AN ASSESSMENT OF AUV NAVIGATION PERFORMANCE USING LONG BASELINE ACOUSTIC MODELING AND SIMULATION

Feijun Song, Anthony LaVigne, Edgar An, Stewart Glegg
Ocean Engineering Department, Florida Atlantic University
101 North Beach Road, Dania, FL 33004
fsong@oe.fau.edu, lavigne@oe.fau.edu, ean@oe.fau.edu, glegg@oe.fau.edu

Abstract

Navigation software is a critical part of an Autonomous Underwater Vehicle (AUV) system. Due to its complex nature, testing and debugging of navigation software often involve long and costly sea trials. Modeling and simulation is a cost effective way to minimize unnecessary operational overhead. Navigation system can be tested with different simulated sensor signals so that even very well hidden bugs can be possibly simulated and debugged. This paper describes our preliminary modeling and simulation work on the AUV navigation performance using a long baseline acoustic sonar system. The goals of this work are twofold: to develop a mechanism with which to test vehicle position software against a benchmark analytical solution, and to assess the LBL navigation performance as a function of operational conditions. Specifically, the Method of Gaussian Ray Tracing is used to model the propagation of the underwater acoustic signal, the result is then fed into a Hardware-In-the-Loop simulation system that communicates directly with the AUV embedded navigation software. This way the performance of AUV navigation using LBL positioning can be characterized.

Introduction

Autonomous Underwater Vehicles (AUVs) are un-tethered, unmanned robotic platforms that can be pre-programmed to achieve accurate navigation, control and guidance tasks. This enabling technology specifically addresses the fundamental issue of spatial aliasing in conventional oceanographic surveys using point observations, and provides a customizable solution in terms of strategizing spatio-temporal sampling. As AUVs typically operate underwater, the effects of waves and currents on the vehicle motion are greatly reduced, thus providing the required stability for a suite of sensors onboard, such as imaging sonars. Further mission enhancement could involve multiple, small AUVs deployment with long-endurance, adaptive, co-operative capabilities such that the sampling objective can be achieved by means of "divide and conquer" [7][14].

Typical AUV missions include four-dimensional oceanographic data and rapid environmental assessment in which littoral variability can be timely quantified. This added utility value to the underwater community provides an unprecedented opportunity to validate existing ocean models for better forecasting capabilities. Recent results from the Navy funded research on the use of small AUVs for shallow water mine reconnaissance have shown that AUVs have significant potential to augment or replace humans and mammals for some Very Shall Water Mine Reconnaissance tasks. Other notable examples of demonstrated AUV capabilities include wide area side scan survey, targeted multi-sensor survey, and site inspection [1][2].

Unlike terrestrial and space environments, the underwater domain typically imposes the most restrictive physical control and sensor limitations upon any AUV (e.g. corrosion, ground fault, acoustic variability). Although there now exist many remotely operated submersibles (ROV) and a matured field of autonomous robot research, only few AUVs exist and their capabilities are still limited albeit extending. Difficult, tedious and potentially hazardous at-sea testing, technical challenges to directly observe the AUV *in-situ*, environment restrictions, unacceptable loss of an AUV due to tremendous investment in time and resource, are all the reasons for justifying the use of modeling and simulation for AUV development, testing and personnel training so that component and system level troubleshooting and mission planning can be carried out more cost-effectively [3].

Models can generally be considered as mathematical descriptions of the corresponding physical systems of interest whereas simulation addresses how these models are to be implemented in terms of hardware and software configuration. Modeling and simulation are often implemented at vehicle dynamics and control level such that the main objective is to accurately replicate the actual vehicle response subject to a variety of anticipated disturbances, such as waves and currents. This might deem suitable when the focus is placed on theoretical analysis of vehicle dynamics and motion controller design. It should be emphasized that an AUV is a complex system in which the interaction among low-level motion control, navigation, guidance and emergency/event handling tasks determine if a mission can be successfully carried out. When the focus is placed on the systems' reliability, the coupling effect that arises from these tasks cannot be ignored. A general rule of thumb for performing modeling and simulation is to ensure that the behavior of a simulated system closely matches that of the corresponding physical system. To take this further, part or all of system hardware and software can be directly connected to the simulation, (Hardware-In-the-Loop, or HIL).

