NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

UNDERSEA NAVIGATION VIA A DISTRIBUTED ACOUSTIC COMMUNICATIONS NETWORK

by

Matthew J. Hahn

June 2005

Thesis Advisor:	Joseph A. Rice
Co-Advisor:	Clyde L. Scandrett

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			*Form Approved OMB No. 0704-0188*
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** June 2005	**3. REPORT TYPE AND DATES COVERED** Master's Thesis	
4. TITLE AND SUBTITLE: Title (Mix case letters) Undersea Navigation via a Distributed Acoustic Communications Network			**5. FUNDING NUMBERS**
6. AUTHOR(S)			
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) N/A			**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 Approved for public release; distribution is unlimited			**12b. DISTRIBUTION CODE** A

13. ABSTRACT (maximum 200 words)

Acoustic modems are the basis for emerging undersea wireless communications networks. US Navy Seaweb technology offers an opportunity to perform undersea navigation and tracking by virtue of node-to-node ranging measurements acquired as a by-product of the acoustic communications protocol. A simple localization algorithm is developed and verified with synthetic data and is then tested with an Unmanned Undersea Vehicle (UUV) during an experiment at sea.

14. SUBJECT TERMS undersea navigation, acoustic communications network, ARIES, UUV, Seaweb, network localization, telesonar, navigation, tracking			**15. NUMBER OF PAGES** 70
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UL

NSN 7540-01-280-5500

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

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

Undersea Navigation via a Distributed Acoustic Communications Network

Matthew J. Hahn
Ensign, United States Navy
B.S. United States Naval Academy, 2004

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ENGINEERING ACOUSTICS

from the

NAVAL POSTGRADUATE SCHOOL
JUNE 2005

Author: Matthew J. Hahn

Approved by: Joseph A. Rice
Thesis Advisor

Clyde L. Scandrett
Co-Advisor

Kevin B. Smith
Chair, Engineering Acoustics Academic Committee

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Acoustic modems are the basis for emerging undersea wireless communications networks. US Navy Seaweb technology offers an opportunity to perform undersea navigation and tracking by virtue of node-to-node ranging measurements acquired as a by-product of the acoustic communications protocol. A simple localization algorithm is developed and verified with synthetic data and is then tested with an Unmanned Undersea Vehicle (UUV) during an experiment at sea.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

First and foremost, I would like to thank my advisor, Joseph Rice, who guided me through my research from start to finish. He helped me to choose my topic and provided crucial insights, pointed me toward expert resources, and offered the best opportunities to observe and participate in Seaweb test exercises.

I would also like to thank both Professor Clyde Scandrett and Professor Emeritus David Book for their invaluable help in developing and testing my localization algorithms.

The staff of SPAWAR Systems Center in San Diego offered more assistance than I could have ever hoped for in terms of understanding the rudiments of Seaweb networks. Chris Fletcher and Bob Creber, and Bill Marn taught me the fundamentals of the system database, and Paul Baxley was always quick to offer his assistance on matters of sound propagation data and analysis for the network environments.

Additionally, Professor Tony Healey, Doug Horner, Sean Kragelund, and Ben Wring of the ARIES UUV team helped me to learn the basics of vehicle operations.

The crew of the research vessel *Cypress Sea* provided superb logistics for testing. Benthos Inc., Falmouth Scientific Inc., and Sub Sea Sonics provided Seaweb hardware for the experiment, and Dr. Tom Swean of the Office of Naval Research sponsored this experiment. The SSC San Diego Fellowship Program sponsored my research.

The faculty and staff of the Physics Department at NPS helped me to garner a suitable basis of knowledge for completing this thesis. In particular, I would like to acknowledge Professors Daphne Kapolka, Kevin Smith, Thomas Hofler, Steve Baker, and Bruce Denardo for their help in my more advanced acoustics courses.

Last, but certainly not least, I would like to thank my colleagues LT Hannah Kriewaldt, LT Spyridon Dessalermos (Hellenic Navy), ENS Sean Ouimet, and ENS Chris Hurt, who provided peer advice and moral support without which I could not have completed my degree.

THIS PAGE INTENTIONALLY LEFT BLANK

I. SEAWEB: A BRIEF OVERVIEW

A. SYSTEM BASICS

Seaweb is a set of deployable acoustic transducers that forms an underwater acoustic communications network. Detailed system information regarding the Seaweb system can be found in reference [1], but we shall briefly review some basic theory, particularly that which pertains to underwater navigation.

This thesis considers the viability of tracking an undersea vehicle using a grid of fixed nodes. Similar endeavors have been pursued as far back as 1990 [2-4] in the form of element localization of sonar arrays. More recently, localization techniques for undersea network nodes have been sought [5]. This thesis differs from previous work in that we attempt to localize a *mobile* network node. Figure 1 depicts an experimental implementation of this concept.

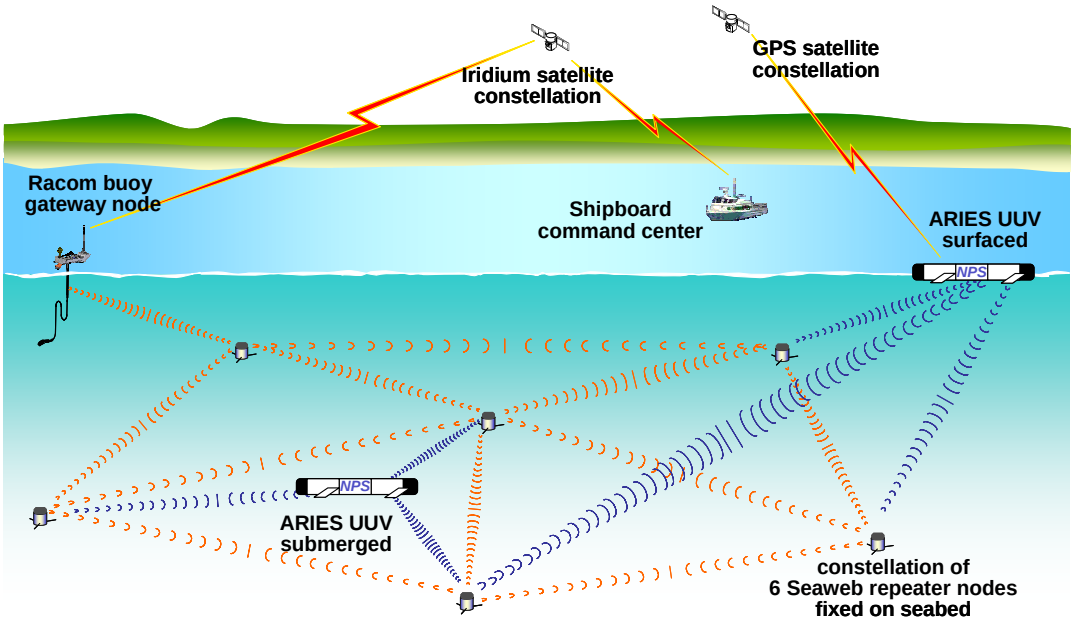


Figure 1. The May 2005 Seaweb ARIES Experiment in Monterey Bay exercised the node-to-node acoustic ranging capability of Seaweb networked modems as a mechanism for tracking the ARIES UUV mobile node relative to a fixed undersea grid. When the UUV is submerged, tracking is accomplished by triangulation from the fixed nodes. When surfaced, Seaweb tracking quality can be compared with that afforded by GPS.

1. Repeater Nodes

The heart of Seaweb communications lies in the exchange of acoustic signals among a network of repeater nodes on the sea floor. Deployment of a Seaweb system involves the placement of a set of repeaters at various locations within the operating area. Each node is outfitted as shown in Figure 2.

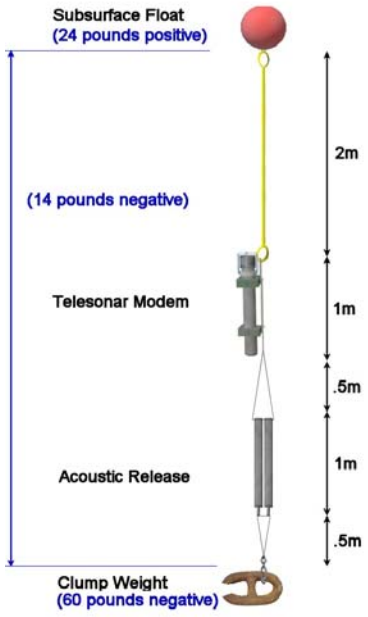


Figure 2. Seaweb Repeater Node. The repeater node is anchored to the sea floor and held 3-5 meters off the bottom by a subsurface float. The acoustic release mechanism allows retrieval of the telesonar modem following the end of an experiment.

These repeaters exchange and process acoustically-modulated acoustic data through the use of omnidirectional transducers and the through-water propagation channel. Directional transducers would increase the practical range between nodes, but the omnidirectional aspect of the current modem is convenient for this thesis, since we hope to establish links with a mobile node at an unknown position.

2. Racom Gateway Buoy

The racom (radio-acoustic communications) gateway buoy (Figure 3) provides the link between the undersea network and an operator on the surface. A mooring line attached to the bottom of the buoy also maintains a hardwire connection to a submerged

transducer for acoustic signaling to the repeater nodes. Through this hard-wire connection, signal processing techniques allow conversion of acoustic data to/from radio signals. Thus, a human operator aboard a research vessel gains access to the network via one of two electromagnetic communications methods: FreeWave line-of-sight packet radio or Iridium satellite communications. A large solar panel and a battery bank provide power to all equipment.

Figure 3. Racom gateway buoy. The racom buoy maintains a hard-wire connection to a repeater node attached to its mooring line. It provides a radio link between the undersea environment and the Seaweb operator on a surface vessel.

3. ARIES UUV

The May 2005 experiment depicted in Figure 1 introduces the ARIES UUV as a new component to Seaweb. By equipping the UUV (Figure 4) with the same telesonar hardware as the repeater nodes, we transform ARIES into a network mobile node. For this thesis, we only consider the UUV insofar as it provides us with navigational data, though a great wealth of background information has been written about this vehicle [6, 7].

Figure 4. The ARIES UUV was equipped with the same equipment as the repeater nodes.

B. RANGING DATA

1. Signal Processing

Inter-node communications yield range data as a by-product of the link-layer protocol. For a data exchange between two nodes (*i* and *j*) a ranging signal precedes the transmission of a utility packet from node *i*. A matched filter at node *j* detects this signal, which is a Hyperbolic Frequency Modulated (HFM) chirp. Node *j* then determines the time of arrival (TOA) as the peak of the matched filter response. Picking the peak response allows resolution of multipath propagation for most cases, since reflected and head waves will typically be characterized by amplitudes lower than those of direct paths.

Following a specified dwell time τ_j after detection of the ranging signal peak, node *j* replies by sending a utility packet to node *i*. The utility packet itself may contain several different types of information. For ranging purposes, however, the only relevant information is a random time delay that may be implemented in some situations. If present, the utility packet includes the random delay value as a digital parameter. Reciprocity dictates that the return travel time will equal that of propagation in the opposite direction and that the same characteristic peak will be picked. Therefore,

$$d_{ij} = d_{ji} \tag{1}$$

The total elapsed time, t_j-t_o, from the initial sending of the ping (t_o) to the reception of the echo (t_j) is measured at node *i*. Hence, we may calculate the one-way travel time d_{ij}.

$$d_{ij} + d_{ji} + \tau_j = t_j - t_o$$
$$2d_{ij} = t_j - t_o - \tau_j \qquad (2)$$
$$d_{ij} = \frac{t_j - t_o - \tau_j}{2}$$

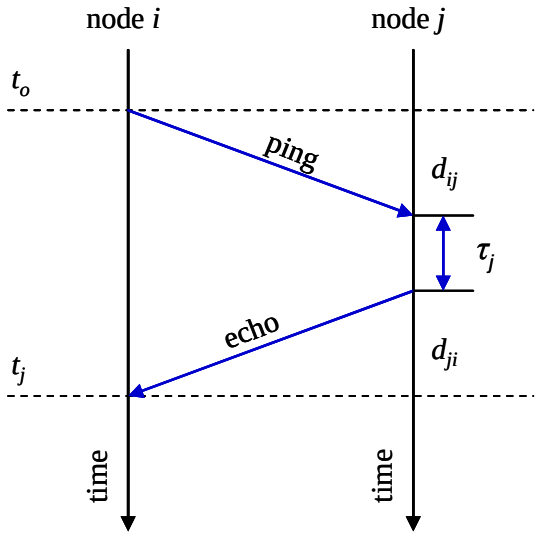


Figure 5. The Seaweb ranging process: a utility packet (e.g., ping utility packet) travels from node *i* to node *j*, then the signal experiences a dwell time at node *j*, and a replying utility packet (e.g., echo) travels back from node *j* to node *i*. We measure the sum of the three legs by taking the total elapsed time at node *i* from when the ping is first broadcast until the echo is received. The dwell time τ_j is embedded in the echo utility packet and sent from node *j* to *i*. The important result of this method is the fact that all time measurements are computed at node *i*. Thus, no clock synchronization is required.

We now compute a range by multiplying the travel time by the speed of sound propagation, for which the present software assumes the general value of 1500 m/s:

$$r_{ij} = 1500 \cdot d_{ij} \qquad (3)$$

The ranging protocol makes several assumptions which may lead to ranging errors. The most obvious assumption is the 1500 m/s value for sound speed, which clearly will not apply in all cases. Also, the range calculation itself assumes straight-line sound propagation. Lastly, the analog-to-digital converters used will result in a small degree of rounding error in calculating the travel time. Despite these assumptions, past ranging tests indicate that the accumulation of such errors in typical operating areas

generally does not exceed that of a typical hand held GPS device with a nominal error of 3 m.

