Virtual Long Baseline (VLBL) Autonomous Underwater Vehicle Navigation Using a Single Transponder

by

Cara E. G. LaPointe

B.S., United States Naval Academy, 1997
M.Phil., University of Oxford, 1999

Submitted to the Department of Mechanical Engineering in Partial Fulfillment of the Requirements for the Degrees of

Naval Engineer
and
Master of Science in Ocean Systems Management
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
June 2006

Signature of Author……
Department of Mechanical Engineering
May 12, 2006

Certified by……
Dana Yoerger, Associate Scientist
Woods Hole Oceanographic Institution
Thesis Supervisor

Certified by……
Henry Marcus, Professor of Marine Systems
Department of Mechanical Engineering
Thesis Reader

Certified by……
Michael Triantafyllou, Professor of Mechanical Engineering
Chairman, Department Committee on Graduate Students
Center for Ocean Engineering

Accepted by……...
Lallit Anand, Professor of Mechanical Engineering
Chairman, Department Committee on Graduate Students
Department of Mechanical Engineering

Virtual Long Baseline (VLBL) Autonomous Underwater Vehicle Navigation Using a Single Transponder

by

Cara E. G. LaPointe

Submitted to the Department of Mechanical Engineering on May 12, 2006
in Partial Fulfillment of the Requirements for the Degrees of Naval Engineer and
Master of Science in Ocean Systems Management

Abstract

Acoustic long baseline (LBL) navigation systems are often used for precision underwater vehicle navigation. LBL systems triangulate the position of the vehicle by calculating the range between the vehicle and multiple transponders with known locations. A typical LBL system incorporates between two and twelve acoustic transponders. The vehicle interrogates the beacons acoustically, calculates the range to each beacon based on the roundtrip travel time of the signal, and uses the range data from two or more of the acoustic transponders at any point in time to determine its position.

However, for accurate underwater navigation, the location of each deployed transponder in the array must be precisely surveyed prior to conducting autonomous vehicle operations. Surveying the location of the transponders is a costly and time-consuming process, especially in cases where underwater vehicles are used in mapping operations covering a number of different locations in succession. During these extended mapping operations, the transponders need to be deployed, surveyed, and retrieved in each location, adding significant time and, consequently, significant cost to any operation. Therefore, accurate underwater navigation using a single location transponder would provide dramatic time and cost savings for underwater vehicle operations.

This thesis presents a simulation of autonomous underwater vehicle navigation using a single transponder to create a virtual long baseline (VLBL). Similarly to LBL systems, ranges in a VLBL are calculated between the vehicle and the transponder, but the vehicle position is determined by advancing multiple ranges from a single transponder along the vehicles dead reckoning track. Vehicle position is then triangulated using these successive ranges in a manner analogous to a 'running fix' in surface ship navigation. Navigation data from bottom survey operations of an underwater vehicle called the Autonomous Benthic Explorer (ABE) were used in the simulation. The results of this simulation are presented along with a discussion of the benefits, limitations, and implications of its extension to real-time operations. A cost savings analysis was also conducted based both on the idea that a single surveyed beacon could be deployed for underwater navigation and on the further extension of this problem that the 'single beacon' used for navigation could be located on the ship itself.

Thesis Supervisor: Dana Yoerger
Title: Associate Scientist, Woods Hole Oceanographic Institution

Acknowledgements

First and foremost, I would like to thank my advisor, Dr. Dana Yoerger, for his continual guidance and support. I am also grateful to Mike Jakuba and Hanu Singh in the Deep Submergence Lab for always making time to answer my endless stream of questions.

I extend thanks to Professor Henry Marcus. The invaluable lessons which I have learned from him and from all of the courses in the Ocean Systems Management program will serve me well throughout my career and beyond.

The United States Navy has once again provided me with an amazing opportunity, and for that, I am eternally grateful. In particular, Commander Timothy McCoy has been incredibly supportive of me.

I would also like to recognize some of the people who are the backbone of MIT and WHOI: Eda Daniels, Leslie Regan, Marsha Gomes, Ann Stone, Pete Beaulieu and Mary Mullowney. They have always pointed me in the right direction, and I never could have done this without them.

