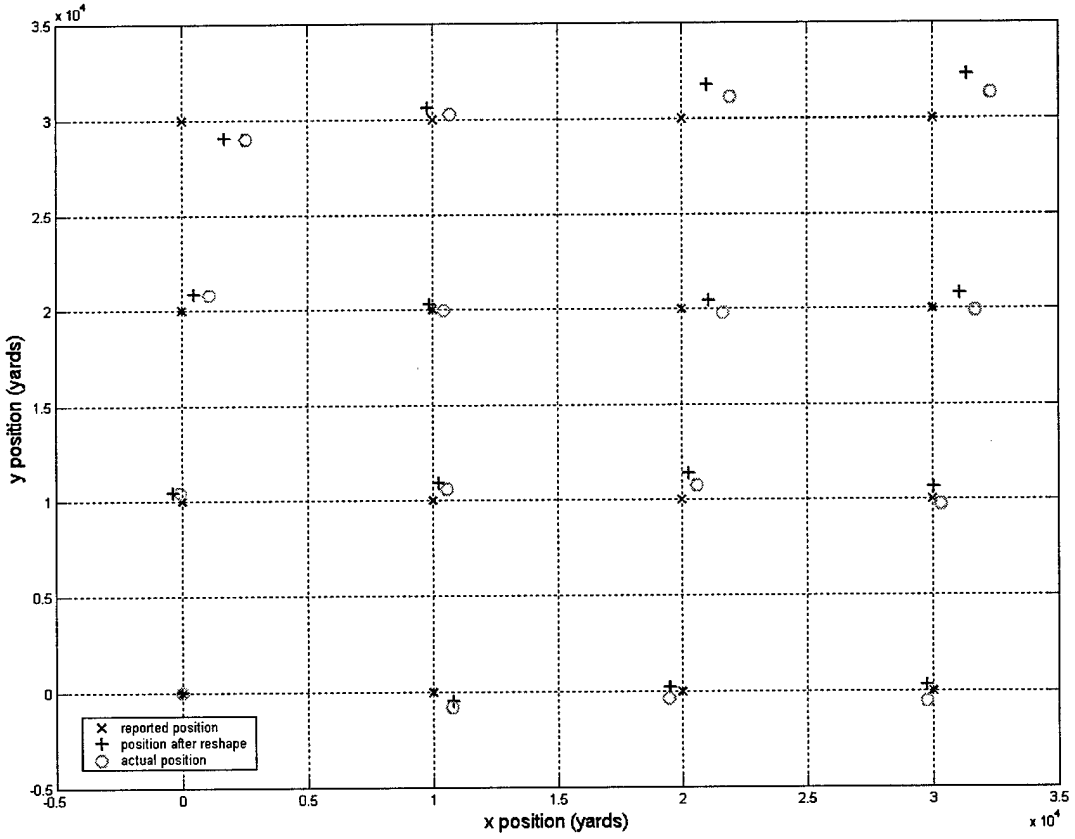

Figure 4.2. Typical Example of Field Reshaping Alone

	initial	intra-group	inter-group	reshaping	antirotation
error (yards)	884.6	737.3	386.7	$\sim 10^{-11}$	$\sim 10^{-11}$

Table 4.1. Average Distance Error Results for Ideal g_{ij} (in yards)

buoy number	Average Position Error by Step (yards)					
	initial	intra-group	inter-group	reshaping	antirotation	shaping only
1	0	0	0	0	0	0
2	823.5	535.3	535.3	114.3	130.1	185.4
3	787.3	475.2	475.2	228.2	257.3	367.8
4	846.0	544.1	544.1	342.6	386.4	551.5
5	1171.2	1171.2	709.9	114.5	129.1	185.0
6	1157.9	1146.1	640.0	161.0	181.2	259.1
7	1202.6	1200.6	611.0	254.8	287.2	411.7
8	1130.3	1167.4	551.8	361.5	407.2	580.4
9	1396.4	1396.4	798.2	228.0	257.5	370.3
10	1443.4	1426.6	802.0	256.4	288.9	413.9
11	1355.0	1386.8	705.7	323.8	365.5	521.5
12	1400.0	1441.7	602.3	412.6	465.1	663.2
13	1750.2	1750.2	941.1	342.1	386.6	553.5
14	1684.8	1713.8	957.8	361.6	408.2	584.3
15	1643.3	1648.4	780.4	413.1	465.8	665.9
16	1650.9	1696.3	761.5	483.9	546.2	783.7
All Buoys	1215.2	1168.8	651.0	275.0	310.2	443.6

Table 4.2. Average Position Error Results for Ideal g_{ij} (in yards)

affects the rotational orientation of the whole field. Consequently, the results in Table 4.1 are universally positive; the Shaping and Inter-Group algorithms decrease the Average Distance error.

The results in Table 4.2 have other positive aspects: the position error tends to decrease, with the final error being less than the starting error. There is a slight increase in position error for the intra-group shaping step that should be noted such as in the case of buoys 8 and 11. This is not a negative result: this step is expected to effect only the distance error. The conclusion is that the decrease in average distance error offsets the increase in average position error.

There is a negative result concerning the Anti-Rotation Algorithm. This step tends to increase the Average Position error. The only purpose of this step was to remove the rotational effects that are induced by Field Reshaping. Consequently, success of this effort can only be measured in terms of the actual buoy positions. While individual cases exist where the results are improved by the Anti-Rotation algorithm, the average result is to generate more error. The increase is due to one of several factors. First, the initial assumption that the local minima are close together and of approximately equal value may be poor. Next, the choice of an anchor buoy on the edge of the field maximizes the linear distance that the furthest buoy moves. By choosing an anchor buoy in the center of the field, the results may be improved. Finally, the idea of matching the fields before and after the reshape step may not be viable. In this case, the step should be discarded. Regardless of the circumstances, this step should not be included in the final process until this issue is resolved. Figure 4.3 provides a graphical interpretation of how the error is adjusted for the entire buoy field. The plot shows a scatter plot of ending positions for each buoy in the field over all the Monte Carlo trials. The positions are all adjusted to show relative error with respect to the actual position for each trial. This figure emphasizes the rotational effect of the Reshaping process. The results appear to fall on arcs centered on the anchor buoy. The spread in final position increases with distance from the anchor buoy.