2. Broadcast Ping

While a single range provides the basis for network localization, a set of three or more ranges (for a three-dimensional model, at least four ranges are required) would prove a more useful tool for fixing locations. The broadcast ping command provides this tool. Briefly, the broadcast ping orders a single node, the UUV for our case, to acquire ranges to each of its neighboring nodes. The UUV sends a single ping, akin to the ranging signal discussed earlier, to all neighboring nodes. To avoid signal interference, each node computes a random delay of up to 60 seconds. At the conclusion of its random delay, each node sends an echo signal back to the UUV, from which a range measurement is calculated as discussed in the previous section. The experiment conducted for this thesis employed a fixed grid of 6 nodes. Thus, the UUV may acquire up to 6 inter-node ranges for a broadcast ping. The lack of simultaneity among the set of ranges for a single broadcast ping poses a formidable obstacle in the tracking/navigation of fast-moving targets, but the UUV moved at a top speed of only 1.2 m/s, so we accept the motion error as part of the error budget and do not yet attempt to account for target motion in the localization algorithm.

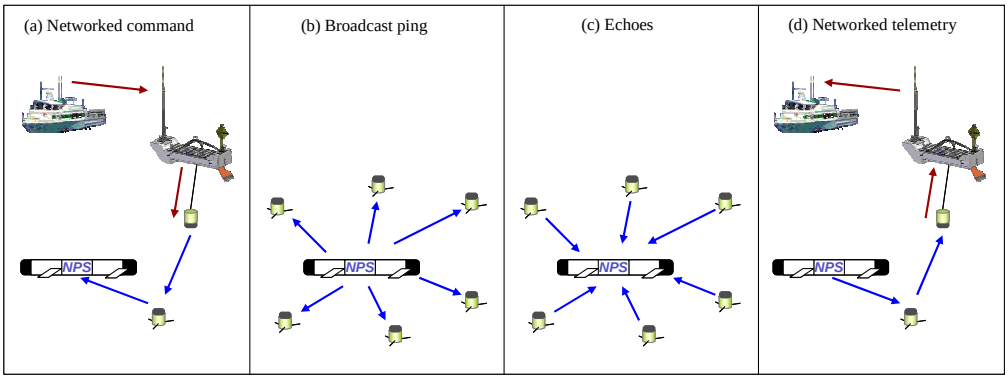


Figure 6. Broadcast Ping. The operator commands the UUV (a) to broadcast a ping (b). This elicits echoes from neighboring nodes (c). The UUV telemeters the calculated set of ranges back to the operator (d).

II. MATHEMATICAL CONCEPTS

A. RANGE CIRCLES

The solution to the positioning problem lies in simple geometric concepts. We acquire a set of between three and six known reference nodes on a 2-dimensional plane and wish to locate a target based on straight line distances from the known nodes. When considering a single range R, we know the target lies somewhere along the circumference of a circle of radius R centered at the known location. Thus, a single range yields an infinite number of possible solutions, each satisfying the equation of the range circle.

$$r^2 = (x - x_o)^2 + (y - y_o)^2 \tag{4}$$

The solutions can be described by

$$\begin{aligned} x &= x_o + \sqrt{(y - y_o)^2 - r^2} \\ y &= y_o + \sqrt{(x - x_o)^2 - r^2} \end{aligned} \tag{5}$$

where x_o and y_o are the coordinates of the reference node (center of the circle).

Although a single range holds little value for a localization problem, when it is combined with a second range from another reference node, we may then cross-fix the target. Two ranges from separate locations result in two separate range circles. Intersections of the two circles occur where the range circle equations for both nodes are satisfied. These intersections, then, represent possible solutions. One of three cases will prevail for a system of two ranges: one, two, or zero solutions.

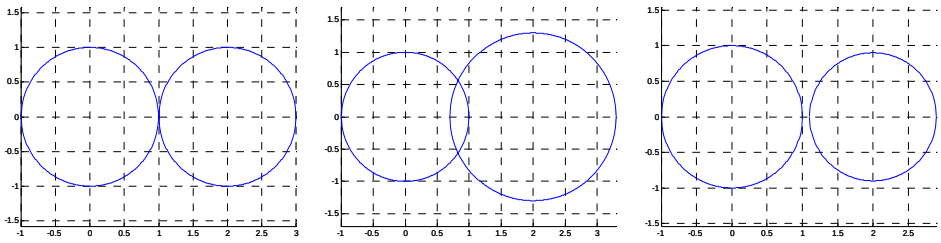


Figure 7. Three possible cases for a set of two range circles: The third case may only occur if one or both of the ranges contains error.

For a 2-dimensional system, the addition of a third range yields a unique solution. A unique solution requires that the range circles must not contain errors. For perfect data, the three circles will intersect at a single point. This thesis does not consider the 3-dimensional case, but note that an additional dimension requires an additional range, since the new dimension presents a new unknown (depth). The presence of additional ranges results in an over determined system. In the absence of range errors, additional range circles will cross through the solution

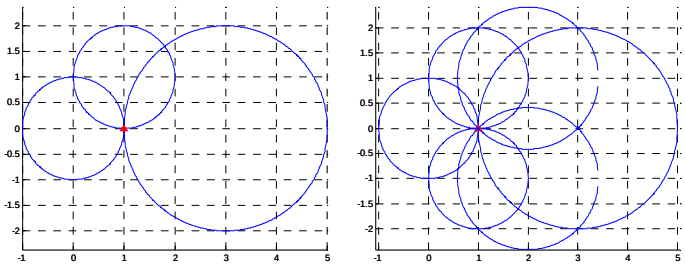


Figure 8. Three range circles suffice for target localization in two-dimensional space. Additional ranges result in an over determined system, which will prove useful for minimizing the effects of errors in the range data.

The range circle equations, while useful, do not work well when we introduce range errors. As shown by Figure 8, the addition of range errors disrupts the intersection points. The tight cluster of intersection points clearly points to the general region of the solution, but a unique intersection point no longer exists. No set of coordinates will satisfy the system of equations.

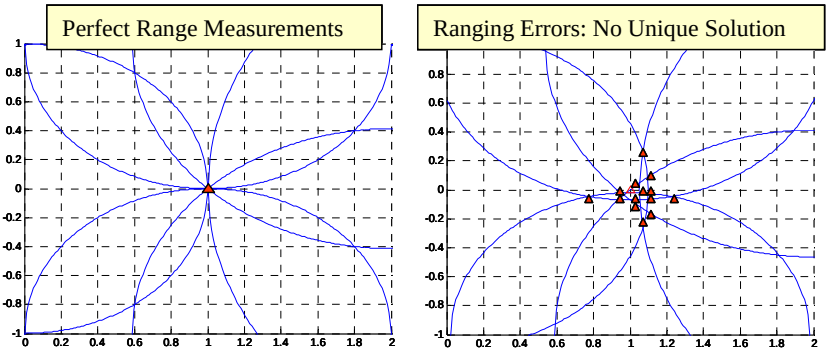


Figure 9. The addition of range errors causes the single intersection point at the solution to spread. A single solution no longer exists. We now have only a region of probable location whose size is proportional to the magnitude of the range errors and the bearing angles of the different sources.

The nature of the equations themselves also poses concerns for implementation within a computer program. Many computer tools assume a positive square root and neglect the second (negative) solution. MATLAB software, the proposed tool for algorithm implementation, exhibits this problem. Thus, multiple equations are required to define a circle. This complicates defining the system of equations.

B. TRIGONOMETRY

The failure of the circle equations in the presence of range errors requires that we search elsewhere for a localization method. A useful exercise for this search is to compute the locations of the range circle intersection points. To accomplish this we consider each pair of ranges and compute their intersection points. Basic trigonometric laws offer a straightforward route to this end.

Consider two reference nodes, i and j, located a distance r_{ij} apart. Let node i be located at the origin and node j be located on the positive x-axis. The values r_i and r_j represent target node ranges for the respective reference nodes. A triangle formed by the intersections of the three available ranges defines the target location. Note that two solutions exist because the range vectors r_i and r_j may point either above or below the line segment r_{12}. The data form two triangles for which all side lengths are known, as shown in Figure 10. The law of cosines allows rapid calculation of any of the three angles based on the following equation, where sides r_i and r_{ij} are adjacent to the angle θ and r_j defines the opposite side.

$$\theta = \cos^{-1}\left[\frac{r_i^2 + r_{ij}^2 - r_j^2}{2r_i r_{ij}}\right] \tag{6}$$

Calculation of the angle values allows simple calculation of the target position via trigonometric functions. The angle ambiguity of the *even* cosine function maintains the existence of two solutions, since θ may be positive or negative.

$$\begin{aligned} x &= r_i \cos\theta \\ y &= r_i \sin\theta \end{aligned} \tag{7}$$

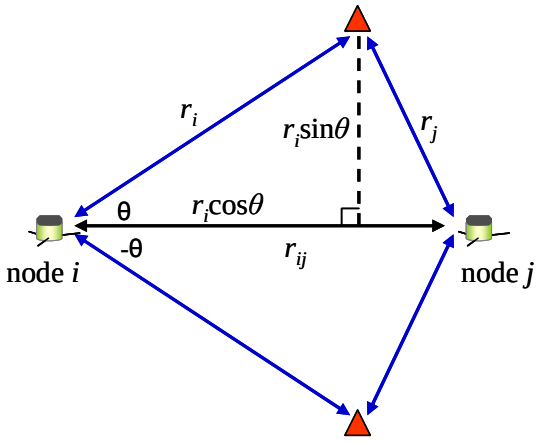


Figure 10. The law of cosines allows us to calculate the angle θ, which we use in conjunction with the side lengths to compute two possible target positions.

We have now computed two solutions for the source pair. The two solutions are images of one another, differing only in the sign of θ. This means that the x value will not change, because θ is contained within a cosine term and its sign does not matter. Likewise, the y values will have the same absolute value, because θ is contained within the *odd* sine function.

Repeating this calculation for N reference nodes yields ½N(N-1) range pairs and creates N(N-1) solutions if each range pair produces two solutions Recall that we have computed each solution pair within a separate coordinate system, where one reference node defines the origin and a second reference node defines the x-axis. To display the solutions in a meaningful fashion, each solution pair must be transformed to a common coordinate system.

C. COORDINATE TRANSFORMATIONS

The solutions of each range pair must be transformed from their respective local coordinate systems to a common reference (Figure 11). As discussed earlier, we compute solutions relative to a set of local axes with the origin defined as one source and the x axis defined as the line segment between the source pair. To transform a solution from a local reference to the common one, we perform both translational and rotational calculations.

We perform the translation by simply adding the local origin coordinates to the solution coordinates. The rotational component requires a bit more sophistication. We have already obtained the solution angles relative to the local (x') axis. Now we must

consider the angle of the x' axis itself relative to the common x axis. The arctangent of the angle between x and x' defines this angle, which we call φ. Combining φ and the solution angles $\pm\theta$ defines the solution angles relative to the common x axis.

Determination of the direction of the x' axis and resolution of angle quadrants is required. We handle both issues by carefully considering the positions of the two sources. For simplicity, we assign a number to each source and always define the local origin as the lower number of the pair. To determine the quadrant of φ, we compare the coordinates of the two reference nodes i and j to verify the angle quadrant based on relative locations.

$$\varphi' = \tan^{-1}\left|\frac{y_i - y_j}{x_i - x_j}\right| \tag{8}$$

The value of φ' represents the inter-node axis angle with respect to the common x-axis if we assume it is located in the first quadrant. We now compare the relative positions of the reference nodes to implement an angle quadrant correction.

$$\varphi = \begin{cases} \varphi' & x_i \le x_j, y_i \le y_j \\ \varphi' + \pi/2 & x_i > x_j, y_i \le y_j \\ \varphi' + \pi & x_i \ge x_j, y_i > y_j \\ 2\pi - \varphi' & x_i < x_j, y_i > y_j \end{cases} \tag{9}$$

In addition to these equations we treat the special case of a vertical x-axis, which has an infinite arctangent value and cannot be resolved by the previous equations.

$$\varphi = \begin{cases} \frac{\pi}{2}, x_i = x_j, y_i < y_j \\ \frac{-\pi}{2}, x_i = x_j, y_i > y_j \end{cases} \tag{10}$$

Now that we have the angle of the local x-axis relative to the common x-axis, we can define the solution angle γ relative to the common axis. The solution angles for the two solutions given by each range pair are defined as the sum of the relative local x-axis angle and the solution angles within the local coordinate system.

$$\gamma = \varphi \pm \theta \tag{11}$$

The complete transformation may now be performed by combining the translational and rotational components as follows:

$$x = x_o' + r_i \cos\gamma = x_o' + r_i \cos(\varphi \pm \theta)$$
$$y = y_o' + r_i \sin\gamma = y_o' + r_i \sin(\varphi \pm \theta) \quad (12)$$

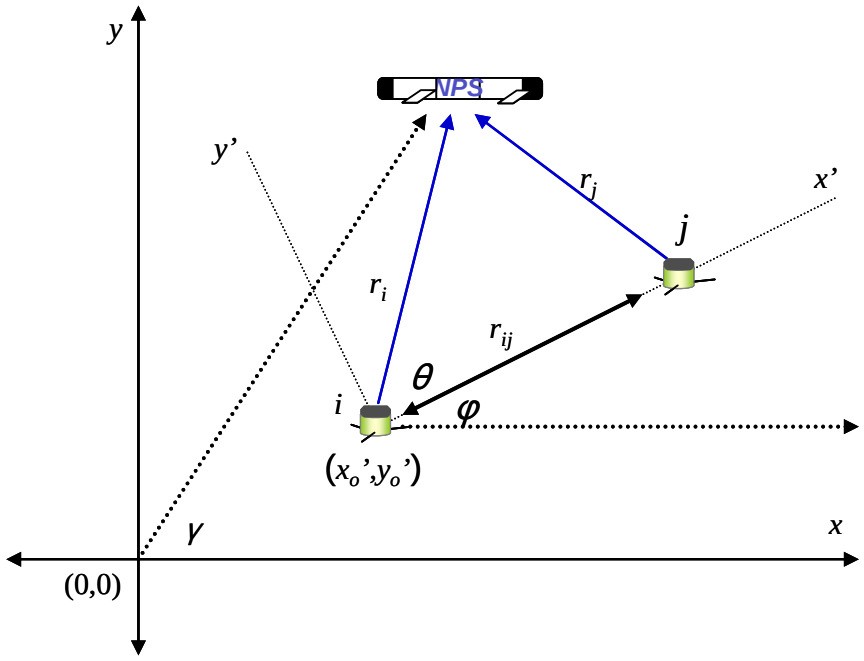


Figure 11. We perform coordinate transformations on each solution pair in order to acquire a set of all solution coordinate pairs within a common reference frame. To perform a transformation we exploit the known parameters x_o', y_o', θ, and φ to compute the solution angle γ within the common (x, y) axes.