Finally, I would like to thank my family. My parents have been unwavering in their support of me throughout all of my endeavors. My sisters Jessica and Amanda are always there at the other end of the phone line when encouragement is needed the most. My brother-in-law Tom gave me a home away from home and a quiet place to study during our interminable house 'remodeling.' The newest members of 'Team LaPointe,' Stuart, Anna and Nawai, have continually reminded me not to take life too seriously. Most importantly, I want to thank my husband Matthew, my rock who keeps me grounded despite everything.

Table of Contents

List of Figures

List of Tables

List of Acronyms

ABE	Autonomous Benthic Explorer
AUV	Autonomous Underwater Vehicle
DR	Dead Reckoning
DRNS	Dead Reckoning Navigation System
DSL	Deep Submergence Laboratory
DVL	Doppler Velocity Log
EKF	Extended Kalman Filter
GPS	Global Positioning System
INS	Inertial Navigation System
KF	Kalman Filter
LBL	Long Baseline
LUSBL	Long & Ultra-Short Baseline
MIT	Massachusetts Institute of Technology
MVLBL	Moving Virtual Long Baseline
SBL	Short Baseline
SLBL	Synthetic Long Baseline
USBL	Ultra-short Baseline
VLBL	Virtual Long Baseline
WHOI	Woods Hole Oceanographic Institution

Chapter 1: Introduction

Section 1.1: Motivation

Underwater vehicle navigation has seen exponential improvements over the last three decades. However, the effectiveness of underwater vehicles, particularly autonomous underwater vehicles (AUV) is still limited by the precision and accuracy of navigation schemes. Underwater vehicles generally rely upon navigation algorithms that incorporate information from onboard sensors with acoustic ranging data from external transponders in known locations. The acoustic ranges are triangulated to determine vehicle position fixes in a global coordinate system, while onboard sensor outputs are used to provide a dead reckoning estimate of vehicle position between position fixes.

While recent advances in inertial navigation systems (INS) and Doppler navigators have achieved very accurate results in dead reckoning navigation during underwater missions, these systems can be prohibitively expensive. Less expensive variations, like a Doppler navigator and a magnetic compass, have error growth that may be unacceptable for many tasks. Small, low cost vehicles for full ocean-depth applications generally rely on acoustic Long Baseline (LBL) navigation systems combined with dead reckoning plots from a suite of less expensive internal sensors. The effectiveness of these systems has been proved repeatedly in real-world operations.

A key characteristic of Long Baseline navigation systems is the requirement to deploy acoustic transponders and to accurately survey their positions. An absolute minimum of two transponders is required for these operations, but in practice four or more transponders are generally deployed to achieve redundancy. The process of deploying, surveying, and, ultimately, recovering each transponder can be time intensive. Operating

expenses of the vessels used to deploy underwater vehicles can be extreme, especially in light of recent dramatic increases in fuel costs.

A combination of conventional, Long Baseline navigation methods and an inexpensive Doppler navigator-based system could provide an accurate cost-effective alternative method. The use of a single LBL beacon could control error growth while taking advantage of the dead-reckoning capability of the Doppler navigator and compass. The development of a navigation system using only one external transponder could provide significant savings over the accumulated course of vehicle operations in multiple locations. This is particularly true on long survey operations in which a minimal number of underwater vehicle dives are done at each of many different locations.