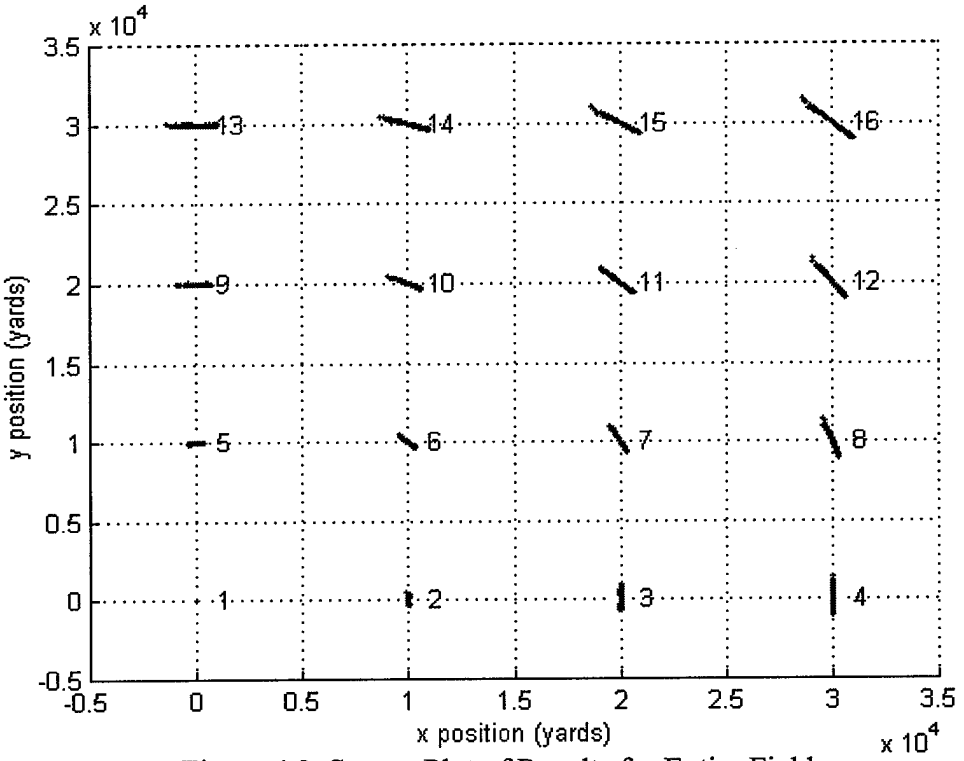

Figure 4.3. Scatter Plot of Results for Entire Field

The overall improvement in buoy position is shown in Figures 4.4 and 4.5. Figure 4.4 is a scatter plot of the initial position errors while Figure 4.5 is a plot of the final position errors. The bias caused by the choice of anchor position is also apparent: the arcs from Figure 4.3 can be discerned on the scatter plot in Figure 4.5.

E. PROCESS FLOW RESULTS WITH DISTANCE ERROR CONSIDERED

This section presents the results of the process assuming that there are errors in the "true" distances, g_{ij}. The same error measurements are used to determine the effectiveness of the algorithm. Table 4.3 and Table 4.4 list the average distance error measurements and the average position error measurements versus the standard deviation of the errors applied to the g_{ij}. In each case, the errors decrease in the same way as with errorless g_{ij}. However, the benefits of the algorithm decrease with increasing error levels as noted by the increase in the error metrics at all steps of the process as standard deviation of the error increases. A clear illustration of this increase is given by Figure 4.6

and Figure 4.7. These graphs show the nature of the error metric increase. This result is a measure of how well the reshaping algorithm converges when the minimum point of the error surface does not correspond to an error of zero.

For the final average distance error shown in Figure 4.6, the increase is approximately linear with respect to the standard deviation of the noise applied to g_{ij}. The equation for a linear fit of the data are given on the plot. Convergence of the algorithm is not affected until the standard deviation of the noise exceeds one nautical mile (2000 yards). For all cases the step size α was consistently set to one thousand times the sum of the g_{ij} to obtain these results. By adaptively lowering step size, convergence can be maintained, but this results in a larger number of iterations necessary to reach the minimum error point. There is a corresponding increase in the convergence time and computational cost as the number of iterations increase. This increase in time and cost offsets the positive aspects of using an adaptive step size adjustment.

For the average position error, shown in Figure 4.7, the increase is more than linear with respect to the error standard deviation. There are two factors that effect this relationship. First, the results of average distance error are reflected in the position errors. If the field is not of the proper shape, there will be an increase in the position error measurement. This contribution averaged across many trials should tend to be linear. This is so because, for a given angle of rotation, a linear increase in inter-buoy distance error would contribute linearly to an error in the distance between positions. The second factor is an error from an increase in the rotation of the field. The combination of these two errors creates a relationship that appears exponential. The equation for an exponential curve that fits the data is shown on the plot.