In an HIL simulation, sensor modeling is a critical part. Recently, we were able to simulate the propagation of acoustic signals in shallow water using Gaussian Ray tracing method. This can be useful for underwater vehicle Long Baseline (LBL) navigation system assessment.

A Long Baseline (LBL) system is a type of underwater acoustic positioning system that consists of a transponder and several beacons. The transponder is attached to the vehicle, whereas the beacons are located at pre-known fixed positions. The transponder sends out acoustic signals to each beacon and waits for its reply. Based on the reply time, the distance between the vehicle and each of the beacons can be determined. These distances can be used to estimate the position of the vehicle.

The first part of this paper presents the Method of Ray Tracing that simulates the signals from a beacon over a flat bottom in shallow water. The algorithm traces the acoustic rays as they bounce between sea bottom and sea surface. The echoes are added together to simulate what the vehicle transponder would hear. The reply time from a beacon to the vehicle can thus be determined.

The second part of this paper describes the HIL simulation, as well as how the reply time obtained from Ray Tracing can be used in the simulation to assess the LBL navigation software performance. The paper concludes with results and discussions.

The Acoustic Models

Two Acoustic Models are used to simulate underwater acoustic signal propagation, i.e., Gaussian Ray tracing and the Method of Images. The former can be applied to a variety of 3-dimensional environments but is computationally intensive. The Method of Images is fast and more accurate but with limited applications. Both can account for transient point sources located in near and far field. In some simple cases, the Method of Images can be used to verify the Gaussian Ray Tracing (GRT) Method.

The GRT method models a source by emitting a finite number of rays that flux out of a point. Each of the ray paths is traced and its angular distance is used with the following Gaussian weighting function [4]

$$W(q) = C * \exp(-a * q^2)$$

Where the constants, C and a, can smooth out the solutions in order to more closely resemble reality [10]. $W(q)$ is a function of the angular distance, q, which is defined as the angle between any given ray and its eigen-ray. An eigen-ray is defined as any ray that passes a given receiver point exactly. Figure 1 shows an eigen-ray (denoted by "o" at center) and a distribution of discrete rays (denoted by "x") with their relative weighting values.

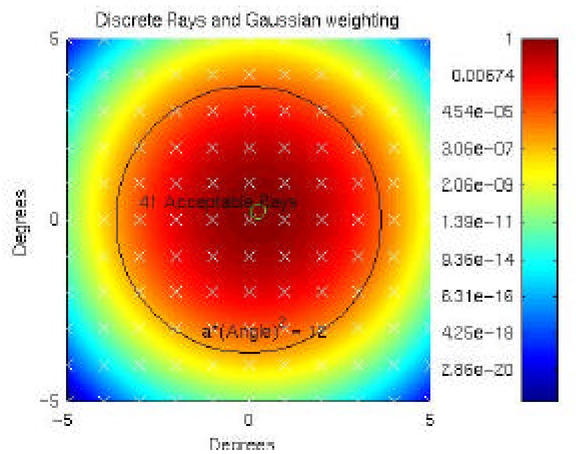


Figure 1: Eigen rays.

The sum of the rays weighted around the eigen-ray is unity. It should be noted that any ray that is outside the circle has a weight value less than $6\text{x}10^{-6}$. The GRT algorithm thus only adds all the rays within the circle of reach, and models each of the included rays as a simple acoustic source with its associated weighting.

The sound pressure from an individual ray is calculated as [4]

$$p_i = CB_i \exp(-aq_i^2 + jwt) / S_i$$

Here w is the angular frequency, t represents time, S_i is the length of the ray, and B_i is a constant.

Enveloping the pressure for a single ray with the time window $H(t+S_i/c_w)$, and summing all the rays together, we have:

$$P = \sum_{i=1}^{N} H(t + S_i / c_w) CB_i \exp(-aq_i^2 + jwt) / S_i$$

Where N is the total number of the rays within the circle, and c_w is the sound speed.

Converting the pressure into voltage by multiplying the above equation with the constants Q and R to, we get:

$$V = \sum_{i=1}^{N} QRH(t + S_i / c_w) CB_i \exp(-aq_i^2 + jwt) / S_i$$

Where Q is the source strength, R is the receiver sensitivity.

Taking the real part of the above equation, we get:

$$V = \sum_{i=1}^{N} B_i QRW(q_i) H(t + S_i / c_w) \cos(w(t + S_i / c_w)) / S_i$$

Note that in the above equations, the time for the ray to travel from source to the receiver point is already considered as the time changes from t to $t + S_i / c_w$.