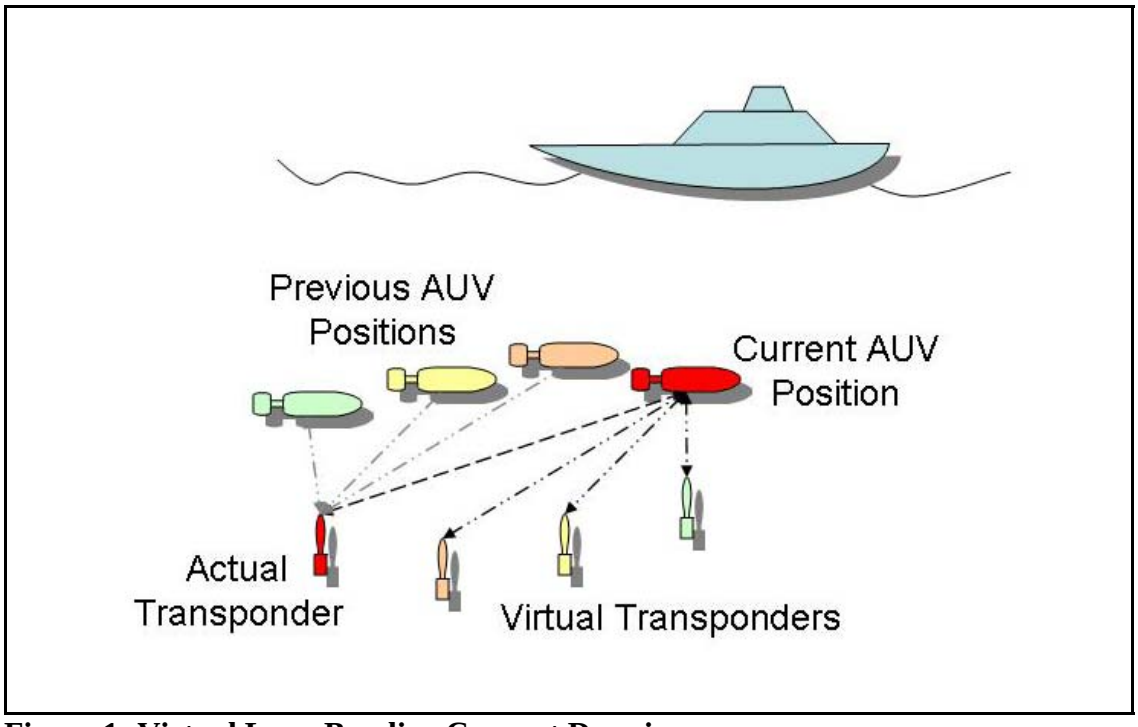

Figure 1: Virtual Long Baseline Concept Drawing

Therefore, the motivation of this thesis was to create a cost-effective navigation system for small, low-cost underwater vehicles operating in the deep ocean. The result was a single transponder navigation model which I have called Virtual Long Baseline (VLBL) navigation. The Virtual Long Baseline algorithm creates a virtual net of acoustic

transponders at any given time by using multiple range measurements from a single transponder taken at different times, which is combined with dead reckoning position estimates of modest quality. As shown in Figure 1, the position of each virtual transponder is created by adjusting the actual transponder position based on the dead reckoning track of the vehicle from the time that the range was taken until the virtual baseline is created. Ranges to each of the virtual transponders in the virtual baseline are used to triangulate, or 'fix,' the vehicle's position in global coordinates.

The Virtual Long Baseline navigation algorithm was applied to both simulated and real-world data sets. The results of the application of the Virtual Long Baseline were compared to the ideas presented in the existing body of literature on single transponder acoustic navigation systems. Much of this literature is based on an approach which uses an adjusted extended Kalman Filter used for real-time position estimation. However, the Virtual Long Baseline approach uses a geometric transformation of the inputs to be used in a standard navigational computational regime. This thesis compares the results of these two approaches.

Furthermore, this thesis presents a cost savings analysis to illustrate the benefits of a real-time application of the Virtual Long Baseline navigational approach.

Section 1.2: Thesis Outline

Chapter Two provides a brief overview of the methods of underwater vehicle navigation and a more specific overview of Long Baseline Acoustic Navigation systems.

Chapter Three explains the development of the Virtual Long Baseline model using a single external acoustic transponder. A basic review of existing literature on single transponder navigation is included.

Chapter Four shows the application of the Virtual Long Baseline navigation model using simulated and real-world data from deep-ocean bottom survey operations of the Autonomous Benthic Explorer underwater vehicle.

Chapter Five contains an analysis of the cost savings which could be realized through real-world implementation of Virtual Long Baseline navigation.

Finally, Chapter Six highlights the specific contributions of this thesis along with possibilities for future work.