SD of error in g_{ij} (yards)	Average Distance Error by Preceding Step (yards) *indicates divergence during a shaping algorithm				
	initial	intra-group	inter-group	reshaping	antirotation
250	895.5	771.4	413.5	70.2	70.2
500	880.2	773.3	459.4	140.8	140.8
750	899.5	831.7	511.5	210.6	210.6
1000	864.3	830.9	563.3	278.9	278.9
1250	884.5	908.5	646.4	343.5	343.5
1500	900.3	963.6	703.3	424.9	424.9
1750	883.0	992.1	794.0	496.9	496.9
2000	848.3	1010.9	856.1	577.5	577.5
2500	897.3	1153.6	1046.2	*	*

Table 4.3. Average Distance Error Results for g_{ij} with Error

SD of error in g_{ij} (yards)	Average Position Error by Preceding Step (yards) *indicates divergence during a shaping algorithm				
	initial	intra-group	inter-group	reshaping	antirotation
250	1242.5	1190.2	694.3	351.8	365.6
500	1226.4	1195.5	727.1	416.5	421.3
750	1229.6	1234.4	801.6	489.6	513.7
1000	1210.8	1233.2	870.5	587.5	602.4
1250	1246.3	1323.0	1022.9	710.8	741.1
1500	1245.6	1355.7	1067.4	837.7	832.0
1750	1231.2	1397.4	1200.4	936.6	941.2
2000	1193.0	1425.1	1283.1	1126.4	1031.9
2500	1245.7	1572.8	1570.2	*	*