The Method of Images treats each eigen-ray as if it were from an image source either above the surface or below the bottom. Four image sources are shown in Figure 2.

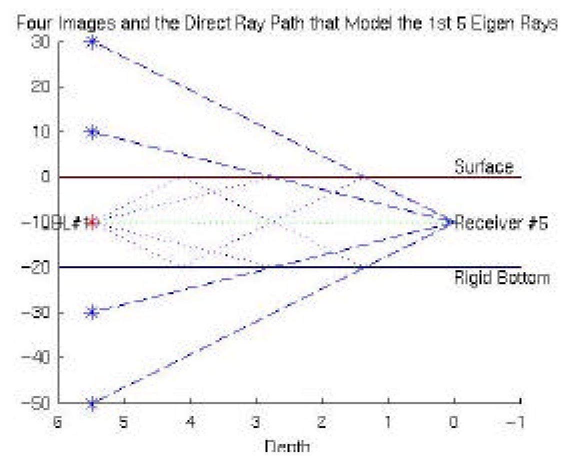


Figure 2: Image sources

The actual paths of the eigen-rays are the dotted lines with the dashed lines being the rays from the image sources. What an acoustic sensor would sense is simply the summation of all the rays from all the image sources. The voltage signal from the Method of Images as a function of time and space can be exactly described as:

$$V = \sum_{i=1}^{N} B_i QH(t + S_i / c_w) \cos(w(t + S_i / c_w)) / S_i$$

Where subscript *i* goes from 1 to *N*, and *N* is the total number of eigen-rays taken into account [4].

Acoustic Simulation Scenario

The scenario for the simulation considered in this paper consists of a flat bottom with a constant bathymetry of 20 meters and sound speed of 1535 meter/second. The receivers and sources were located in mid water column within a grid of 100 x 100 meters, and the emitted signal was normalized and had a frequency of 12kHz with a three cycle Hanning ramped 0.01-second window. It is also assumed that the principle of reciprocity holds. *C* and *a* were set to 0.2841 and 2930 respectively during the whole process of our simulation [13].

The path in which a ray passes a receiver directly is called direct path, or indirect when it passes the receiver after a few bounces over the sea surface and bottom. As illustrated in Figure 3, there is one direct path, and 4 indirect paths (rays have either one reflection and two reflections).

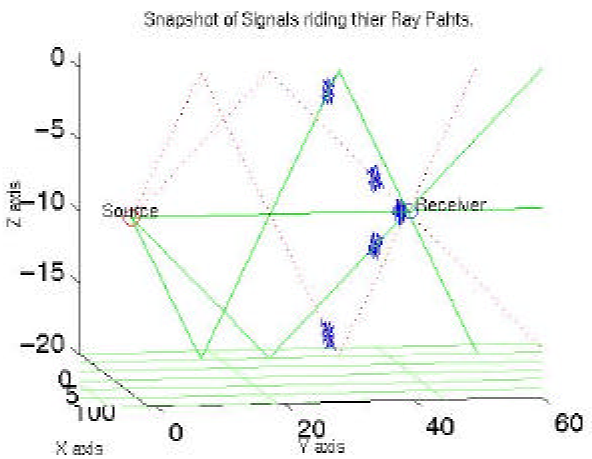


Figure 3: A direct path and four indirect paths

From the figure, one can see that the direct path arrives at the receiver first since it's the shortest path. The more reflections the longer the path thus the longer it takes to reach the receiver, and the weaker the signal is. With a perfect flat bottom and surface, and the receiver and source are at mid water depth, the geometry of each pair of reflecting rays is symmetrical about the direct path. Therefore, the signals of each pair of rays arrive at the receiver at the same time. Reflections off the pressure releasing sea surface cause a 180-degree phase change while reflections off the flat bottom have no phase shift. Consequently, any reflected pair of rays with an odd number of bounces cancels each other out at the receiver point, while any pair of rays with an even number of bounces complements by doubling the magnitude of the signal perceived by the receiver.

Different pairs of eigen-rays have different impact on the signal perceived by a receiver. A typical signal that arrives at the receiver is shown in Figure 4. The receiver and source are 43.4 meters apart. With a constant 1535 m/s sound speed, it takes 0.0282 seconds for the signal to arrive directly after being emitted from the source.