Chapter 2: Underwater Vehicle Navigation

Section 2.1: A Brief Review of Underwater Vehicle Navigation

Dramatic technological advances in underwater vehicles over the last four decades have exponentially increased the mission potential of these vehicles. The development of untethered, autonomous underwater vehicles (AUVs) in particular has widened the scope of military, commercial and scientific applications for which underwater vehicles are used. As onboard sensor and mission packages for these vehicles continue to increase in functionality and sophistication, the mission performance capabilities of AUVs are often limited by the precision and accuracy of their navigational systems. [1] There are three main categories of underwater vehicle navigation: (1) dead reckoning and inertial navigation techniques, (2) external acoustic systems, and (3) vision-based and terrain mapping algorithms. Research in all of three of these fields has yielded increasingly sophisticated systems which differ primarily in their cost, size and power requirements. Each of these will be discussed with respect to their appropriateness for use in small, low-cost, deep-ocean vehicle applications.

Section 2.1.1: Dead Reckoning and Inertial Navigation Systems

The most basic navigational techniques for underwater vehicles are those of dead reckoning (DR) and Inertial Navigation Systems (INS). In both of these systems, the vehicle is given an initial position and then uses information from onboard sensors to repeatedly update its position estimate. Since the vehicle position is not reinitialized

during the course of underwater operations, errors in the estimated position accumulate throughout the mission. These errors arise from a number of sources such as inherent error of the onboard sensors, and external environmental forces which are not adequately observed by the sensors used in the particular navigation system. [1]

In dead-reckoning navigation, vehicle velocity is integrated with respect to time in order to estimate the path of vehicle travel. The most primitive dead-reckoning systems estimate speed using *a priori* calibrations of propeller speed versus vehicle water speed. This method only generates an approximation of forward speed without accounting for any current or sideslip effects. In practice, these systems are not tenable for low-speed vehicles such as AUVs. Therefore, navigation systems incorporate accurate velocity measurements such as those from Doppler Velocity Logs (DVL), which measure vehicle speed relative to either the seafloor or water column.

Similarly, in basic DR systems, heading can be determined by magnetic compass alone. However, magnetic compasses may be subject to large variable errors, especially near the ocean bottom where underwater features can cause compass heading to deviate drastically from magnetic north. Therefore, gyrocompasses are incorporated into DR systems to improve the accuracy of heading measurements.

A further improvement in the concept of dead-reckoning navigation is that of the Inertial Navigation System (INS), which generally incorporates an inertial motion/measurement unit with a Kalman Filter (KF) algorithm. Vehicle acceleration measurements from the inertial motion unit are integrated twice with respect to time in order to derive vehicle velocity. The KF is the control algorithm which then incorporates knowledge of the vehicle's prior position, sensor inputs, and a dynamic model of the system to estimate the vehicle's current position.

The fundamental problem with using either DR or INS navigation systems as the sole method of underwater vehicle navigation is that the position estimate error continually increases with time and distance traveled. Many basic INS systems have position drift rates on the order of one to two percent of distance traveled. [1, 2] In shallow water

operations, where the vehicle can periodically surface and reinitialize the navigation system using inputs from the Global Positioning System (GPS), inexpensive INS systems can be very effective. However, for deep water operations, frequent surfacing for system initialization is not possible. Although extremely accurate INS-based navigation systems exist, their prohibitive cost, size and power requirements have traditionally rendered them completely inappropriate for small, low cost vehicles. Advances in component technology continue to drive down the cost and size of highly accurate INS navigation systems, but, to date, external acoustic positioning systems remain the standard for scientific missions by small, low-cost underwater vehicles. [1, 2]

Section 2.1.2: External Acoustic Systems

External acoustic positioning systems are used by underwater vehicles to triangulate their position based on range only or bearing and range information between external acoustic transponders and a transducer mounted on the vehicle. A primary advantage of these systems is that the size and power requirements in the underwater vehicle are minimal compared to navigational methods. However, unlike the other navigational methods, some of the external acoustic systems require the deployment of acoustic transponders to the seabed in the vicinity of operations.