Table 4.4. Average Position Error Results for g_{ij} with Error

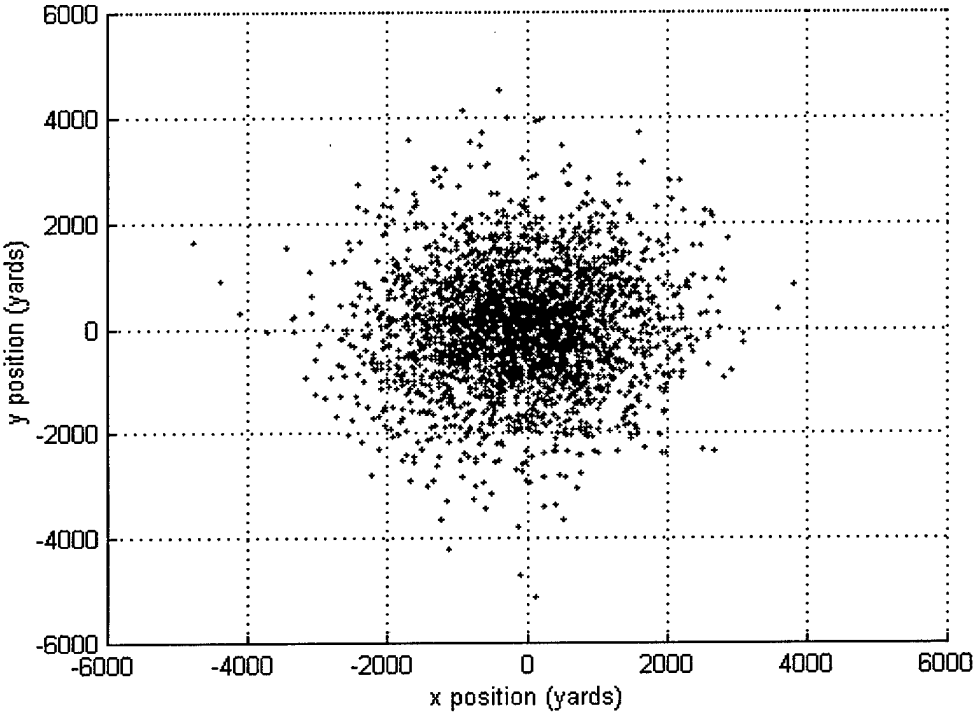

Figure 4.4. Scatter Plot of Initial Position

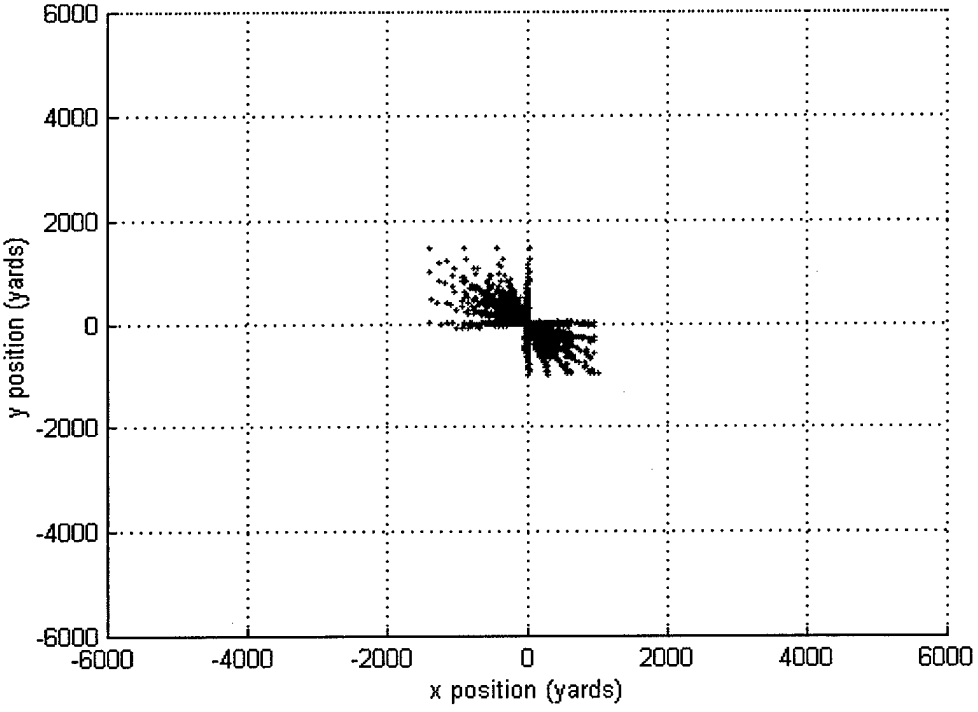

Figure 4.5. Scatter Plot of Final Position

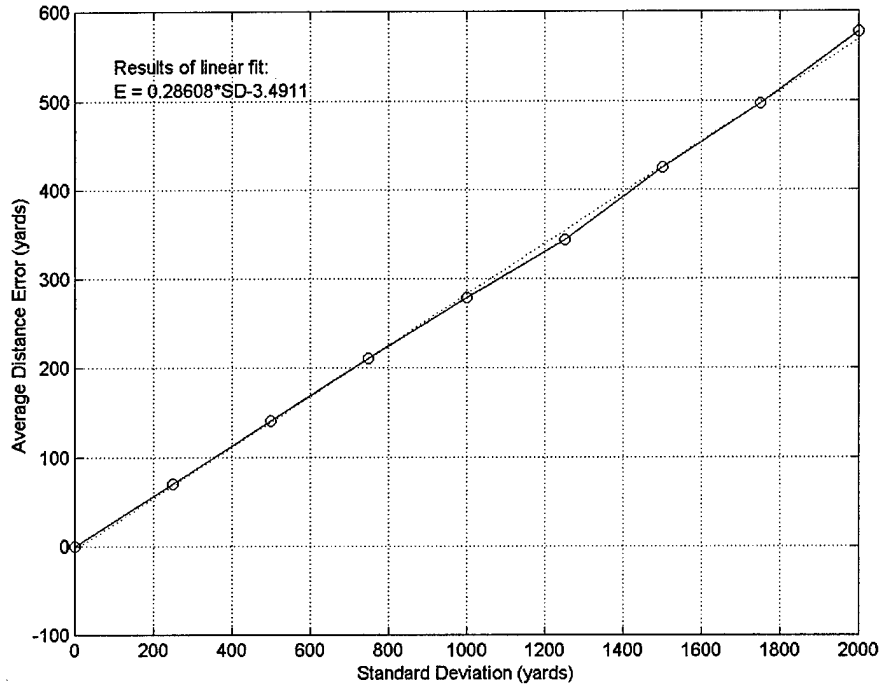

Figure 4.6. Relation Between Average Distance Error and Error in g_{ij}

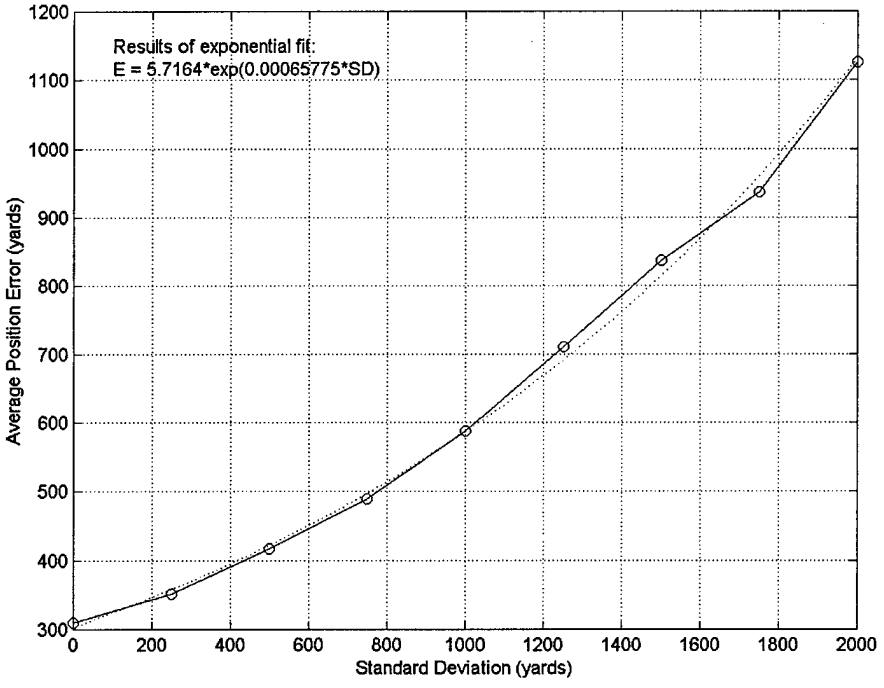

Figure 4.7. Relation Between Average Position Error and Error in g_{ij}

V. DETECTION AND CLASSIFICATION SUBMODULE

A. DETECTION AND CLASSIFICATION PROCESSING OVERVIEW

The Detection processing used is based on the software currently employed in the P3 aircraft with some modification [Ref.1]. An overview of the process flow that was developed is presented in Figure 5.1. This chapter addresses the first five steps of the process flow. Analysis to support the last step was also performed. Those results are classified and are presented in Appendix C (published as a separate document). A list of

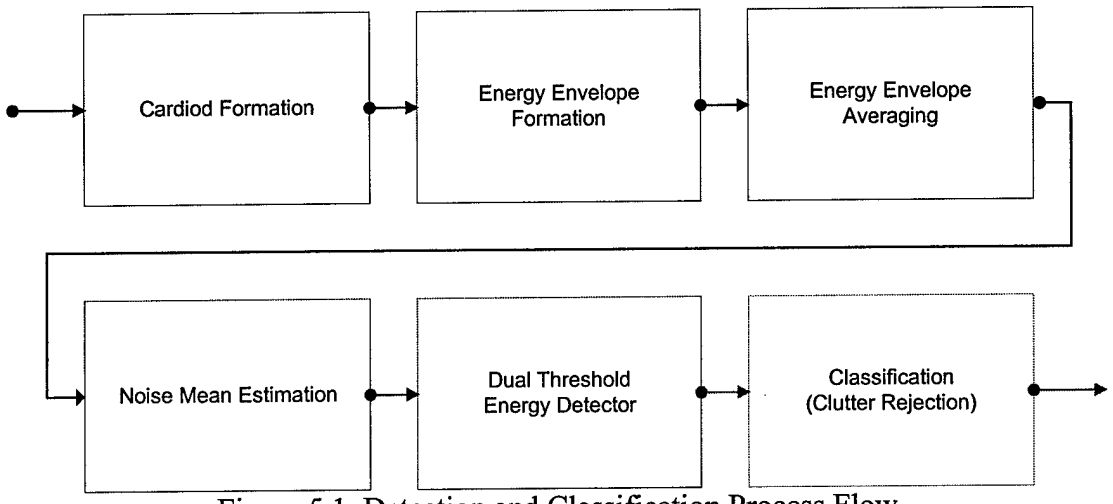

Figure 5.1. Detection and Classification Process Flow

the MATLAB files used is given in Appendix B.