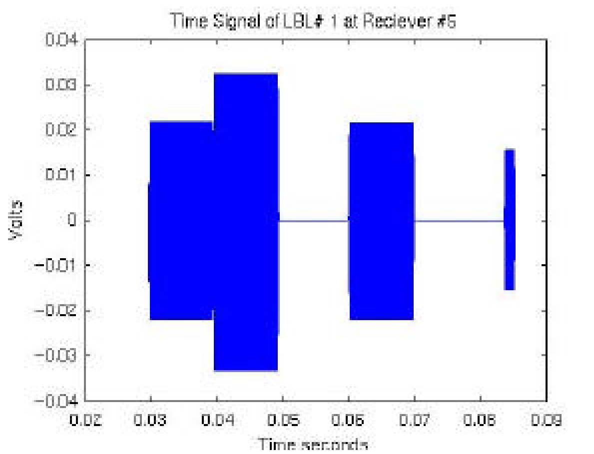


Figure 4: Overall signal perceived by a receiver

The pairs of rays with 1, 3, and 5 bounces arrive at 0.0311, 0.0482, and 0.0710 seconds respectively. However those pairs of rays with an odd number of bounces cancel each other, therefore have no impact on the signal perceived by the receiver. The arrivals of pairs of rays with 2 and 4 bounces occur at 0.0384 and 0.0593 seconds. Note that a signal from an acoustic source (beacon) is transient, and lasts only for 0.01 second, as illustrated in Figure 4.

Typically, the GRT model will stop propagating a ray if it has lost a large percentage of its energy due to repeated bounces [10]. Our simulation assumes perfectly rigid bottom and pressure releasing surface, which will not dampen a signal after being reflected. To run the algorithm more efficiently, the number of bounces was limited to 10.

Evaluation of the Ray Tracing Method

Given our simulation scenario the Method of Images can model the eigen-rays exactly, we can evaluate the Ray Tracing Method by comparing it with an exact solution.

To compare the Method of Images and the Method of Ray Tracing, we compare the signals perceived by distributing 80 receivers uniformly in a 100 x 100 meter grid with an acoustic source (beacon) located at each corner. The geographic locations of receivers and beacons are illustrated in Figure 5. Receivers are numbered columnwise starting from the lowerleft corner. Beacons are numbered counter clockwise starting from the lowerleft corner

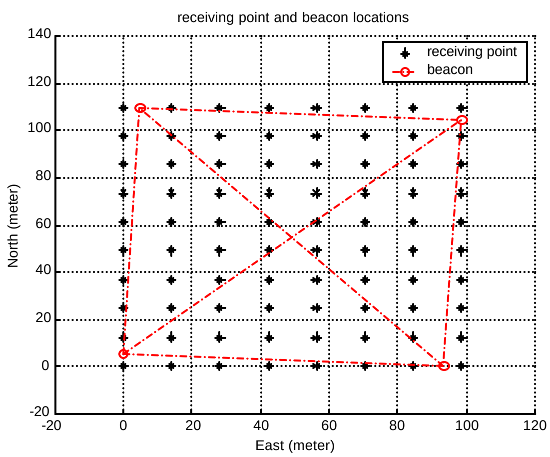


Figure 5: The geographical location of beacons and receiving points.

The Standard Deviation of the signals generated by the Gaussian Method relative to the signals generated by the Method of Images was calculated for each of the beacons and is shown in Figure 6.

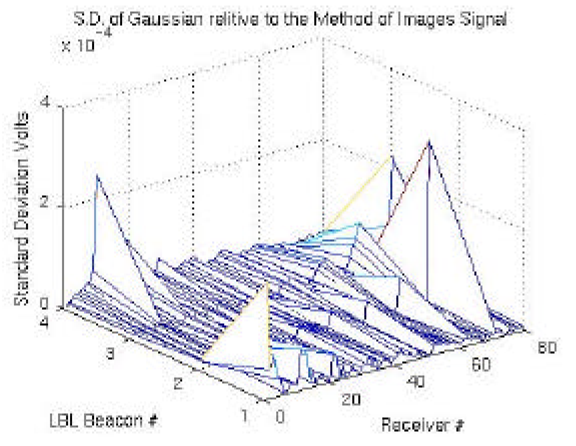


Figure 6: Signals generated by method of images and method of ray tracing

Results show that the Method of Ray Tracing is very close to the Method of Images. The discrepancy becomes large only when the receiver is close to a beacon.