In an external acoustic system, the vehicle calculates its range to each transponder using acoustic signal time of flight and an estimate of the speed of sound in the water column between the vehicle and the transponder. The availability of bearing information is dependent upon the geometry of the acoustic transponder net. Three different primary geometries are used for external acoustic navigation systems: Short Baseline (SBL), Ultra-Short Baseline (USBL), and Long Baseline (LBL). Although other hybrid systems exist, such as the Long & Ultra-Short Baseline (LUSBL), they are based on elements of the three primary geometries, so they will not be discussed independently. [3]

As a point of clarification, although the terms ‘transponder’ and ‘beacon’ are often used interchangeably in recent literature, early literature on navigation made a technical distinction between the two. According to P.H. Milne in his seminal book on the subject

of underwater acoustic positioning systems, a ‘transponder’ sends out an acoustic response only when interrogated, whereas a ‘beacon’ sends out an acoustic signal at predetermined intervals so its clock must be exactly synchronized with that of the vehicle. [4] In this thesis, the terms ‘transponder’ and ‘beacon’ both refer to Milne’s definition of transponder, one which is responds only when interrogated by a transducer.

All types of external acoustic positioning systems experience some common challenges. One such challenge is that of achieving coordinate system compatibility between all measurements. Sensor orientation, vehicle attitude and host-vessel attitude for hull-mounted transducers all affect coordinate system transformations. The following discussions on the different navigation system geometries all assume that the proper coordinate transformations have been conducted in order to calculate vehicle position in a common reference frame. The most fundamental challenge for underwater vehicle navigation is effective rejection of outliers and spurious returns. Real-world acoustic transmissions can produce complicated multi-path scenarios, whereas navigation algorithms are developed under the assumption of direct path transmissions. Therefore, the spurious returns must be identified and rejected to achieve accurate underwater navigation. Extensive study has gone into overcoming this challenge; see for example the paper on outlier rejection by Vaganay et al. [5] The following discussions assume that outlier rejection is accomplished as part of the position calculation process.

Section 2.1.2.1: Short Baseline Navigation

The earliest type of external acoustic system developed was the short baseline (SBL) acoustic positioning system, which is used for tracking or navigation of underwater vehicles over short ranges. Primitive SBL systems were used as early as 1963 with limited success when one was installed on the USNS *Mizar* for navigating the submersible *Trieste I* during the search for the USS *Thresher*, a nuclear submarine lost at sea in April of that year. [4]

These systems incorporate a single transponder or transducer mounted on the underwater vehicle and an acoustic net usually mounted on the hull of the host vessel.

The acoustic net is made up of a combination of three or more acoustic transducers, hydrophones and transponders, which are mounted on the hull of the ship so as to achieve the maximum feasible geometric separation, which is generally on the order of 10 to 20 meters, as shown in Figure 2. The geometry of the hull-mounted acoustic net must be precisely surveyed upon initial installation of the system. [3]