D. WEIGHTING METHOD OF POSITION ESTIMATION

We have now computed the coordinates of a solution pair within the common reference frame. By performing the transformation on all solution pairs, a set of possible solutions is defined within a single coordinate system. If all the solutions are plotted, a tight cluster of points will clearly indicate the target position. Though the solution is obvious to the human eye, we require an automated method for pinpointing the estimated position within a computer program. This thesis uses a simple adaptive weighting method to complete this task.

The weighting method exploits the proximity of solution pairs to estimate a position. By computing the relative proximity of a single solution to all other possible

solutions, we assign a dimensionless weight to that solution. Once a weight has been assigned to each solution, we average the solutions based on their relative weights.

$$W_i = \sum_{j=1, j\neq i}^{m} \left[\frac{1}{(x_i - x_j)^2 + (y_i - y_j)^2} \right]^{\alpha} \tag{13}$$

The variable *m* refers to the number of solutions that have been calculated, and the exponential term is an arbitrary constant used to either amplify or reduce the severity of weighting. A high alpha value will drastically increase the weights of close solution pairs while decreasing the influence of distant pairs. For our purposes we choose the α value of 2, realizing it may be adjusted to alter performance. Once all weight values have been computed from (9), we sum the individual weights to find a total weight W_{total}:

$$W_{total} = \sum_{k=1}^{m} W_k \tag{14}$$

Averaging the set of solutions based on the ratio of individual weights to the total weight yields estimated position coordinates:

$$\begin{aligned} x_{est} &= \sum_{i=1}^{m} \left[\frac{W_i}{W_{total}} x_i \right] \\ y_{est} &= \sum_{i=1}^{m} \left[\frac{W_i}{W_{total}} y_i \right] \end{aligned} \tag{15}$$

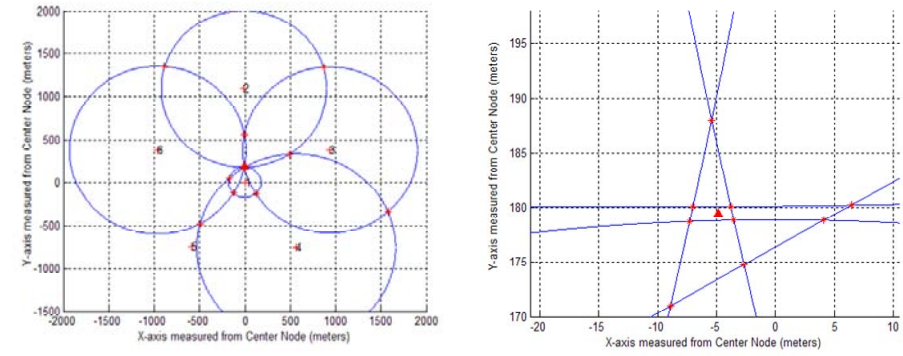


Figure 12. The figures above show the ideal outcome of the weighting method. The tightest cluster of possible solutions represents the region of probable location, and the triangle indicating the estimated target position (right figure) shows that the weighting method has chosen a position near the center of the tightest cluster.

E. SPECIAL CASES

Recall the discussion of the three possible cases for a pair of ranges (Figure 6). Until now, we have considered only the case of two non-unique solutions. This is the most common situation, but, for completeness, we consider the other two cases.

1. Unique Intersection of Two Range Circles

In rare cases, two range circles will intersect only once, resulting in a single unique solution. This occurs for scenarios where the target lies directly on a straight line intersecting the two reference nodes in consideration, as shown by Figure 13.

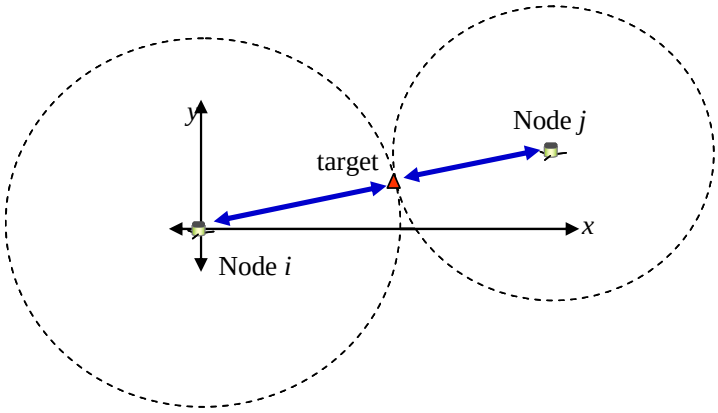


Figure 13. Rare case of a unique range circle intersection. This results in a double solution and these solutions will exhibit infinitely high weight values such that all other solutions would be neglected.

In this situation, the value of θ has decreased to zero. For a node pair exhibiting a single intersection, the algorithm will still compute two solutions, but these solutions will be located at the same point. Hence, when we calculate the weight values for all solutions, the two solutions from this pair will encounter a term in the summation of the weight equation (10) with a zero denominator. This will result in infinitely high weight values for the two solutions of this range pair; subsequently, all other solutions will be effectively neglected (11), and the final estimated target position (12) will be "focused" at a location directly over the solution pair.

Since we expect all experimental range measurements to contain at least a minimal degree of error, we know the estimated position will hardly ever be perfect, even if focused by the "double solution". However, we do expect any double solution to be located very close to the true target position, so we allow the presence of double roots

into the field of possible solutions for a given set of ranging data. Inclusion of such cases requires a minor algorithm revision, since most computer programming tools do not handle zero denominators well. To eliminate this problem, we add a small value to the denominator term of the weight equation (10). The value of this term should not affect the general performance of the weighting method, since all solutions will include it.

$$W_i = \sum_{j=1, j\neq i}^{m} \left[\frac{1}{(x_i - x_j)^2 + (y_i - y_j)^2 + C} \right]^{\alpha} \tag{16}$$

We choose C=0.01 because a term of that order of magnitude will be lower than what we expect in terms of typical solution separation distances, which will likely be on the order of a few meters for very good ranging data.

2. Non-intersecting Pair of Range Circles

a. Two Cases for Non-intersecting Circles

Sometimes a combination of range error and target position results in non-intersecting range circles. As in the previous section, this situation can happen for target positions on or close to the straight line drawn between two reference nodes (Figure 13). One or both of the target ranges must then be calculated to a value shorter than the true distance in order for the circle pair to lack an intersection point. Non-intersecting circle pairs are not limited to this situation, however. Two reference nodes with a similar target bearing may also produce such a result. Figure 14 illustrates this scenario.

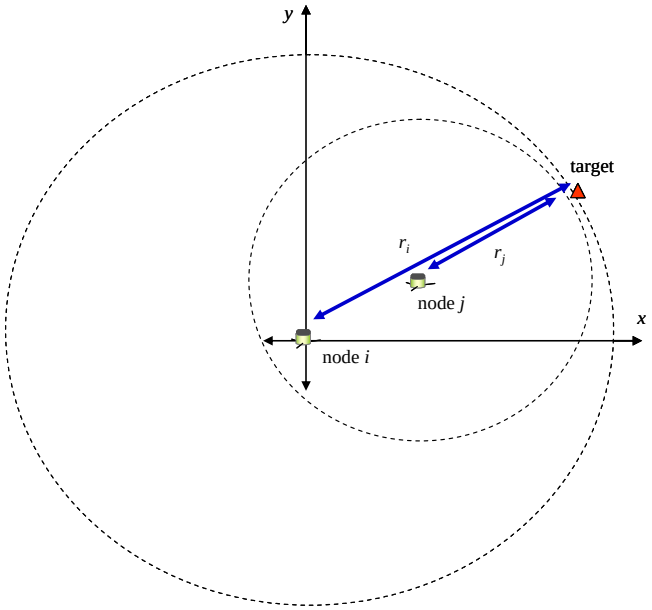


Figure 14. Non-intersecting range circles with similar target bearings. This situation requires a short value for r_j, a long r_i value, or both. Depending on the magnitude of range errors, the node pair may still be able to produce good solutions.

b. Defining an Improvised Solution Pair

The fact that a circle pair does not intersect should not lead us to believe that good solution estimates cannot be drawn from the data. In fact, choosing the point on each circle along the straight line distance between the reference nodes allows us to include such cases intelligently.

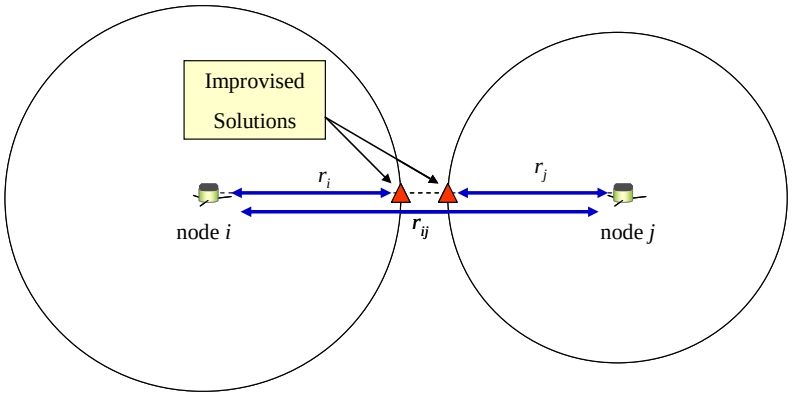


Figure 15. For non-intersecting circles, we define two solutions, one located on each range circle at the point of greatest proximity. Using this method, circles with only a small distance between will produce more heavily weighted solutions than circles separated by large distances.

c. Implementation

Implementation of this method requires significant additions to the computer program. To begin, we determine the locations of our projected solutions relative to the reference nodes. If both r_i and r_j are less than the inter-node distance r_{ij}, then we know that the solutions will be located between the reference nodes, as shown in Figure 14. Likewise, if r_i is greater than r_{ij} and r_j is shorter, then the solutions will be located along the inter-node axis beyond node *j*. The converse is also true, and Figure 13 illustrates this concept.

Mathematically defining the improvised solutions involves finding the coordinates of a point on a line of known slope (φ), located a given distance (range) from a known point (reference node). Because the calculations for the two cases are nearly identical, we will perform example calculations for only one of cases, those in which the improvised solutions are located between the reference nodes (Figure 15). Figure 16 illustrates the method used to calculate the solution coordinates.

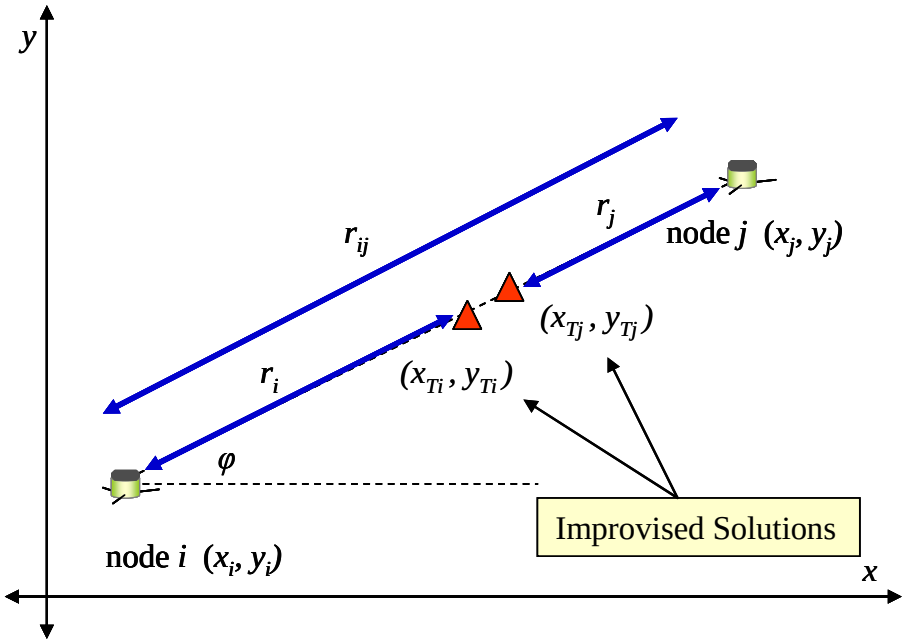


Figure 16. Mathematical concepts underlying improvised solutions for non-intersecting range circles.

We employ the φ values calculated earlier (section C.) to determine exactly where each solution should be located relative to the reference nodes. For the scenario given by Figure 16, we use the following equations to solve for the solution coordinates.

$$\begin{aligned} x_{Ti} &= x_i + r_i \cdot \cos\varphi \\ y_{Ti} &= y_i + r_i \cdot \sin\varphi \\ x_{Tj} &= x_j - r_j \cdot \cos\varphi \\ y_{Tj} &= y_j - r_j \cdot \sin\varphi \end{aligned} \tag{14}$$

As with all other solutions, pairs of improvised solutions are included in the revised weight equation (13). Having considered all three cases for a pair of range circles and devised methods for computing two solutions regardless of which case occurs, we may more readily calculate the number of solutions. If N is the number of solutions, we know that k possible combinations of reference nodes are available.

$$k = \frac{1}{2} \cdot (N) \cdot (N-1) \tag{15}$$

If we have k possible combinations of reference nodes, and each node pair produces two solutions, then the total number of possible solutions (used in the weight equation summations (11,12,13), m, is defined as a function of N.

$$m = 2k = (N) \cdot (N+1) \tag{16}$$

As a result, the number of computed solutions for consideration of a single position is a function of the number of ranges only.