Trigger Level

In practice, the transponder attached to an AUV has certain sensitivity to the acoustic signal. It will not respond if the signal strength is below certain threshold or trigger level. In an LBL system, different trigger levels generate different reply times, and different positioning results. A large trigger level indicates that the transponder is not sensitive to the signal, which may generate an erroneous reply time and position. Therefore, a procedure to set the maximal trigger level is desirable.

Theoretically, the pressure field caused by a point source with strength Q is given as [11]

$$P = Q/r$$

Where r is the distance from the point to the source, P is the pressure at that location.

Since the voltage generated by a receiver is the pressure P at the receiver location, times the receiver sensitivity R, it follows that

$$A = PR$$

Where A is the lowest voltage generated by the receiver, R is the receiver sensitivity. Combined the above two equations together, and cancel P we get:

$$A = RQ/r$$

In our simulation, QR is fixed to 1. The maximal value for r, the largest distance between a receiver and a beacon, is 150 meter. Therefore, the smallest value for A is 0.0067. This value should be used as an upper bound for the trigger level. A trigger level larger than 0.0067 would cause errors in reply time.

Hardware-In-the-Loop Simulation

A Hardware-In-the-Loop (HIL) simulation refers to a simulation where part or all of the system hardware is involved [8]. In the case of AUV application, a connection between AUV hardware and the simulator should be established. Figure 7 illustrates the principle of an AUV HIL simulation.

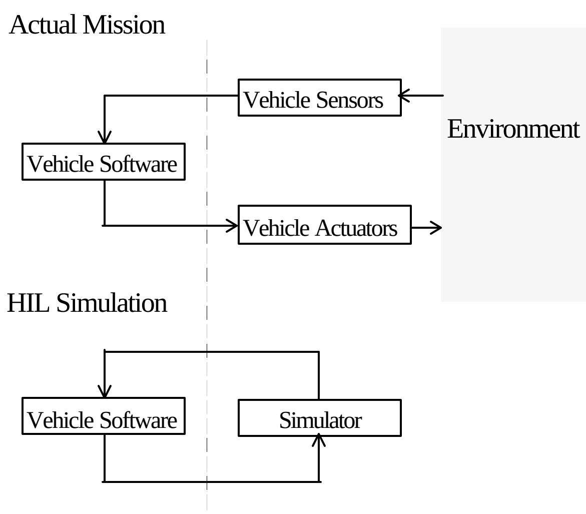


Figure 7: Principle of the AUV hardware-in-the-loop simulation

As illustrated in the figure, the simulator needs to generate all the sensor data based on a vehicle model, and send the data to the vehicle embedded software. In return, it receives the actuator commands from the embedded software, and propagates them to the vehicle model accordingly. All the vehicle software will run on the vehicle hardware platform, receiving sensor data and sending out commands as if the vehicle were in a real mission.

In our implementation, the simulator runs on a PC installed with a Linux operating system, whereas the vehicle software runs on a PC104 stack installed with a QNX operating system. The connection between the Linux machine and the QNX machine is established through TCP/IP sockets.

LBL Simulation Results and Discussions

In an LBL simulation, since all the beacons and the transponders are absent, the reply time needs to be simulated. However, with the GRT Method, the computation of the reply time for a specific vehicle location is time consuming (on the order of days). To simulate the reply time in real time, a table of reply times for a pre-defined number of locations can be pre-computed. The reply time for the vehicle position other than at a pre-defined location can be interpolated in real time during the simulation.

To build a reply-time table, a grid that encompasses all the beacons is divided into a number of cells. The reply time for each of the grid intersections (also known as receiver locations) is computed and stored as an entry in the reply-time table (see Figure 5). To reduce the complexity of the computation process only the one way reply-time is computed off-line and stored in the table. During simulation the round-trip reply time was generated simply as twice the stored one-way reply time based on perfect reciprocity and no turn-around delay assumptions.

A lawn mower mission was carried out in the HIL simulation. Given the reply-time table, a linear interpolation method was applied to generate a reply time for each of the beacons when the vehicle was not at any of the receiver locations.