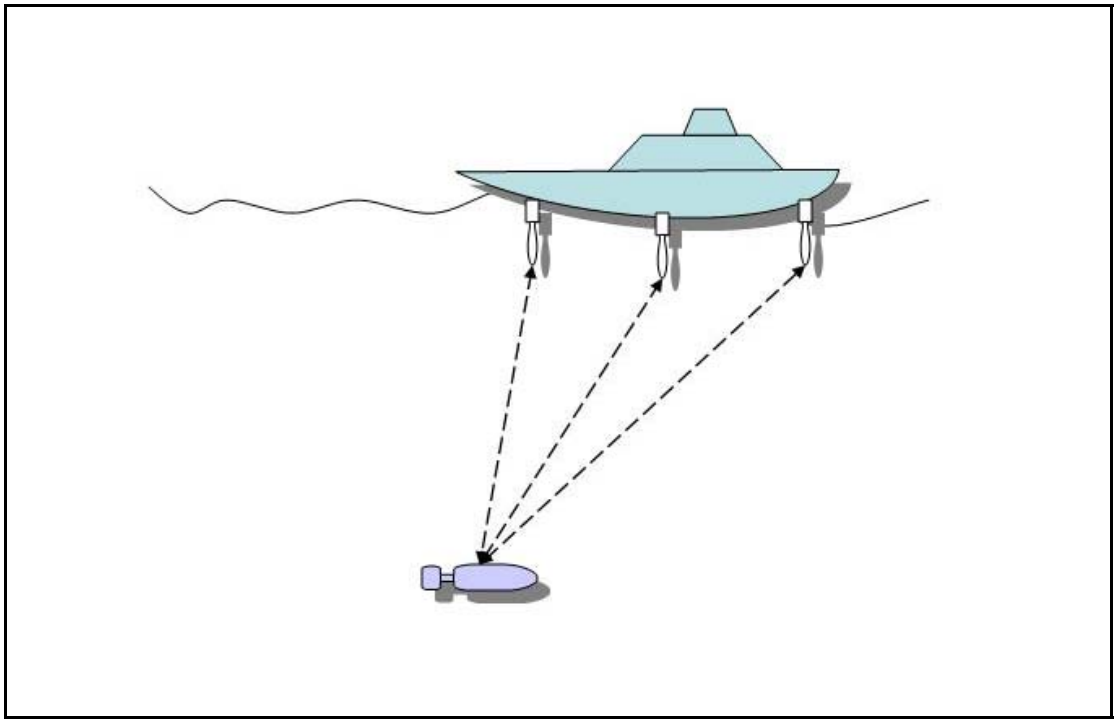

Figure 2: Short Baseline Acoustic Positioning System Geometry

In a vessel-operated tracking configuration, a transponder is mounted on the underwater vehicle and the SBL net will typically include one or two transducers and several hydrophones. One transducer in the acoustic net interrogates the transponder on the vehicle, and all of the elements in the acoustic net receive the transponder response. Ranges between the vehicle and each element of the acoustic net are then calculated and used to determine vehicle position.

For autonomous vehicle navigation, the geometry is inverted, so that a single transducer is located on the AUV and the acoustic net is made up of transponders. Although the acoustic net is often hull-mounted, systems have been developed in which

the acoustic net is mounted in a known geometry on a deployable frame as well. [6] In this configuration, the AUV interrogates the acoustic net and calculates its own position estimate relative to the location of the acoustic net. For the AUV to determine its global position, the acoustic transponder net must either remain in a fixed location or its location at each interrogation must be conveyed to the AUV through acoustic communication.

Section 2.1.2.2: Ultra-Short Baseline Navigation

In the 1970s, ultra-sort baseline (USBL) navigation systems were developed as a simpler alternative to SBL systems. [4] These USBL systems can be operated from either the underwater vehicle or its host vessel. USBL systems operated from an AUV, which are sometimes called inverted USBL systems, allow the AUV to navigate relative to the location of a single external acoustic transponder.