III. SIMULATION

A. CREATION OF SYNTHETIC DATA

Testing and refinement of the positioning algorithm were accomplished through a series of MATLAB simulations. These simulations generated hypothetical target nodes at randomly chosen locations. Simulating large numbers of cases revealed even the rarest of scenarios, and analyzing the results provided a systematic method for debugging and refining the positioning algorithm. Having achieved an acceptable level of performance for random numbers, further simulations treated a series of three hypothetical tracks. Simulating actual tracks served as the final test of algorithm performance before field experimentation.

The first simulation computed solutions for a set of 50,000 random positions spread among a field of 6 nodes as shown:

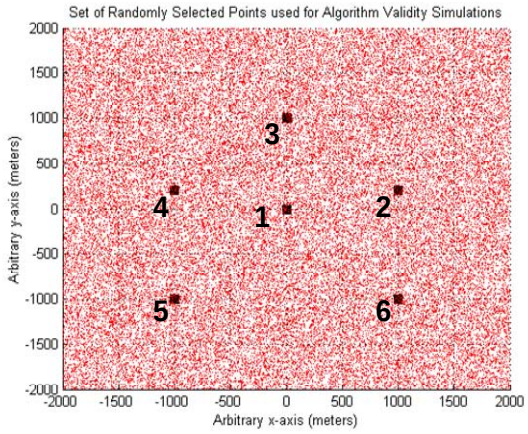


Figure 17. The large squares represent the fixed nodes, and the smaller dots shows a typical distribution of the 50,000 randomly created positions used for simulations.

The MATLAB program created a set of range data for each of these points by calculating the straight line distance from each reference node to the random location. The x_i and y_i values indicate the coordinates of the 6 fixed nodes, and x and y represent the random position true coordinates.

$$r_i = \sqrt{(x - x_i)^2 + (y - y_i)^2} \tag{17}$$

In order to model actual range measurements and to account for errors caused by assumptions, we perturb each range value by applying random error. For initial simulations, errors were held within ±10 meters of true ranges. Note that the rand(*n*) function in MATLAB creates an (*n* x *n*) matrix of random values between 0 and 1.

$$r_i = 20 \cdot (0.5 - rand(1)) \cdot r_{i(true)} \tag{18}$$

For some simulations, we wish to offset the data to give only short or long errors. The following equations produce errors from -20:0 meters and 0:20 meters, respectively:

$$\begin{aligned} r_i &= -20 \cdot [1 - rand(1)] \cdot r_{i(true)} \\ r_i &= 20 \cdot [1 - rand(1)] \cdot r_{i(true)} \end{aligned} \tag{19}$$

B. PRELIMINARY IMPLEMENTATION

1. Simulation 1 Results

The results of the first simulation indicated fundamental problems within the algorithm code. The majority of problems did not stem from the theory of the algorithm, but, rather, from flawed implementation.

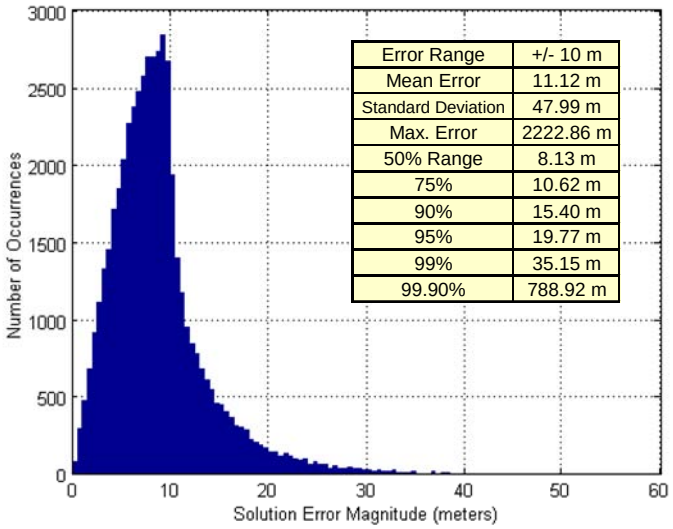


Figure 18. Solution Error Distribution for Simulation 1. The algorithm calculated accurate position estimations for the vast majority of 50,000 simulated locations. However, a number of outlying solutions with errors over 1 km suggested fundamental flaws, which were discovered within the MATLAB code. The range percentages indicate error magnitudes within which a certain percentage of the solutions have been estimated. For example, half of the solutions have been calculated to within 8.13 meters of the true solution.

2. Debugging the MATLAB Script File

Debugging was performed via detailed simulation of the cases for which poor solutions had been found. This method quickly revealed a software bug, which is illustrated by the following figure:

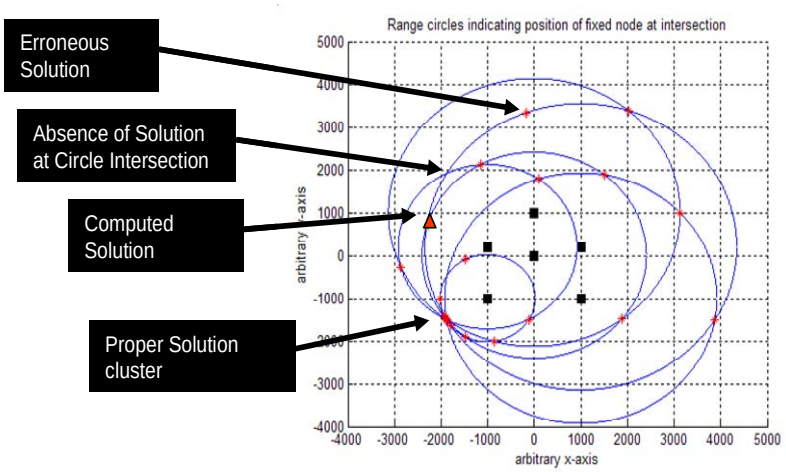


Figure 19. The asterisks represent possible solutions, which should exist at every intersection of range circles. For this case, an erroneous solution has been calculated less than a meter away from a true solution. Due to the close proximity of the solution pair, each of their weight values has been amplified tremendously. In fact, the weights of the solutions have been increased to the point where all other solutions are neglected, and the estimated solution, indicated by a red triangle, is located in at outlying pair.

The erroneous solutions were determined to be the result of improper coordinate transformations. Due to a coding error, the MATLAB script had added an offset angle of $3\pi/2$ to solutions calculated from node pairs lying along the x-axis. In short, solutions calculated from the node pairs 2/4 and 5/6 (see Figure 19) were incorrect.

3. Simulation 2 Results

Correction of this error eliminated the very high error cases and displayed a subsequent improvement in standard deviation. The maximum error case decreased to just over 100 m, and the mean error decreased to 9.52 m with a standard deviation of 12.26 m.

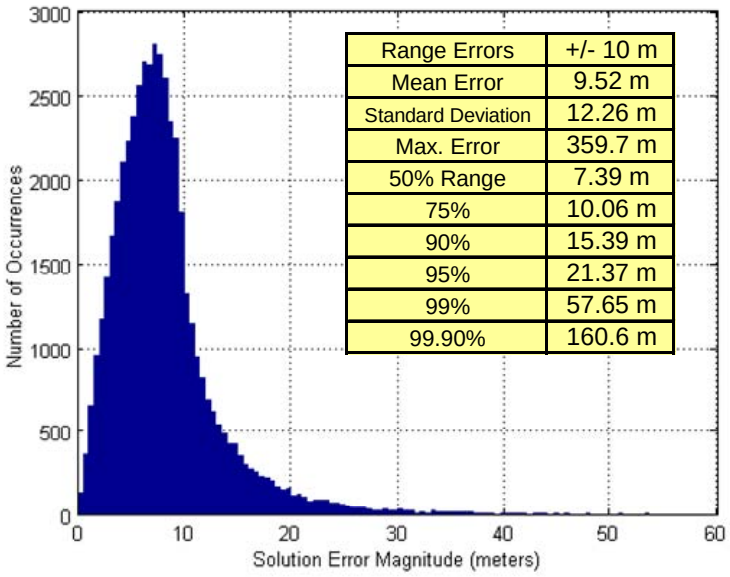


Figure 20. Solution Error Distribution for Simulation 2, which corrected coordinate transformation errors. The error magnitudes for outlying solutions have diminished significantly, and general performance has also improved.

Examination of the high error cases revealed a problem related to shallow angles. For nodes with similar bearing angles to the target, the range circles intersect with a very shallow angle. When coupled with only moderate range errors, these intersections tend to shift the solution for the given node pair. For the maximum error cases, a pair of shifted solutions would often lie in close proximity to one another, amplifying their respective weights and producing in a poor solution.

Shallow angle cases can occur for nearly any target position, but the cases of maximum error occurred predominantly for positions well outside the network perimeter. For such cases, several nodes (or even *all* nodes, for positions at extreme distances from the network center) may exhibit similar bearing angles. A greater number of shallow angle intersections results in a higher probability that many solutions will be shifted along the tangent of the range circles, and a large number of shifted solutions offers an increased likelihood that the estimated position will contain a high error. A replication of the maximum error case for Simulation 2 illustrates this concept, as seen in Figure 21.

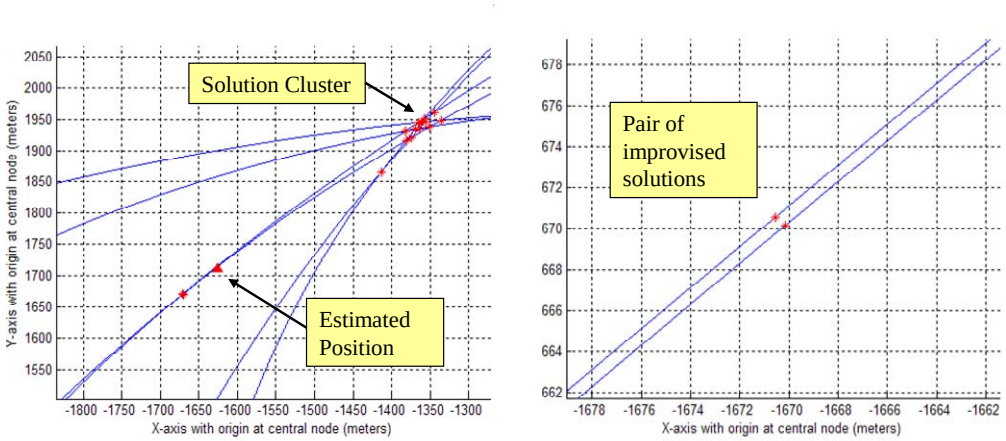


Figure 21. Shallow Angle Error: The estimated target node position (left), indicated by a red triangle, has been shifted from the main cluster of solutions. A closer look (right) reveals that a combination of shallow angle errors and a highly-weighted improvised solution pair (for non-intersecting range circles) has shifted the solution by over 350 m from the true target location. Note the target location, is over 1000 m outside the network perimeter.

C. REFINING THE ALGORITHM

1. Constrained Operating Area

Given the magnitude of error for some of the worst shallow angle scenarios, we conduct further simulations to investigate the more general impact of this phenomenon. We now conduct a simulation for random data within a more constrained operating area. For Simulation 2, the grid of random positions was spread over a 4000 m by 4000 m box centered at the origin. For Simulation 3, we reduce the region sides to 2400 m.

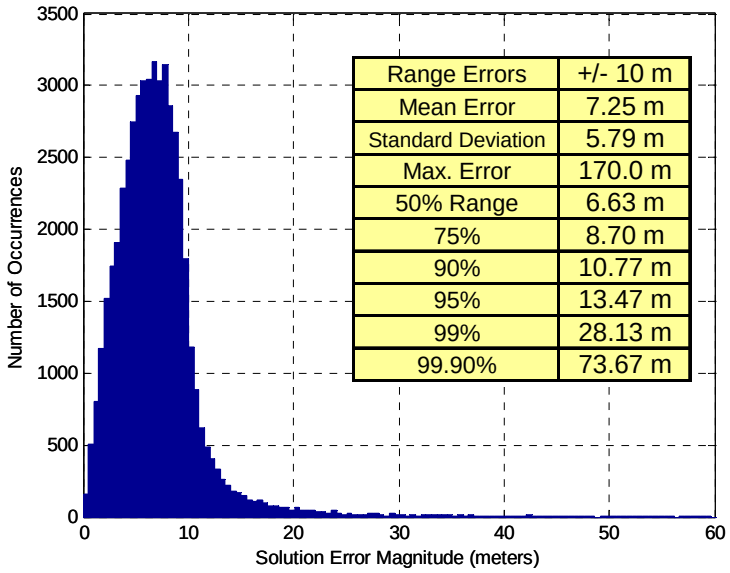


Figure 22. Solution Error Distribution for Simulation 3, which was conducted within a constrained operating area. The improved results suggest that the shallow angle/ improvised solutions problem did, in fact, skew the results of Simulation 2 to a significant degree.

The results of Simulation 3 show a significant improvement over those of Simulation 2, suggesting that the shallow angle effect is significant, especially for localization outside the network perimeter. The maximum error case of 170.0 m is attributed to a case nearly identical to the maximum error case shown in Figure 21. The error magnitude is smaller for this case because the constrained operating area has reduced the size of the range circles (resulting in more well-defined defined arcs and a lower probability of a large solution shift).

2. Modification of the Exponential Term

The root of the shallow angle problem stems from a single solution pair "outweighing" all other pairs. Adjustment of the arbitrary exponential term α in the general weight equation (Equation 9) offers a possible remedy. We are at liberty to adjust α because the weights are dimensionless coefficients. The exponential term tends to amplify weight values for solutions close to one another while disregarding isolated solutions. The magnitude of α determines how severely weight values will be amplified for close solutions. Hence, a smaller α may constrain the range of weight values such that a single pair of close solutions will not suffice to shift the estimated location. As

discussed in Chapter II, we originally used the value α=2 by arbitrary choice. Simulations with smaller α values will indicate the usefulness of modifying the exponential term. We now run simulation for α values of 0.5 and 1 and present the resulting statistics in Table 1.