Suppose $\mathbf{X_1}$, $\mathbf{X_2}$, $\mathbf{X_3}$, and $\mathbf{X_4}$ are four adjacent receiver locations. In the reply-time table, the reply times for the four locations are $\mathbf{T_1}$, $\mathbf{T_2}$, $\mathbf{T_3}$, and $\mathbf{T_4}$ respectively. Suppose $\mathbf{X_1} = [x_1\ \ y_1]^T$, $\mathbf{X_2} = [x_2\ \ y_1]^T$, $\mathbf{X_3} = [x_1\ \ y_2]^T$, $\mathbf{X_4} = [x_2\ \ y_2]^T$, the linear interpolated reply time for the vehicle position $\mathbf{X} = [x\ \ y]^T$ among $\mathbf{X_1}$, $\mathbf{X_2}$, $\mathbf{X_3}$, $\mathbf{X_4}$ is defined as:

$$\mathbf{T} = [\mathbf{T_1} * (1 - s) + \mathbf{T_2} * s] * (1 - t) + [\mathbf{T_3} * (1 - s) + \mathbf{T_4} * s] * t$$

Here $s = (x - x_1) / (x_2 - x_1)$
$t = (y - y_1) / (y_2 - y_1)$.
$x_1 \leq x \leq x_2$, $y_1 \leq y \leq y_2$.

When generating the reply-time table, a trigger level must be set. A trigger level is a measurement of the transponder's sensitivity to the acoustic signals. The smaller the trigger level, the more sensitive the transponder is, the more accurate the reply time is, and the better the simulation result will be.

The simulation results with different reply-time tables generated from different trigger levels are shown in Figure 8, Figure 9, and Figure 10 respectively. The solid blue line is the vehicle true path during the mission whereas the black dots are the estimated vehicle position using the vehicle position estimator software. The red circles in each plot indicate the four beacon positions.