B. DETECTION PROCESSING IMPLEMENTATION

1. Cardiod Formation

Recall that the DIFAR data is derived from three basic sensor elements: omnidirectional, north/south and east/west. These are denoted by $x_O(n)$, $x_{NS}(n)$, and $x_{EW}(n)$ respectively (see Figure 5.2). The sensor element inputs can be used to form cardiod sensitivity patterns using the following formulas:

$$
\begin{aligned}
y_N(n) &= x_O(n) + x_{NS}(n) \\
y_S(n) &= x_O(n) - x_{NS}(n) \\
y_E(n) &= x_O(n) + x_{EW}(n) \\
y_W(n) &= x_O(n) - x_{EW}(n) \\
y_O(n) &= x_O(n)
\end{aligned}
\tag{5.1}
$$

where the $y_d(n)$ are outputs corresponding to cardiod shaped beams in the indicated direction d (d = N, S, E, W). The omnidirectional signal is passed on as $y_O(n)$ for further processing alongside the directional signals.[Ref.1]

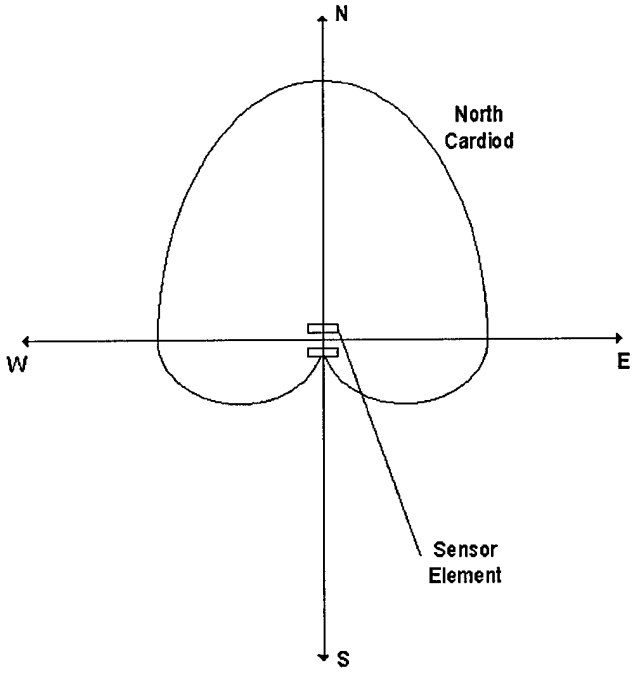

Figure 5.2. Example of a Cardiod Shaped Beam (Not to Scale)[Ref.1]

2. Energy Envelope Formation

Each of the cardiod outputs is processed in time and frequency to develop time series outputs in certain bands of interest suitable for subsequent detection processing. The processing in terms of time and frequency is illustrated in Figure 5.3 and is explained as follows.

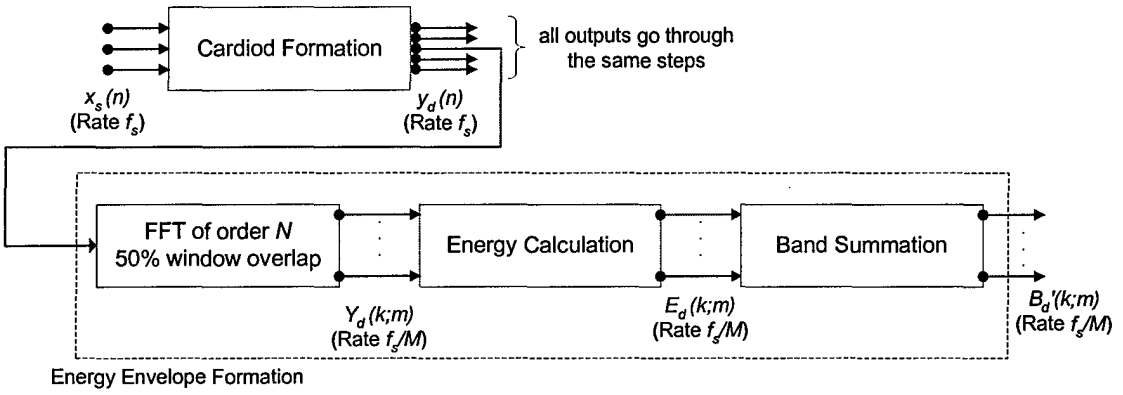

Figure 5.3. Data Flow from Submodule Entry to Band Formation

The first step in processing the data is to take the Discrete Short Time Fourier Transform (DSTFT) of the directional time series data using a Fast Fourier Transform (FFT) program. The transform is applied to the omni signal and each of the cardiods in the following manner:

$$Y_d(k;m) = \sum_{n=0}^{N-1} y_d(Mm+n)e^{-j2\pi kn/N} \tag{5.2}$$

where N is the number of points in the FFT (width of the window), M is the number of points the window is moved for each FFT computation, k is an index in frequency and m is an index in time. The relationship between M and N is determined by the desired overlap. For this implementation, an overlap of 50% is chosen, i.e., $M=N/2$. $Y_d(k;m)$ is a time-frequency representation of the data commonly known as a sonagram, or simply a "gram".[Ref.1,4]

The above processing results in a change from five time series data streams to $5N$ data streams downsampled by a factor of M. One interpretation of the DSTFT is that the

resulting streams represent the output of filters with a sinc-shaped frequency response. From this viewpoint, the transform is a bank of narrowband filters. Because we are interested in the response in specific bands, the energy is summed over sets of bins:

$$\beta'_{d,i}(m) = \sum_{k=l_i}^{u_i} |Y_d(k;m)|^2 \tag{5.3}$$

where i is the band index and u_i and l_i are the upper and lower limits for the bins to be summed. This provides a measurement of the energy in the band of interest. The resulting time series is $\beta'_{d,i}(m)$ for band i, where $1 \le i \le P$ and P is the number of bands, is referred to as the "energy envelope" of the signal. Note that this definition permits overlap between bands and the index i does not necessarily imply any ordering in frequency. The limits l_i and u_i which define the bands are reported in Appendix C.[Ref.1,4]

3. Energy Envelope Averaging

The resulting sampling rate f_S/M is typically very high compared to the duration of a detection event. Therefore, prior to the detection step, the data are passed through a first order smoothing filter:

$$\beta_{d,i}(m) = (1-c)\beta'_{d,i}(m) + c\beta'_{d,i}(m-1) \tag{5.4}$$

where c is a parameter related to the integration time of the filter. The choice of the value for c is discussed in Appendix C. This promotes accuracy in the subsequent steps by removing noise from the process. All remaining steps use the smoothed output $\beta_{d,i}(m)$.[Ref.1]

4. Noise Mean Estimation

Under normal circumstances, the ocean is a dynamic environment; background noise levels change over time. During an EER mission, this background noise is compounded by the energy from each exploder buoy ping event and the reverberation that results. A return from a target of interest will be very difficult to detect unless we have knowledge of the background noise and reverberation effects in which it appears.