α	0.5	1	2
Range Errors	+/- 10 m	+/- 10 m	+/- 10 m
Mean Error	10.72 m	6.01 m	7.61 m
Standard Deviation	4.73 m	3.25 m	3.68 m
Max. Error	35.20 m	65.09 m	62.13 m
50% Range	10.55 m	5.75 m	7.53 m
75%	13.87 m	7.69 m	9.48 m
90%	16.85 m	9.44 m	11.36 m
95%	18.79 m	10.75 m	13.43 m
99%	22.58 m	16.45 m	19.53 m
99.90%	27.67m	30.75 m	30.94 m

Table 1. Algorithm Performance for Various Alpha Values

The results indicate that better results are generally obtained using a value of 1 for the exponential term, though shallow angle errors still exist. For all further simulations, we shall use a value of 1 for the exponential term.

3. Shallow Angle Correction Factor and Short Range Treatment

A second possibility for reducing shallow angle error may lie in adjustment of individual solution weights based on bearing angles themselves. For an ideal triangulation scenario, range circles would intersect at close to 90°.

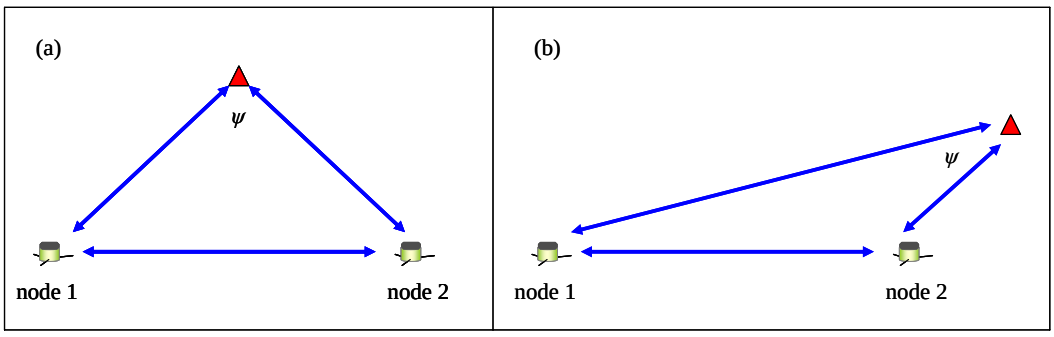


Figure 23. Illustration of the bearing angle difference ψ. We desire a value close to $\pi/2$ to minimize shallow angle error (a). Implementing a correctional factor based on $\sin(\psi)$ will decrease the weight values of shallow angle cases, such as the one shown in (b).

To implement a shallow angle correction factor, we include the following correction factor in the weight equation, where ψ is the bearing angle difference between the 2 nodes of a single pair.

$$W_i = \sum_{j=1, j\neq i}^{m} \left[\sin\psi_i \cdot \sin\psi_j \frac{1}{(x_i - x_j)^2 + (y_i - y_j)^2 + 0.01} \right]^{\alpha} \tag{20}$$

In addition to the correction term, we now direct the algorithm to neglect ranges less than 150 m. We prefer to not use short ranges in field experiments at this point because the *z* (depth) component of UUV-node distances will become significant for such small values, and neglecting *z* will exaggerate the range value. Neglecting short ranges will result in a number of solutions less than the value of *m* that was calculated in (16).

Angle Correction	No	Yes
α	1	1
Range Errors	+/- 10 m	+/- 10 m
Mean Error	6.01 m	6.07 m
Standard Deviation	3.25 m	3.30 m
Max. Error	65.09 m	63.70 m
50% Range	5.75 m	5.82 m
75%	7.69 m	7.76 m
90%	9.44 m	9.46 m
95%	10.75 m	10.82 m
99%	16.45 m	16.64 m
99.90%	30.75 m	32.22 m

Table 2. Algorithm Performance for Angle Correction Term

The simulation results suggest that the shallow angle correction has done little to reduce the effects of those cases. Shallow angle cases remain a troublesome issue, especially for non-intersecting range circles, which tend to heavily shift position estimates. We choose to not include the correction factor in the weight equation for further tests. We maintain the 150 m minimum range requirement, however, because we the accuracy of ranges should decay exponentially below this threshold (see Figure 34).

4. Skewed Error

For all previous simulations, range errors have been limited to within 10 m of the true range. To better simulate true conditions, we offset the data to create simulations with generally short or generally long range data. To do this, we use the same ± 10 m as

before, but we offset the error by -10 and 10 m according to (19). Such simulations yield the following results:

Alpha	1	1	1
Range Errors	+/- 10 m	-20-0 m	0-20 m
Mean Error	6.01 m	13.15 m	12.56
Standard Deviation	3.25 m	6.30 m	5.04 m
Max. Error	65.09 m	108.71 m	96.23 m
50% Range	5.75 m	12.83 m	12.59 m
75%	7.69 m	16.16 m	15.75 m
90%	9.44 m	18.94 m	18.11 m
95%	10.75 m	21.23 m	19.40 m
99%	16.45 m	35.10 m	25.48 m
99.90%	30.75 m	64.80 m	48.02 m

Table 3. Algorithm Performance for Skewed Error.

As indicated by the table above, an offset of the range error will adversely affect algorithm performance. Field experimentation will almost certainly encounter a range measurement bias, though probably not as severe as that used in these simulations.

D. SYNTHETIC TRACK SIMULATIONS

Formatted: Indent: Hanging: 0.75"

Now that we have attained a satisfactory level of algorithm performance, we create three synthetic tracks to run a more practical set of tests. The purpose of these tests is to simulate the types of tracks that may be encountered during field experiments. We employ the same six-node system grid as before, and we run each track simulation 1000 times to acquire accurate data. The tracks include an interior path that circles the central node but stays well inside the network perimeter, an exterior path that circles the outer perimeter, and a more general track that meanders through the network. We run each track simulation both with and without the angle correction term, and we maintain the 150 m minimum range requirement for all simulations as well as an α value of 1 for the exponential term. Simulations are performed with ranging errors between -10 and 10 meters of the true ranges.

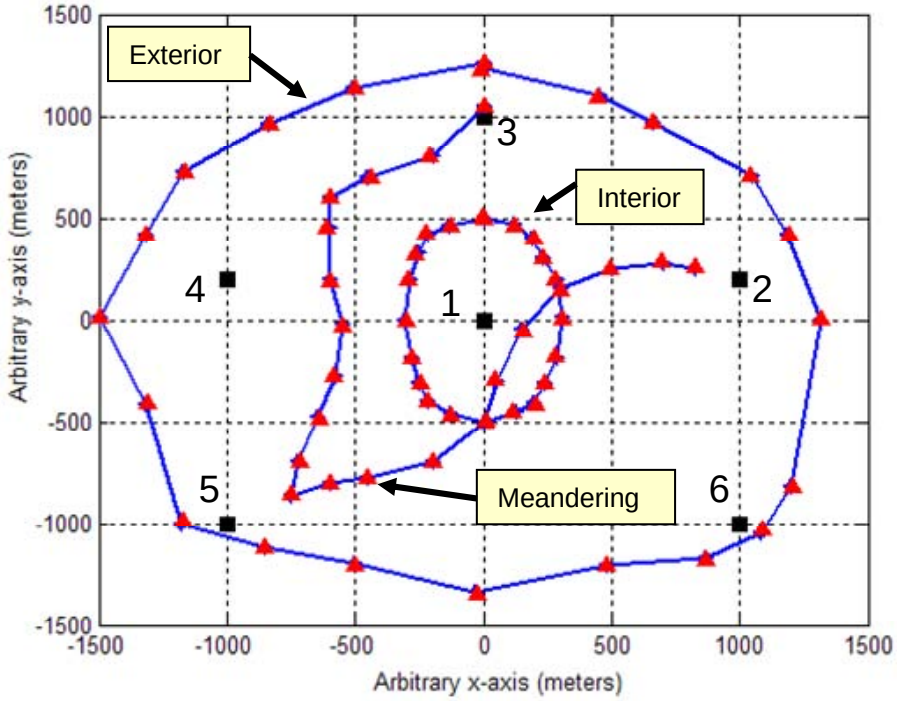


Figure 24. Synthetic Track Simulations: The plot shows the three synthetic tracks used for practical algorithm simulations. The solid blue lines represent the tracks themselves, which are defined using sets of 21 discrete points. The red triangles indicate the mean algorithm solutions for each waypoint, averaged over 1000 simulations, each with independent random error.

The following table lists the results for each track. We calculate the mean error for all 21 waypoints of each track and average that value over 1000 simulations to acquire statistically significant results. The outcome shows that the algorithm produces consistent position estimates and is ready for field testing.

Track Name	Mean Error (m)	Std. Dev. (m)
Interior	6.54	2.82
Exterior	6.53	2.77
Meandering	6.52	2.77

Table 4. Results for Synthetic Track Simulations. Consistent results indicate a reliable positioning algorithm that is ready for field tests.

IV. EXPERIMENT SETUP

Field tests occurred as a major objective of the three-day Seaweb ARIES May 2005 experiment in Monterey Bay. Chapter 1 describes the equipment used. Six repeater nodes served as the sources of range data, and the Seaweb ARIES team operated the system from aboard the *Cypress Sea* workboat.

A. OPERATING AREA

The operating area exhibited a sandy, relatively flat bottom with a gentle depth increase moving from east to west (Figure 25), and the landward buoy was deployed approximately 1 km from the surf zone. The bottom exhibited little variation in the north/south direction.

B. SOUND PROPAGATION CHARACTERISTICS

1. Sound Velocity Profiles and Ray Tracing Models

A series of CTD (conductivity, temperature, depth) casts during the first two days of testing provided sound speed data. The results indicate the presence of a very thin surface layer, below which sound speed steadily increased with depth. Having obtained sound velocity profiles, ray tracing methods then predicted propagation for various source depths. Of particular interest are ray models for sources at the nominal ARIES and Seaweb repeater node depths, assumed to be 5 and 25 meters, respectively.

Of particular importance for range measurements is direct-path propagation between the UUV and repeater nodes. Our assumption of straight line paths is a source of range error for even direct paths. A reflected ray from either the bottom or surface would greatly decrease the accuracy of range measurements under the straight line path assumption. The model for 25 m source depth (Figure 27) shows that a direct path to a receiver at 5 m will be available out to approximately 1000 m at our experiment site.

2. Environmental Conditions and Ambient Noise

The operating area offered favorable environmental conditions. The surf zone, while located nearby (Figure 25), contributed only minimal ambient noise levels. Shipping traffic in the area was non-existent during the tests. Winds presented the most significant noise. Each afternoon, surface conditions rapidly deteriorated as the local winds picked up. This elevated noise levels and made equipment recovery difficult.

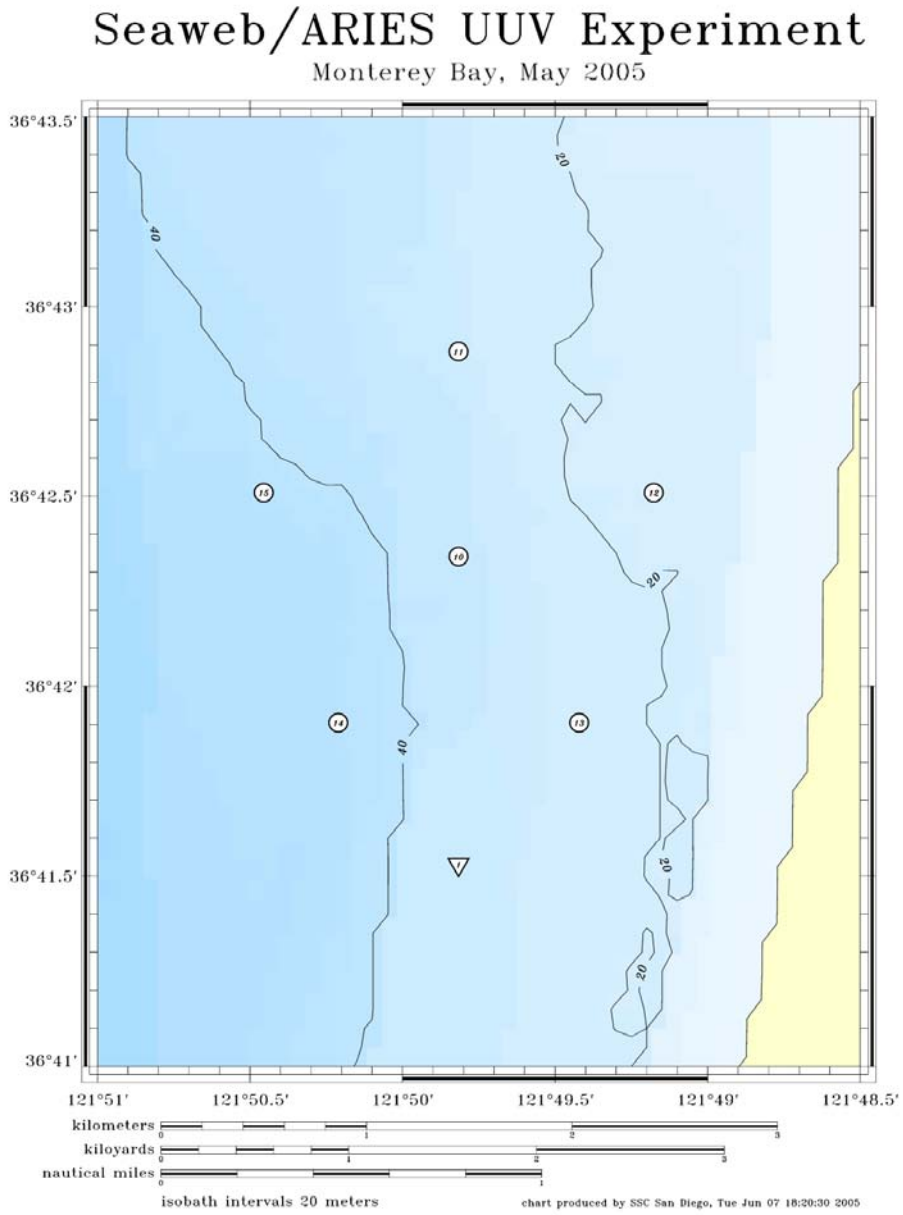


Figure 25. Seaweb/ARIES May 2005 Experiment, Operating Area. The six fixed grid nodes (circles, numbered 10 through 15) were spaced approximately 1000 m apart in a pentagonal geometry, with the sixth node located in the center, approximately equidistant from the others. The RACOM buoy (inverted triangle) was moored just south of the network.