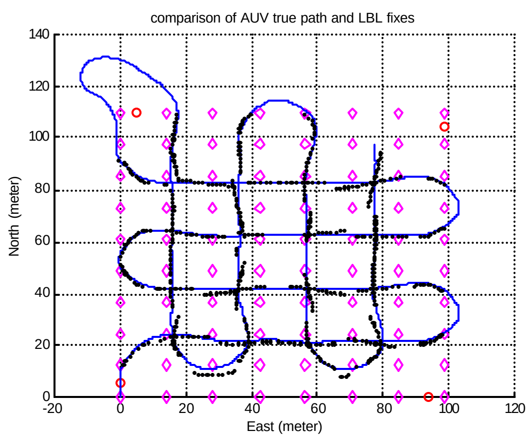


Figure 8: Comparison with 0.005 trigger level

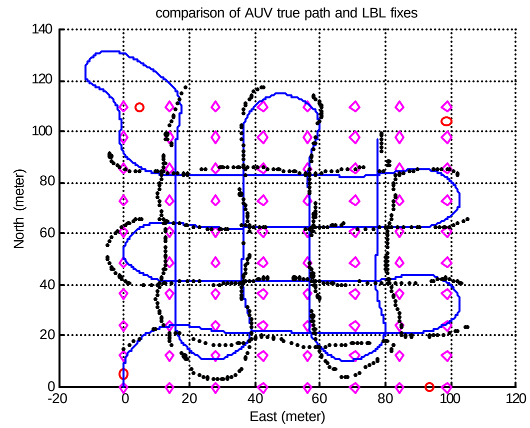


Figure 9: Comparison with 0.01 trigger level

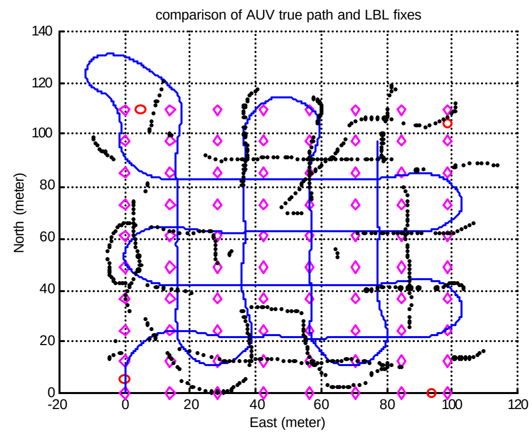
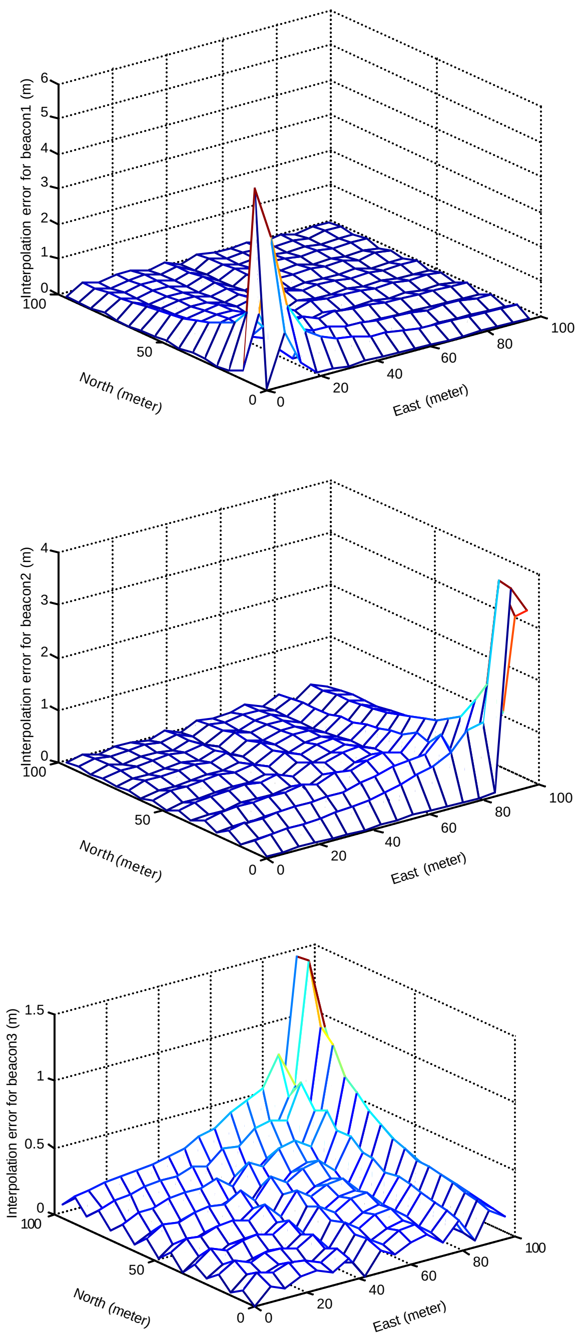


Figure 10: Comparison with 0.013 trigger level

As Figure 8 indicates, even with a very small trigger level, the vehicle embedded LBL software could not always compute a fix. There are several reasons for that, including the errors in the reply-time table, and the errors introduced by interpolation. Figure 11 shows the error introduced by the linear interpolation method for the four beacons. Significant errors (peaks) always appear in the area close to the transmitted beacon. However, most of them are quite small, and can not explain the LBL failures in the center part of the rectangle.



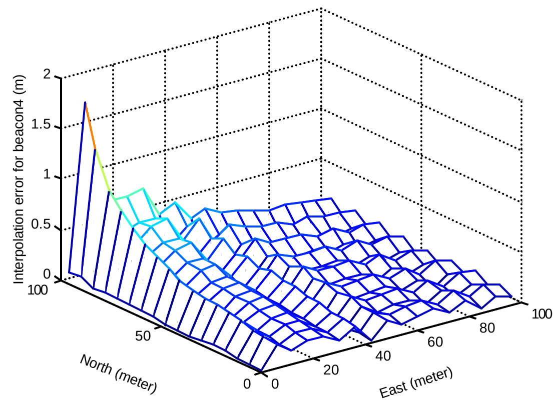


Figure 11: Linear interpolation error for different beacons

These results led us believe that there are algorithmic bugs in the embedded LBL position estimation software. To validate that, another HIL simulation with exact reply time hard coded in the sensor output was conducted. The reply time for a given beacon was computed as its distance to the vehicle divided by the constant sound speed 1535 meter/second. In this case, there was no interpolation performed. Figure 12 shows the simulation result, which is almost identical to that in Figure 8. This simulation explains the LBL failures: There still exist hidden and nonfatal algorithmic errors in the embedded software.