Consequently, a statistic known as a Noise Mean Estimate (NME) is calculated for each data point of the energy envelope, using a *two-pass* process. This statistic provides a dynamic estimate of the background noise that is used for thresholding in the subsequent energy detector.

The algorithm for NME is as follows. On the first pass, an initial estimate of the noise is made by averaging $\beta_{d,i}(j)$. This estimate has the form:

$$\eta'_{d,i}(m) = \frac{1}{2b}\left[\sum_{j=m-a-b}^{m-a-1}\beta_{d,i}(j) + \sum_{j=m+a+1}^{m+b+a}\beta_{d,i}(j)\right] \tag{5.5}$$

where a and b are positive integers chosen to form the areas shown in Figure 5.4. The idea is to obtain an estimate of the background level without including any signal that might be present at or near the point m.[Ref.1]

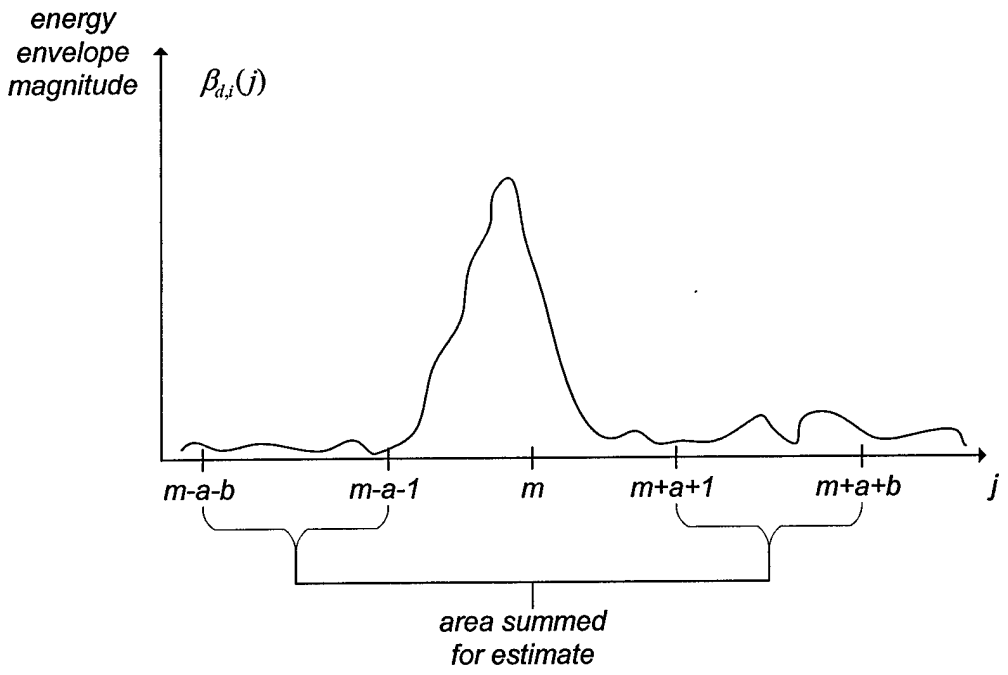

Figure 5.4. Example of Windowing for NME Calculation

A single pass would allow a spike in the windowed region to raise the NME at point m. Once again, possible sources of spikes are returns from targets, returns from non-target objects, or excessive reverberation. An attempt is made to avoid this adverse effect by clipping the original signal according to the following expression:

$$\hat{\beta}_{d,i}(m) = \begin{cases} \beta_{d,i}(m) & \beta_{d,i}(m) < a_c \cdot \eta'_{d,i}(m) \\ a_r(\eta'_{d,i}(m)) & \text{otherwise} \end{cases} \tag{5.6}$$

where the constants a_C and a_r are the clipping constant and the replacement constant, respectively. This thresholding uses the initial NME pass to remove spikes from the average energy envelope that cause an unwarranted increase in the noise estimate. The clipped sequence, denoted by a circumflex, is a temporary result which is discarded after this step. It does not permanently replace the average energy envelope $\beta_{d,i}(m)$. The clipped sequence is used to generate the final NME:

$$\eta_{d,i}(m) = \frac{1}{2b}\left[\sum_{j=m-a-b}^{m-a-1} \hat{\beta}_{d,i}(j) + \sum_{j=m+a+1}^{m+b+a} \hat{\beta}_{d,i}(j)\right], \tag{5.7}$$

where the parameters a and b are the same as those in Equation 5.5. The bias introduced by any signal returns is thus lessened.[Ref.1]

5. Dual Threshold Energy Detector

In order to create a detection event, a dual threshold energy detector is used. The output of each of two threshold detectors is given by:

$$T^u_{d,i}(m) = \begin{cases} 1 & \beta_{d,i}(m) > a_u \cdot \eta_{d,i}(m) \\ 0 & \text{otherwise} \end{cases} \tag{5.8}$$

where a_u is the thresholding constant and u denotes the level as high or low (h or l). This creates two binary-valued output streams as shown in Figure 5.5. These two streams are then used to create the detection events as follows. Whenever the average energy signal causes a one in the high level stream, a detection event is generated. The use of the high threshold for detection lowers the probability of false alarms. The detection event is characterized by a start index and a length. The start index and the length are determined using the low threshold level, which gives a more sensitive response to an energy spike.