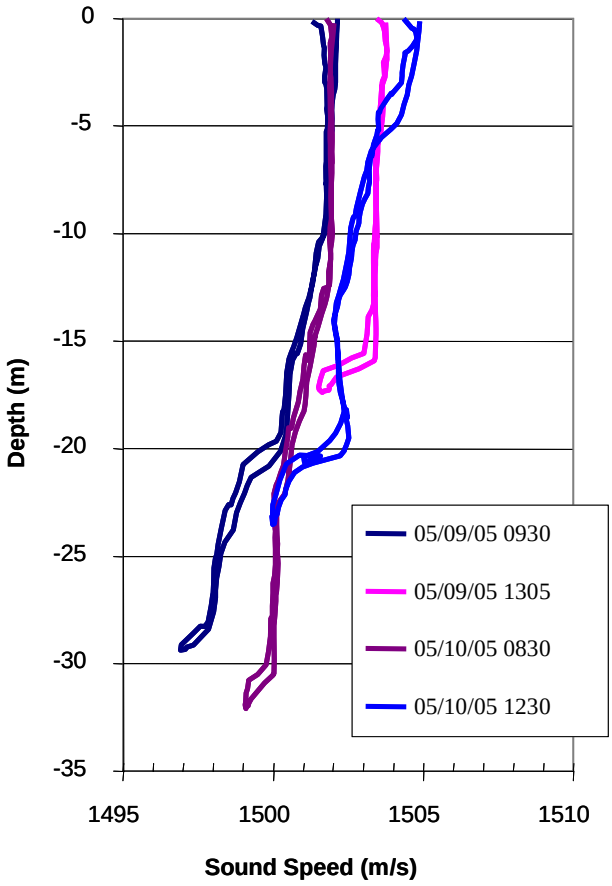


Figure 26. Sound Velocity Profiles. CTD casts provided accurate data within the operating area during the first two days of testing.

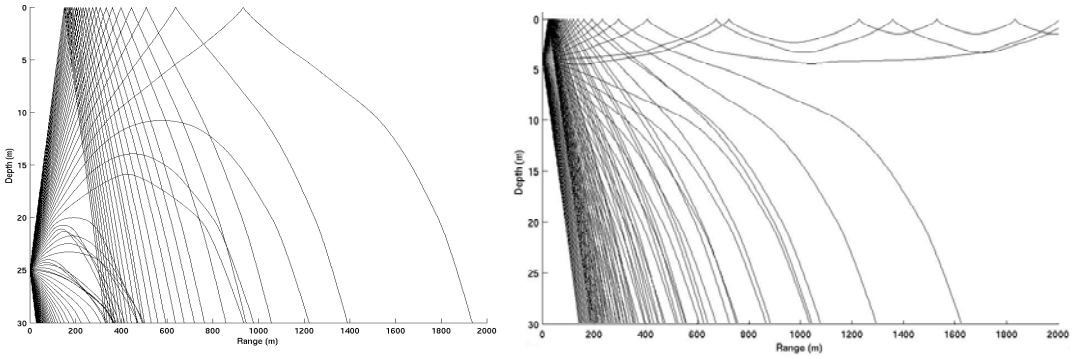


Figure 27. Ray Tracing Models for nominal operating depths of Seaweb repeaters (left) and ARIES UUV (right).

C. EXPERIMENTAL PROCEDURES

For the actual tests, the ARIES UUV performed a series of tracks through the network. To accomplish this, the UUV would surface so that a navigation track would be downloaded by radio command to the UUV by ARIES operators on the workboat. Each navigation track included a set of waypoints. ARIES uses its own inertial navigation system to attempt to follow the track, periodically coming to the surface to obtain a correctional GPS (Global Positioning System) fix. Upon obtaining a correctional fix, ARIES would submerge and continue its track, having corrected its position. The task of the Seaweb team was simple: obtain as much range data as possible and input the data into the positioning algorithm. The goal was to obtain a large number of ranging fixes and compare that information to ARIES' own GPS fixes and inertial navigation data.

V. EXPERIMENTAL RESULTS

A. CALIBRATION FIXES

Data collection began with a series of "calibration fixes" designed to test the accuracy of the algorithm with a nearly stationary target. For these fixes, ARIES was positioned directly astern of the research vessel. During these tests, a series of 3 broadcast pings acquired range data. Since the UUV was positioned next to the research vessel, it was possible to use either a handheld GPS unit or ARIES' own GPS system to compute a simultaneous satellite fix for ground truth. In addition to these three fixes, two more broadcast pings were obtained with simultaneous GPS fixes during testing to test the practical accuracy of the algorithm. Of these 5 fixes, the algorithm achieved a mean error of 7.5 m with a maximum error of 14.1 m.

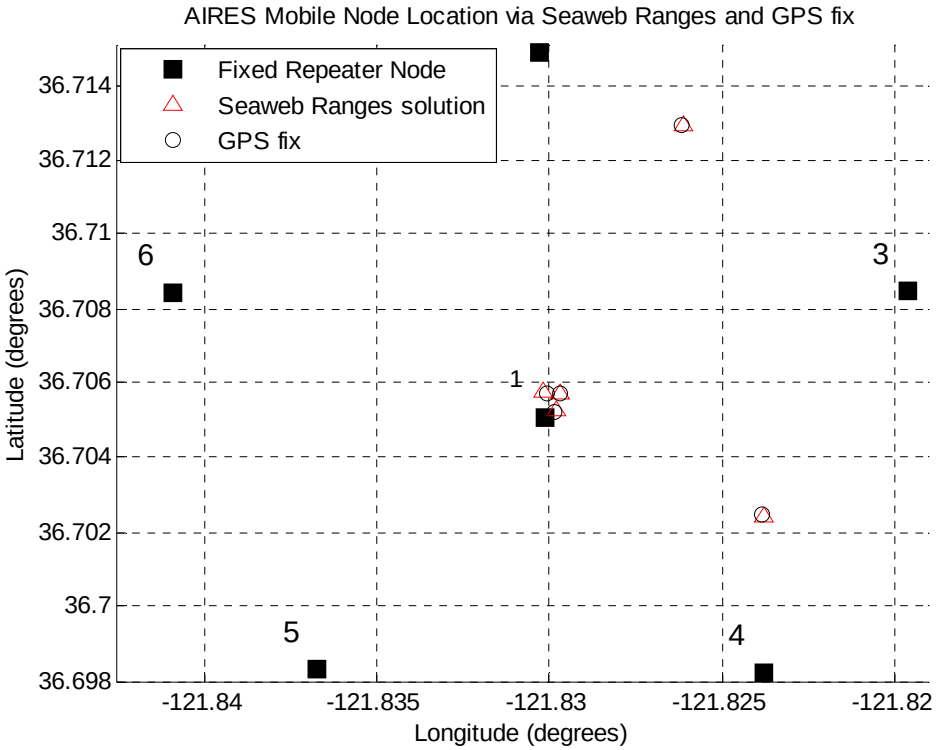
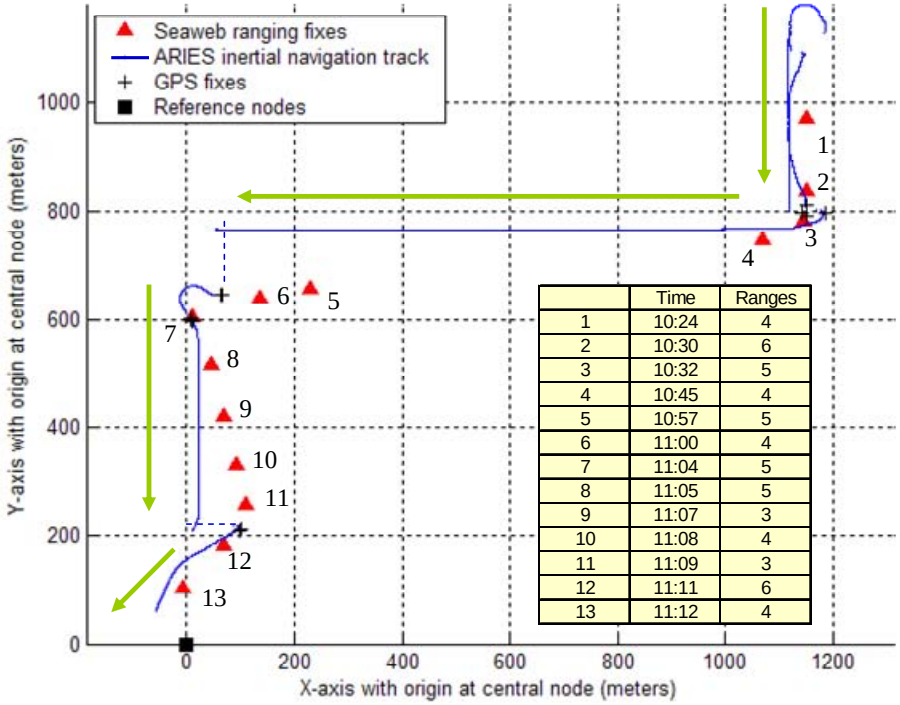


Figure 28. Calibration Fixes. The three fixes next to Node 1 represent those obtained with the UUV held astern of the research vessel. The other two fixes show broadcast pings conducted while ARIES was surfaced, allowing simultaneous acquisition of a GPS satellite fix. Due to their proximity to the central node, the three fixes near Node 1 neglected range data from that reference node.

B. TRACK RUNS

1. Track 1

The following track presents navigation data obtained on day three of the experiment over a period of 48 minutes.

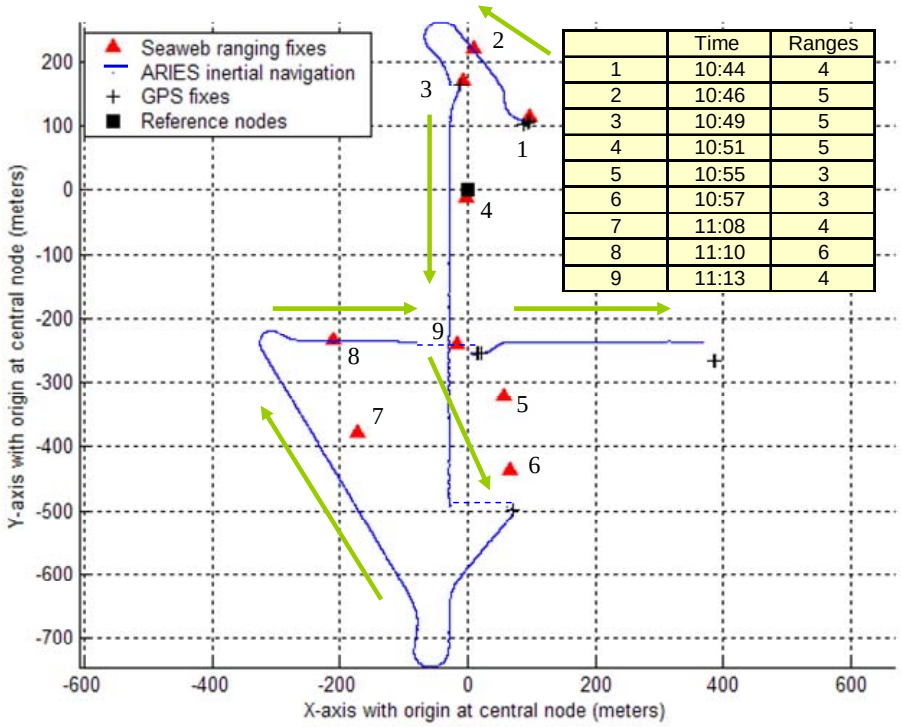


	Time	Ranges
1	10:24	4
2	10:30	6
3	10:32	5
4	10:45	4
5	10:57	5
6	11:00	4
7	11:04	5
8	11:05	5
9	11:07	3
10	11:08	4
11	11:09	3
12	11:11	6
13	11:12	4

Figure 29. Track 1 results. The triangles indicate Seaweb ranging fixes, with numbers corresponding to the timestamps in the table. The “Ranges” column shows the number of ranges available for that particular fix. The solid lines represent ARIES inertial navigation data points, and the plus signs represent GPS fixes. The dotted lines indicate instances when the UUV surfaced and corrected its position with a GPS fix. The GPS fixes near fixes 6 and 11 show large inertial navigation drift. The corresponding Seaweb ranging fixes demonstrate the accuracy of our positioning method, especially compared with inertial navigation.

2. Track 2

The following track, though more complex than the previous one, again demonstrates the algorithm's ability to outperform inertial navigation. This track was conducted on the day two of the experiment over a period of 29 minutes.