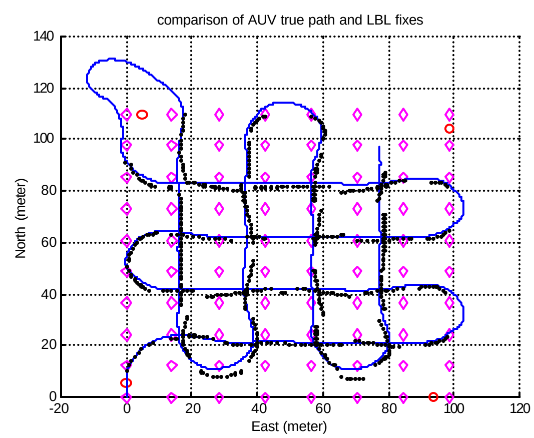


Figure 12: Comparison with exact reply time

Conclusion

Preliminary LBL measurement modeling and results has been presented. The reason for choosing an ideal environment is to allow us to verify the modeling accuracy since closed-form solutions are available for such an environment. The GRT method matched very closely to the Method of Images solution, which indicates that the Ray Tracing Method is an effective method for underwater acoustic modeling. The HIL simulation is a cost-effective way to assess acoustic navigation performance of the embedded software, and the simulation results revealed unexpected holes in the LBL fixes, which could have been caused by hidden bugs in the position code.

Future development includes varying sound velocity profile as a function of depth and range, accounting for varying sea states and using updated bathymetry data of the Atlantic Ocean in the Gaussian Ray Racing Method. An eigen-ray interpolation table for more accurate and efficient results could replace the reply-time table such that the calculation of the reply time can be more realistic and accurate.

Acknowledgements

This research was supported by the Office of Naval Research under the grant number N00014-96-1-5029.

References

[1] J.G. Bellingham, Capabilities of Autonomous Underwater Vehicles, *Scientific Environment Data Collection in Autonomous Underwater Vehicles*, ed J. Moore, MIT Sea Grant Report 92-2, 1992.

[2] J.G. Bellingham, C.A. Goudey, R.R. Consi, C. Chryssostomidis, "A Small, Long Range Autonomous Vehicle for Deep Ocean Exploration *Proceedings of the Second International Offshore and Polar Engineering Conference*, San Francisco CA pp.461-467, 1992.

[3] D. Brutzman, "Virtual world visualization for an autonomous underwater vehicle," *MTS/IEEE OCEANS'95*, pp.1592-1600, 1995.

[4] H.P. Bucker "A Simple 3-D Gaussian beam sound propagation model for Shallow water," *Journal of the Acoustic Society of America*, vol.95, no.5, pp.2437-2440, 1994.

[5] T.B. Curtin, J.G. Bellingham, D. Webb," Autonomous Oceanographic Sampling Networks," *Oceanography*, 6, pp.86-94, 1993.

[6] F. Devie and J. Lemaire, "A Flexible hardware in the loop simulator for a long range autonomous underwater Vehicle," *MTS/IEEE OCEANS'98 Conference*, pp.1359-1363, 1998.

[7] S.E. Dunn, S. Smith, P. Betzer, T. Hopkins," Design of Autonomous Underwater Vehicles for Coastal Oceanography Underwater Robotic Vehicles: Design and Control," *J. Yuh Editor,* TSI Press, pp.299-326, 1994.

[8] K.L. Hall, "What is Hardware-In-the-Loop simulation," *The Society for Computer Simulation Conference*, pp.62-65, 1987.

[9] D.E. Humphreys, "Vehicle hydrodynamic & maneuvering model for the FAU Ocean Explorer vehicle (OEX)," *Technical Memorandum 96-05, Vehicle Control Technologies Inc.,* 1996.

[10] F.B. Jensen, W.A. Kuperman, M.B. Porter, H. Schmidt "Computational Ocean Acoustics", *AIP Press*, 1994.

[11] L.E. Kinsler, A.R. Frey, A.B. Coppens, J.V. Sanders, "Fundamentals of Acoustics," *John Wiley & Sons*, 1982Y.

[12] Kuroda, T. Ura, and K. Aramaki, "Vehicle control architecture for operating multiple vehicles," *Symposium on Autonomous Underwater Vehicle Technology*, pp. 323-329, 1994.

[13] T. LaVigne "A Gaussian Beam Ray Tracing Algorithm Applied to Three Dimensional Underwater Bathymetries," *MTS/IEEE Oceans'96*, pp.170-174, 1996

[14] S.M. Smith, K. Heeb, N. Frolund, and T. Pantelakis, "The Ocean Explorer AUV: A modular platform for Coastal Oceanography," *International Symposium on Untethered Submersible Technology,* pp.67-76,1995.