Consequently, a more accurate measure of the start and the stop of the event is produced. These two indexes are stored as an event that is referenced to the specific band and cardioid so that the associated data can be accessed for classification of the return.[Ref.1]

C. SIGNAL ANALYSIS FOR CLASSIFICATION PROCESSING

The classification of detection events is the most difficult portion of EER processing. Some data was provided by JHU-APL to allow analysis of known returns. Results of the analysis are provided in Appendix C.

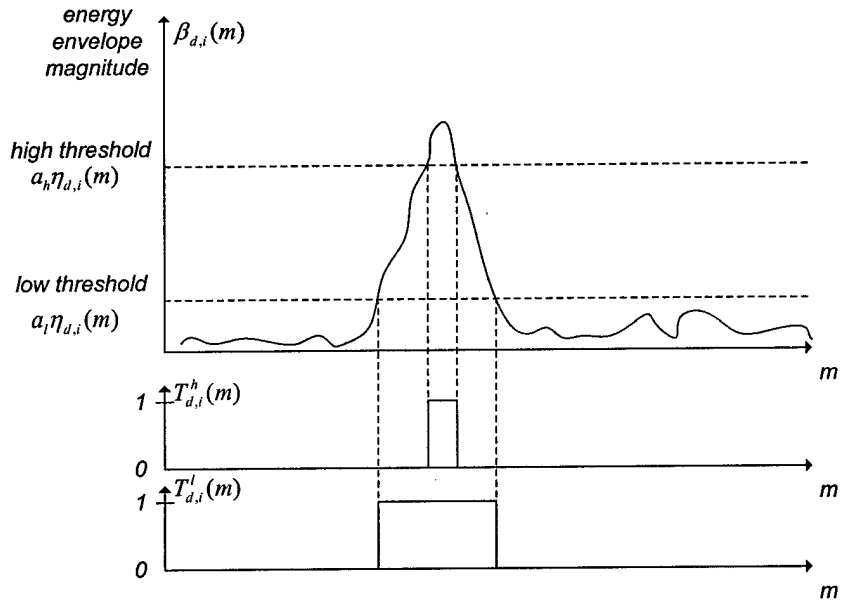

Figure 5.5. Example of Thresholding using NME

THIS PAGE INTENTIONALLY LEFT BLANK

VI. CONCLUSIONS

In this chapter, we summarize the work accomplished in this thesis. There are three main contributions that are discussed in separate sections. A fourth section that gives suggestions for future development is also included.

A. EER MODULE DESIGN

The design of an EER processing system is presented in Chapter II. The design reflects the practical experience gained from the trips to various sites dealing with EER and is designed so it can be incorporated into S2K. Utimately, the merit of the proposed design will need to be borne out in the implementation. The task of implementation represents a significant amount of development. A fundamental step for this development is the acquisition of EER mission packages for processing. The data format will influence the implementation of the submodules and may introduce new difficulties that have not been foreseen at this early stage. A detailed design of the user interface still remains to be completed.

B. VIRTUAL BUOY REPOSITIONING

A virtual buoy repostitioning algorithm was developed and coded in MATLAB. In testing, it was found that the buoy field is a closer approximation of the true buoy positions after the algorithm is applied. In the first tests, it was assumed that the distances g_{ij} derived from the time of arrival data were without error. Later, gaussian noise was added to these distances. The algorithm was found to converge until gaussian error with standard deviation of 2500 yards was added to the g_{ij}. Since this is more than the error that would be expected in the measured data, we believe that the algorithm will be a useful addition to the post-mission processing module.

C. DETECTION AND CLASSIFICATION

A detection algorithm was developed in MATLAB following procedures outlined in Reference 1. It was tested using data provided by JHU-APL that included confirmed target returns. Analysis of the return signals provided by JHU-APL was also completed and these results are presented in Appendix C.

D. SUGGESTIONS FOR FUTURE DEVELOPMENT

Although the virtual buoy repositioning algorithm is a useful technique, some refinements are possible. While the current implementation definitely improves the position of the buoys, it still leaves some error. It is impractical to expect this error to reach zero. It is practical, however, to believe that a better method, an additional step, or some new piece of data exists that can bring the error closer to zero. This area is ripe for further development.

Another area of concern is the testing of the virtual buoy repositioning algorithm on real data. Currently, all of the data is produced artificially given the starting assumptions. The true test of the algorithm will be convergence using data from an EER mission package. This also entails work on the data input submodule to get the information into the module from S2K and the creation of the energy detector that will determine the TOA data. A mission data package was not available for the initial implementation of buoy repositioning. This means that the true test of virtual buoy repositioning is yet to come. The implementation of the input routines and testing using actual mission data is an area for further development.

More work is also needed in the area of detection and classification. During the course of this thesis, we were able to develop only a few results. While algorithms have been developed, they are based on the current software in use. There is much room for refinement of our methods and the development of advanced detection and classification techniques. While the analysis of target returns provided in Appendix C will aid in such development, a larger data set is also required to perform a more complete analysis.

Finally, any future development of the EER module will need to address the issues of how to incorporate the module into S2K. In the current design, most of this functionality is intended to reside in the data input submodule. However, the implications of how data transfer will be accomplished will affect the other submodules as well. Also, all the code that has been developed exists in the MATLAB environment. This code will need to be adapted to Visual C++ for use in S2K.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX A.TRIP REPORTS

This appendix is a narrative account of the trips taken in support of EER research. The intent is to describe the general knowledge that was gained during the research project. The trips will be presented in order with their specific objectives and results.

The first trip was to Whidbey Island, WA. The objective was to gain an overview of the EER mission and the input of the operators. The local BEARTRAP office coordinated the visit. The itinerary was as follows: a visit to the Post Flight Analysis (PFA) facility, a day spent following a P-3 mission from start to finish, and a round table discussion with EER operators.