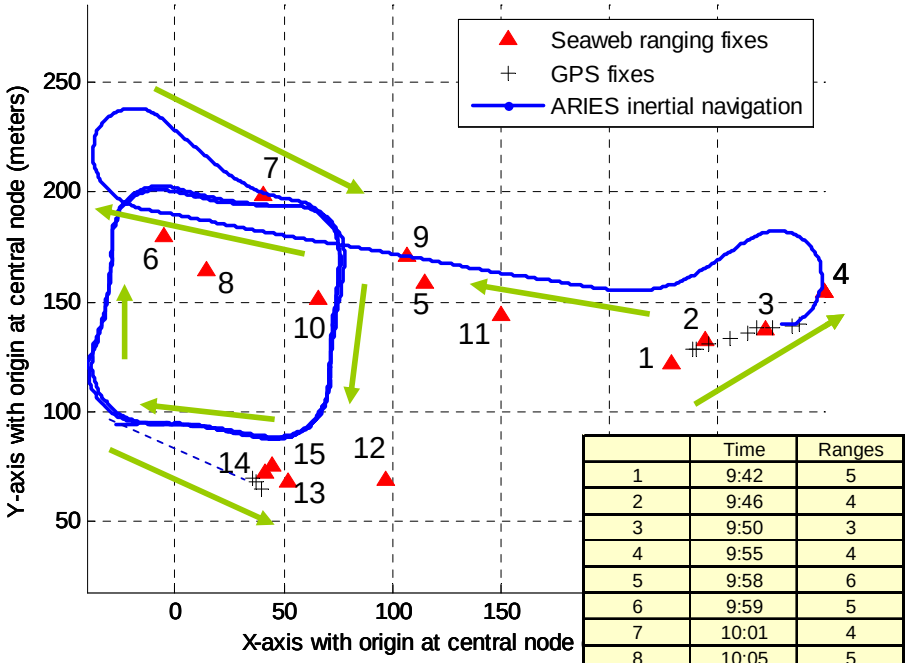
	Time	Ranges
1	10:44	4
2	10:46	5
3	10:49	5
4	10:51	5
5	10:55	3
6	10:57	3
7	11:08	4
8	11:10	6
9	11:13	4

Figure 30. Track 2 Results. The data are denoted as before. Points 4 through 6 demonstrate the same type of drift pattern shown by Track 1. The green arrows indicate the direction of motion of the UUV.

C. ADDITIONAL DATA

Several other tracks were performed during the experiment, but these provided less useful results for a combination of reasons. Most importantly, algorithm performance was limited by the relatively low rate of data arrival. At best, broadcast pings could be conducted 1 minute apart, so the algorithm was unable to keep pace with tracks containing abrupt changes in heading. Track 2 provides a good example of this problem (Figure 24). Between fixes 7 and 8, the inertial navigation attempted to drive the UUV over 100 m past the Seaweb fixes. However, no ranges were collected during that time period, so it is not possible to estimate the true position of the UUV at the time of the

turn. Figure 25 shows how the small update rate prevents the algorithm from achieving good localization performance for the “box” tracks that were used for the initial test runs.



	Time	Ranges
1	9:42	5
2	9:46	4
3	9:50	3
4	9:55	4
5	9:58	6
6	9:59	5
7	10:01	4
8	10:05	5
9	10:07	5
10	10:10	5
11	10:12	4
12	10:14	6
13	10:15	4
14	10:17	5
15	10:19	5

Figure 31. This “box” track illustrates the resolution problem with Seaweb ranging. In this case, the UUV did not surface at all during its run. This meant that GPS fixes were not available for comparative purposes except at the beginning and end of the track. Because the actual “box” portion of the track, for which ARIES completed 2 loops, lacks GPS fixes, so only inertial navigation data is available for verification of the Seaweb fixes. Tracks 1 and 2 (Figures 23, 24) have already proven such data is inaccurate). The result is an incoherent field of fixes spread over the box region. Fix 11 appears to be located almost 80 meters outside of the box perimeter, but we have no way to either verify or refute this position estimate. GPS fixes at the beginning and end of the track correspond well with our position fixes at these times, however.

VI. ANALYSIS

A. ACCURACY OF THE RANGING DATA

To assess the accuracy of the computed range data, we examine the intersection points of the range circles for Seaweb ranging fixes. A tight cluster of points generally indicates accurate ranges, while a cluster spread over a large area suggests ranging errors. We examine the solution clusters for selected fixes to gather insight regarding ranging error. The zoom views of intersection points shown in Figure 32 predict the presence of skewed range errors. The recurring case of non-intersecting range circle pairs suggests that the range data may be skewed toward short values, contradicting the algorithm assumptions. We assume straight-line paths between nodes, but, in reality, sound speed variability induces refraction of the acoustic waves, resulting in longer, curved paths (as traced in Figure 21). Given a uniform depth, a refracted path would travel a shorter distance in a given amount of time than a straight line path, meaning that our straight-line assumption yields exaggerated ranges. We also neglect depth changes in the 2-dimensional model of the operating area. Hence, for UUV-fixed node ranges, we essentially measure the hypotenuse of a triangle as illustrated by Figure 33.

Non-simultaneous ranging data
2-dimensional model of environment
Inaccurate locations of reference nodes
Multi-path sound propagation
Neglection of sound wave refraction
Incorrect sound speed assumption
Quantization error in time measurements

Table 5. Primary sources of range measurement errors. This table lists the main sources of ranging errors in an estimated order of magnitude, since we lack the appropriate data to conduct a more complete error analysis.

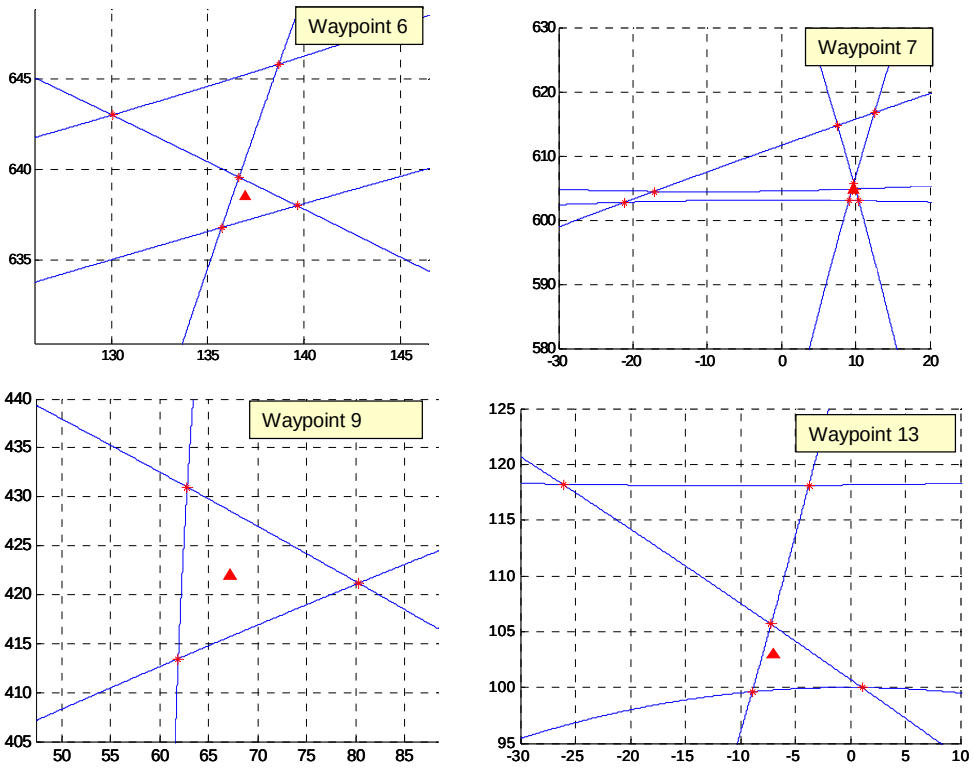


Figure 32. Intersection Points of Fixes for Track 1 (Figure 23) seem to indicate short range measurements. Waypoints 6, 7, and 13 each contain 1 pair of non-intersecting range circles, which can only occur for cases of short range measurements. The likely culprit of this phenomenon is the non-simultaneity of range measurements for a moving target.

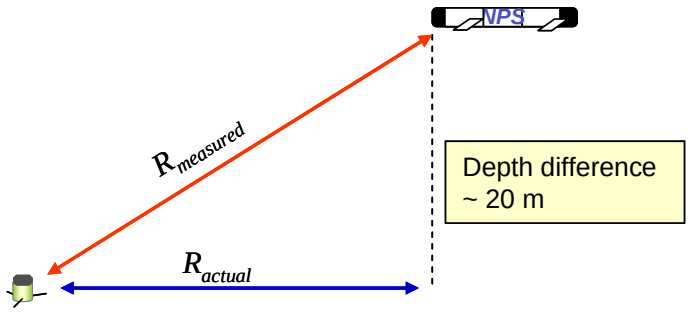


Figure 33. The assumption of a 2-dimensional space should lead to exaggerated range measurements. The 20 m depth change between the nominal UUV and repeater depths indicates that the difference between $R_{measured}$ and R_{actual} is negligible for long ranges. The difference becomes significant at short ranges, however

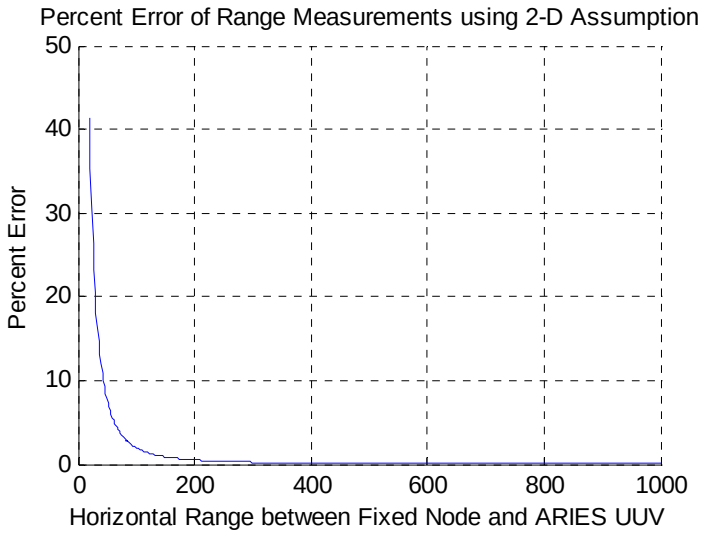


Figure 34. The effects of the 2-dimensional assumption are only manifested at short ranges. A 5% error occurs at about 60 m for the geometry with a 20 m depth change between the fixed grid and mobile node.

The most obvious algorithm assumptions predict long range estimates, yet the data from Track 1 displays the opposite result. A third assumption may hold the answer. As discussed in Chapter I, the broadcast ping does not actually yield simultaneous range measurements, but, rather, a collection of range measurements spread over a time period of up to one minute. A UUV moving at 1.2 m/s will travel 72 m over that time period. If two range measurements were taken near the beginning and end of the time spread, the UUV position may have changed such that the ranges appear to be short, even if they were actually long! The 1500 m/s assumption for sound speed may have also lead to scaling of the range measurements.

Uncertainties in the fixed node positions may account for additional skewed range data. Repeater positions were obtained by reading the output of a handheld GPS receiver at each drop point. Positioning errors could easily result from error within the GPS unit itself, drift of the repeater during its descent to the sea floor, and subsequent “dragging” of the apparatus along the sea floor by ocean currents. No data were taken to record currents or drift patterns, so it is difficult to estimate errors in the repeater positions, though each of the six nodes was recovered at the same Latitude/Longitude coordinates as deployed (within instrument quantization error). This suggests a fairly reliable repeater

deployment technique, though the long-term goal of aircraft deployment of the network will require more sophisticated procedures.

A final comment should be made regarding the assumption of direct path propagation. A large percentage of the range measurements exceeded the 1000 m limit of direct path availability. The range measurements made beyond this limit are suspect, as the travel times may represent surface-reflected rays.

Apart from the issue of skewed data, we examine the results of fixes near the network perimeter to look for shallow angle error. Figure 35 shows three waypoints with multiple shallow angle intersections. None of three cases has produced an extremely skewed solution like those seen in some of the simulations. The experimental tracks were all conducted either within the network perimeter or just outside of it, so the kinds of scenarios that would result in large numbers of shallow angles simply did not occur. The plots shown in Figure 28 have all produced one or more outlying solutions at large distances from the main cluster, but none of those solutions has significantly impacted position estimates.

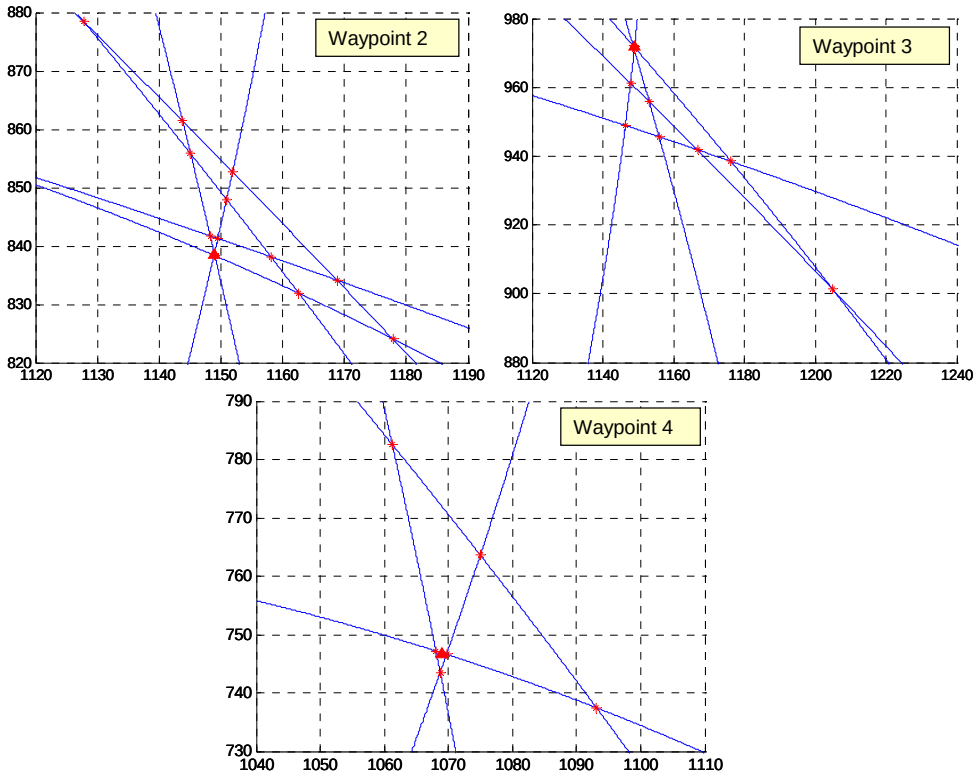


Figure 35. Shallow Angle Intersections for Track 1 fixes. Close-up views for Track 1 waypoints near the network perimeter exhibit shallow angle errors, as predicted by the simulation results (Chapter III). For the cases shown here, the weighting method largely ignored the outlying solutions.