The visit to the PFA facility was the most beneficial event. The EER Tactical Employment Guide was available, which described the details of mission execution. A copy of the document was later procured for the Naval Postgraduate School BEARTRAP working group. The operators were helpful, describing the methodology and systems used for the analysis that they performed. One difficulty that was reported concerned time syncronization of the system's input streams. The time synchronization portion of the Data Input submodule was based upon this problem. Another difficulty was the lack of automation of the analysis process, i.e., the detection and classification tools that are used are limited. While there is currently research in progress, specifically on systems such as Improved EER (IEER), this was noted as an important area for research. Finally, the mission planning software was assessed. The operator input was limited to only a few environmental parameters. The feedback of environmental information estimated from the mission data was noted as another area for design research.

Observation of the scheduled P-3 mission was only moderately informative. It was not an EER mission, so the benefit was limited to a general observation on aircrew briefing, operations and debriefing. The mission was also aborted after thirty minutes of flight. Consequently, little information was gained. The end of the mission turned into a round table discussion for aircrew training. None of the operators present had significant comments to make concerning EER.

The round table discussion concerning EER that occurred later was informative. The operators' main concern was the lack of feedback from the PFA facility. While other missions had a standing method of presenting results to the aircrew, the length of time necessary for EER PFA minimized the contact between the facility and the aircrew. By the time the results are determined, distance in time has turned into distance between the aircrew and the mission execution. The participants in the discussion also expressed the opinion that they lacked training methods. They seemed unaware of several tools that I learned about on later trips. This is the input that led to the concept of making the analysis results available to the aircrew. This concept is discussed in Chapter II. One should keep in mind that better analysis tools will speed up the PFA process. If the PFA process is faster, the distance between the aircrew and their mission is reduced. Consequently, they still have a personal interest in their results.

Next was a trip to Barbers Point, HI. The itinerary was similar, with the exeption that no P-3 mission was available for observation. The visit to the tactical center where mission planning and PFA are conducted was interesting; the version of planning software was newer, and used different inputs. This change accompanied with the news that another upgrade was already being developed caused the idea of environmental feedback to be temporarily postponed. The need to do research on the current and upcoming versions and to adapt the mission data into meaningful feedback was too large to address even for a design overview. Otherwise, the information presented was similar to the information from Whidbey Island.

The final trip passed through the Washington, D.C. area and then led to Jacksonville, FL. In the D.C. area the JHU-APL site was visited. Here, software was presented for the Windows NT environment that met many of the processing and classification concerns that the S2K EER design attempts to address. Another site in the D.C. area that was visited was the office of Dr. Himberger at BBN Sensor Systems & Technologies. This is where the Distant Thunder system was advertised. It is also an advanced tool for analyzing EER mission data that operated on Windows NT systems.

At this point, consideration was given to procuring one or both of these systems and adapting it to the S2K hardware.

After arriving in Jacksonville, some of these concerns were addressed at the Advanced Maritime Projects Office (AMPO), which is the sponsor for the NPS BEARTRAP group. Andy Coon, the representative for JHU-APL, was present and expressed an interest in cooperating with AMPO to export the software. However, concerns about meeting the proper release criteria for the JHU-APL software prevented the release of that code to the BEARTRAP group. Similar concerns of a proprietary nature surround the Distant Thunder algorithms. These issues have not yet been resolved.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX B.SOURCE CODE FILES CREATED

A. SOURCE CODE FILES FOR VIRTUAL BUOY REPOSITIONING

anti_rotation.m

buoy_field_init.m

distMatGen.m

group_move_pv6.m

intra_move_fukunaga.m

monte_carlo_script.m

plot_buoy_field.m

results_script.m

shape_move_fukunaga.m

single_run.m

B. SOURCE CODE FILES FOR DETECTION AND CLASSIFICATION

band_former.m

detector.m

eer_demo.m

eer_process.m

envelop_maker.m

integrator.m

nme.m

plotBuoy.m

plotBuoyChannel.m

plotBuoyRib.m

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

1. RDA Inc., Technical Report, Naval Air Warfare Center, Contract N68335-96-C-0204, *Functional Overview for the Extended Echo Ranging (EER) Processing Mode*, 6 January 1999.

2. Technical Manual, Naval Air Systems Command, NAVAIR 28-SSQ-500-4-1, *Technical Manual for Sonobuoys: ASW Acoustic Systems/Sonobuoys*, 23 August 1996

3. Shields, M., "Preliminary Report 3 Beartrap Post Mission Processing System," Naval Postgraduate School, Monterey, CA, February 1998.

4. Online Help: Orion II, version 2.02, CD-ROM, Maritime Surveillance Associates, Inc., 1999

5. Mullins, M., *Target Tracking in the Automated Quick Look System*, Master's Thesis, Naval Postgraduate School, Monterey, CA, December 1999.

6. Fukunaga, K., *Introduction to Statistical Pattern Recognition*, Academic Press, New York, 1972.

7. Therrien, C.W., *Discrete Random Signals and Statistical Signal Processing*, Prentice Hall, Englewood Cliffs, NJ, 1992.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center 2
   8725 John J. Kingman Rd., STE 0944
   Ft. Belvoir, Virginia 22060-6218

2. Dudley Knox Library 2
   Naval Postgraduate School
   411 Dyer Rd.
   Monterey, CA 94943-5101

3. Chairman, Code EC 1
   Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Monterey, CA 93943-5121

4. Professor Murali Tummala, Code EC/Tu 3
   Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Monterey, CA 93943-5121

5. Professor Charles Therrien, Code EC/Ti 2
   Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Monterey, CA 93943-5121

6. Dr. Michael Shields, Code EC/Si 2
   Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Monterey, CA 93943-5121

7. Engineering and Technology Curricular Office, Code 34 1
   Naval Postgraduate School
   Monterey, CA 93943-5109

8. Officer in Charge 1
   Patrol Wings Atlantic Division
   Advanced Maritime Projects Office
   NAS Jacksonville, FL 32212

9. LT John C Danks 1
   176 Chahyga Way
   Loudon, TN 37774