B. MOTION OF THE ARIES UUV

The algorithm produced good results for Tracks 1 and 2. The nature of the GPS data set combined with a lack of clock synchronization between the Seaweb and ARIES systems precludes the comparison of fixes on a common time scale, but the drift phenomena suggest favorable performance by the positioning algorithm. Drift motion of the UUV is shown most readily for track portions where ARIES attempted to move in a straight line for 500 m or more. Figures 23 and 24 show that the UUV typically experiences a drift during such track legs. As shown in the figures, at the end of each straight line path, ARIES would surface and acquire a GPS fix between 50-150 m to either side of the planned track. The sequence of ranging fixes accounts for such drift in all three cases, twice in Track 1 and once in Track 2.

Despite its ability to account for drift, the algorithm failed to provide adequate sampling for generating a complete navigation track for all test runs, with the exception of Track 1. The three tracks shown in Figures 23-25 demonstrate a mean time interval of 3.35 minutes between fixes. This long interval simply does not allow for the reconstruction of rapidly changing tracks such as those given by Figures 24 and 25. It is evident that the ranging measurements used in combination with inertial navigation would substantially improve the undersea navigation solution.

VII. CONCLUSIONS AND FURTHER WORK

A. METHOD FEASIBILITY

The Seaweb/ARIES Experiment proved the feasibility of undersea navigation through the use of an undersea acoustic communications network. Accurate results were achieved through rigorous simulations, and similar performance was attained by a set of calibration fixes during the experiment. Despite a large error budget, good results were obtained for the experimental tracks. Improvements in resolution will be needed if a stand-alone navigation system is desired, but the positioning algorithm has proven its ability to localize the UUV within the network. Further work may be pursued in several areas, which we discuss in the next section.

B. FURTHER WORK

1. System Integration

Efficient system integration of the algorithm would involve implementation within the UUV navigation system. If the ranging data were fed into a computer onboard the ARIES vehicle, correctional ranging fixes could be obtained while submerged, similar to the present method of surfacing to obtain a GPS fix. This requires an autonomous program that will automatically upload range data from broadcast pings. The present algorithm merely uses a MATLAB script (Appendix) to import data from an Excel file. Further system integration will entail more sophisticated computer programming techniques.

2. Three-Dimensional Model

Modeling the undersea environment in three-dimensional space should greatly improve the accuracy of range data. At present, travel times are used to directly calculate straight line distances using a constant sound velocity of 1500 m/s. On a simple level, the three-dimensional model might involve retaining the straight-line path assumption and simply accounting for the depth change between the UUV and fixed reference nodes. More advanced models would incorporate equations to model the refraction of sound waves for calculating inter-node ranges [2, 4, 5].

3. Sensitivity to UUV Motion

The Seaweb ARIES Experiment employed a slow-moving vehicle (1.2 m/s), yet the effect of the motion was evident due to the non-simultaneity of range measurements within a single broadcast ping. Errors from this source would increase significantly if the algorithm were implemented for tracking a submarine moving at speeds of 10 knots or more. A solution to this problem involves the combination of ranging data with UUV navigation data. Even a very rough dead-reckoning track or heading angle would allow a revised algorithm to incorporate both data sources and produce a better position estimate. This method would prove especially useful given the availability of the individual node dwell times (Chapter 1).

4. Fixed Grid Self Localization

Self localization of the fixed grid remains to be implemented. A long-term goal of Seaweb is to produce a rapidly deployable ad hoc network that can be dropped from an aircraft and quickly initialize undersea communications. Successful routing of the network requires knowledge of fixed node locations, which are presently determined by a handheld GPS unit as the nodes are deployed manually from the deck of a research vessel. Self-localization may be accomplished using a method similar to the positioning algorithm, though self-localization of a fixed grid will not require near-simultaneous measurements because the repeaters will not move a great distance from their initial locations [3, 4]. Range data from the Seaweb ARIES Experiment may be used to perform localization of the fixed grid. Such analysis would lend additional insight into the accuracy of the present deployment techniques.

5. Other Positioning Methods

The weighting method used for this thesis worked quite well, but other methods do exist, and several of these methods have been studied already. Articles by Stan E. Dosso and Kenneth D. Frampton outline a few of these methods [3-5], which could be adapted to the Seaweb navigation problem.

Dosso's method performs localization of a hydrophone array using a set of linearized "Time Difference of Arrival" (TDOA) equations. This is accomplished through an inversion technique described in his journal articles [3, 4]. Frampton also employs TDOA signals [5] for multiple acoustic sources at known locations in or around

an undersea network. Each source transmits a chirp, which is received by the sensor (repeater) nodes. TDOA data is then collected and analyzed at a central location, and the resulting set of nonlinear equations is solved using a least squares technique [5]. A least squares method could be readily adapted to our case using broadcast ping data. Furthermore, Frampton includes depth coordinates in his method outline, so a three-dimensional model has already been provided.

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

1. Rice, J., Seaweb Acoustic Communication and Navigation Networks, In *Proc. Conf. Underwater Acoustic Measurements: Technologies & Results*, Heraklion, Greece (2005).
2. Rice, J., *A Prototype Array-Element Localization Sonobuoy*, Naval Ocean Systems Center, Technical Report 1365, (1990).
3. Dosso, S., et al., Array element localization for horizontal arrays via Occam's inversion, *J. Acoust. Soc. Am.*, Vol. 104.2, (1998).
4. Dosso, S. and B. Sotirin, Optimal Array Element Localization, *J. Acoust. Soc. Am.*, Vol. 106.6, (1999).
5. Williams, Stephen M., Peter L. Schmidt, and Kenneth D. Frampton, Distributed Source Localization in a Wireless Sensor Network, *J. Acoust. Soc. Am.*, Vol. 117.4, (2005) .
6. Marr, W., *Acoustic Based Tactical Control of Underwater Vehicles*, Naval Postgraduate PhD Dissertation (2003).
7. Nguyen, T., ARIES Navigation System Accuracy and Track Following, Naval Postgraduate MS Thesis (2002).

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX

A. ANNOTATED MATLAB CODE

The follow is a portion of the script file used to implement the positioning algorithm in field tests. The fixed node positions were specified by Latitude/Longitude coordinates, which we have convert to *x*/*y* positions within a reference frame with its origin located at the central node.

```
grid=xlsread('fixed_node_positions');     %import Excel data files
range_data = xlsread('range_data');
N=size(range_data,1);

range_set=zeros(size(grid,1),size(grid,1));
for(i=1:1:size(grid,1)) %define inter-node ranges for for fixed nodes
      for(j=i+1:1:6)
            if(i~=j)
                  x_dist=grid(j,1)-grid(i,1);
                  y_dist=grid(j,2)-grid(i,2);
                  dist=sqrt(x_dist^2+y_dist^2);
                  range_set(i,j)=dist;
                  range_set(j,i)=dist;
            end
      end
end

%now we begin the primary loop, which estimates the UUV position
%for each time-stamped set of ranges
for(t=1:1:N)
      clear pairs phi x y psi W WF x_fin y_fin

      ranges=range_data(t,:); %define single line of data as range data

      %now we determine which node pairs will yield a non-imag. soln.
      intersections=zeros(size(range_set,1),size(range_set,2));
      for(i=1:1:size(intersections,1))
            for(j=1:1:size(intersections,1))
                  if(i~=j)
                        R_1=ranges(a,i);
                        R_2=ranges(a,j);
                  R_3=range_set(i,j);
                  theta_a=cosines_law(R_1,R_3,R_2);
                  if((imag(theta_a)==0) & (theta_a~=0)) %two distinct
            intersections
                      intersections(i,j)=2;
                      intersections(j,i)=intersections(i,j);
                  elseif((imag(theta_a)==0) & (theta_a~=0)) %one
      "double" intersection
                      intersections(i,j)=1;
                      intersections(j,i)=intersections(i,j);
                  elseif(imag(theta_a)~=0)
                      intersections(i,j)=0;
                      intersections(j,i)=intersections(i,j);
```

```
            end
        end
    end
end

%now we determine the xy coordinates of the two possible
%solutions for each pair
%that yields a real solution for theta
c=1;
for(i=1:1:size(intersections,1))
    for(j=i:1:size(intersections,1))
        if((i~=j)& ranges(a,i)>=150 & ranges(a,j)>=150)
            org=[grid(i,1) grid(i,2)];
            xax=[grid(j,1) grid(j,2)];
            L_o_x=range_set(i,j);
            delta_x=xax(1,1)-org(1,1);
            delta_y=xax(1,2)-org(1,2);
            if(delta_x==0) %case of vertical tangent
                if(org(1,2)>xax(1,2))
                    phi=-pi/2;
                else
                    phi=pi/2;
                end
            else
                phi_prime=atan(abs(delta_y)/abs(delta_x));
                if((org(1,1)<=xax(1,1))&(org(1,2)<=xax(1,2)))
                    phi=phi_prime;
                elseif((org(1,1)>xax(1,1))&(org(1,2)<=xax(1,2)))
                    phi=pi-phi_prime; %2nd quadrant
                elseif((org(1,1)>=xax(1,1))&(org(1,2)>xax(1,2)))
                    phi=phi_prime+pi; %3rd quadrant
                elseif((org(1,1)<xax(1,1))&(org(1,2)>xax(1,2)))
                    phi=2*pi-phi_prime; %4th quadrant
                end
            end
            if(intersections(i,j)==2) | (intersections(i,j)==1)
                  %find abs value of angle of solution offset
                  %from new xaxis
                  %this is easily determined using the law of
                  %cosines,
                  %where "opp side" (see function code) is range:
                  %xax to mobile node
                theta=cosines_law(ranges(a,i),L_o_x,ranges(a,j));

                %coordinates, angle equal to gamma (a or b)
                gamma_a=phi+theta;
                gamma_b=phi-theta;
                x(c,1)=org(1,1)+ranges(a,i)*cos(gamma_a);
                x(c+1,1)=org(1,1)+ranges(a,i)*cos(gamma_b);
                y(c,1)=org(1,2)+ranges(a,i)*sin(gamma_a);
                y(c+1,1)=org(1,2)+ranges(a,i)*sin(gamma_b);
            else  %(zero intersections case)
                  %now we determine which of three possible cases
                  %for non-intersecting
                %circles has occured (Chapter II, section E.2)

if((ranges(a,i)<range_set(i,j))&(ranges(a,j)<range_set(i,j)))
```

```
                        x(c,1)=grid(i,1)+ranges(a,i)*cos(phi);
                        y(c,1)=grid(i,2)+ranges(a,i)*sin(phi);
                        x(c+1,1)=grid(j,1)-ranges(a,j)*cos(phi);
                        y(c+1,1)=grid(j,2)-ranges(a,j)*sin(phi);

    elseif((ranges(a,i)>=ranges(a,j))&(ranges(a,i)>=range_set(i,j)))
                        x(c,1)=grid(i,1)+ranges(a,i)*cos(phi);
                        y(c,1)=grid(i,2)+ranges(a,i)*sin(phi);
                        x(c+1,1)=grid(j,1)+ranges(a,j)*cos(phi);
                        y(c+1,1)=grid(j,2)+ranges(a,j)*sin(phi);
                elseif((ranges(a,j)>=ranges(a,i))&
    (ranges(a,j)>=range_set(i,j)))
                        x(c,1)=grid(i,1)-ranges(a,i)*cos(phi);
                        y(c,1)=grid(i,2)-ranges(a,i)*sin(phi);
                        x(c+1,1)=grid(j,1)-ranges(a,j)*cos(phi);
                        y(c+1,1)=grid(j,2)-ranges(a,j)*sin(phi);
                end
            end

    %Calculate Weight values
    W=zeros(size(x,1),1);
    for(i=1:1:size(x,1))
        for(j=1:1:size(x,1))
            if(i~=j)
                alpha=1;          %"optimal" value
                x_diff=x(i)-x(j);
                y_diff=y(i)-y(j);
                Wadj=WF(i)*WF(j); %shallow angle correction
                W(i)=Wadj*(W(i)+(x_diff^2+y_diff^2)^-alpha);
            end
        end
    end

    W_sum=sum(W);     %the total weight is the sum of indiv. Wts.

    for(i=1:1:size(W,1))
        x_fin(i)=((W(i)/W_sum)*x(i));       %"weighted average"
        y_fin(i)=((W(i)/W_sum)*y(i));
    end

    soln(a,:)=[sum(x_fin) sum(y_fin)];        %estimated soln.

end   %close the primary loop
```

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1 Defense Technical Information Center
Ft. Belvoir, Virginia

2 Dudley Knox Library
Naval Postgraduate School
Monterey, California

3 Allen Moshfegh
Defense Advanced Research Projects Agency
Arlington, VA

4 Tom Swean
Office of Naval Research
Arlington, VA

5 Tom Curtin
Office of Naval Research
Arlington, VA

6 Doug Ray
Naval Undersea Warfare Center, Keyport
Keyport, WA

7 Paul Baxley
Space and Naval Warfare Systems Center, San Diego
San Diego, CA

8 Chris Fletcher
Space and Naval Warfare Systems Center, San Diego
San Diego, CA

9 Bob Creber
Space and Naval Warfare Systems Center, San Diego
San Diego, CA

10 Bill Marn
Space and Naval Warfare Systems Center, San Diego
San Diego, CA

11 Doug Horner
Naval Postgraduate School
Monterey, CA

12 Tony Healey
Naval Postgraduate School
Monterey, CA

13 Sean Ouimet
Naval Postgraduate School
Monterey, CA

14 Michael Reed
Naval Postgraduate School
Monterey, CA

15 Clyde Scandrett
Naval Postgraduate School
Monterey, CA

16 David Book
Naval Postgraduate School
Monterey, CA

17 CDR Melissa Smoot, USN
Naval Sea Systems Command, PMS NSW
Washington, D.C.