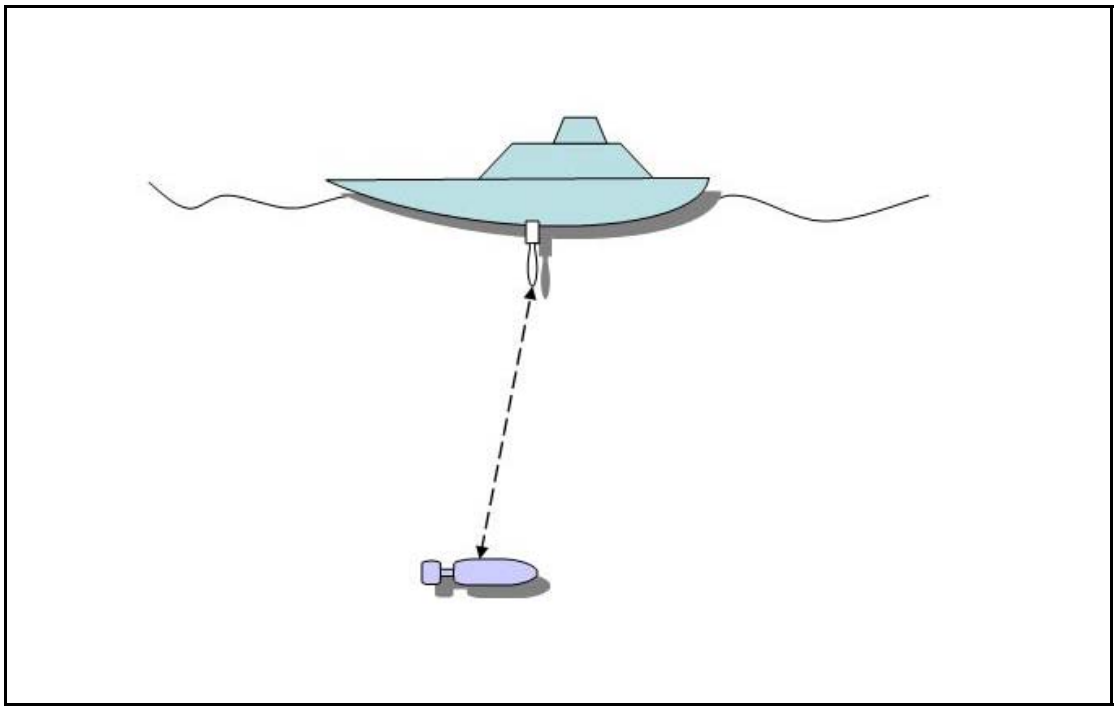

Figure 3: Ultra-Sort Baseline Acoustic Positioning System Geometry

If the transponder is hull-mounted on the host vessel, the AUV navigates relative to the host vessel position. If the transponder is bottom mounted with known geodetic

coordinates, the vehicle can navigate in true coordinates. In this scenario, there is a multi-element receiver array built into a single transceiver assembly which is located somewhere on the AUV. [1, 4] The USBL geometry is shown in Figure 3.

Systems operated from the host vessel are used for tracking or for navigation of remotely operated vehicles. These host vessel systems can provide navigation in global coordinates whenever the location of the host vessel can be precisely determined using GPS. In this scenario, the single acoustic transponder is attached to the underwater vehicle and the transceiver array is located on the host vessel. USBL systems operating at frequencies on the order of 100 kHz can be used for short range navigation tasks on the order of 100-500 m. Deep ocean USBL systems also exist, such as the Posidinia system which is effective to 6000 m and operates at frequencies of 14.5 to 17.5 kHz. [1, 7, 8]

The key difference between USBL and the other types of acoustic positioning systems is that USBL uses the differences in phase of the acoustic signals received by the different transducer sensors to determine bearing to the transponder as well as range. Since the transducer sensors are in a precisely known geometry, the difference in phases between the signals received by the different sensors can be used to compute mechanical angle of incidence. This mechanical angle of incidence can, in turn, be used to compute the bearing between the transponder and the transducer array. [4] Depending upon the particular system in use, the underwater vehicle position can then be estimated relative to the host ship or external acoustic transponder, using the bearing and range information.

Section 2.1.2.3: Long Baseline Navigation

In principle, Long Baseline (LBL) navigation systems are similar to inverted SBL systems, with the difference that the external transponders are deployed individually into the ocean instead of being mounted on the hull of a host vessel or deployable frame. The obvious consequence of this difference is that the geometry of the transponder net is no longer known *a priori* and needs to be determined on site. Typical LBL systems deploy between four and twelve acoustic transponders, depending upon vehicle mission, although they can operate with as few as two transponders. In order for a vehicle to

navigate in a global reference system, the global locations of the beacons need to be determined and conveyed to the underwater vehicle. [2] Self-calibrating beacons exist which can determine their positions relative to one another. These beacons are more expensive, but they reduce survey time because only sufficient survey data to fix the calibrated net into global coordinates are required. [1]

The acoustic transponders are deployed from the host vessel in the vicinity of impending underwater vehicle operations, and then the transponder array geometry is calibrated by 'surveying' the location of each transponder. This calibration is accomplished by repeatedly interrogating each transponder from a transducer located on the hull of the host vessel. The host vessel transits to various locations, interrogates the transponders, and then uses accurate ship position data from GPS coordinates in combination with the calculated range to each transponder in order to globally locate the transponders. This calibration, or 'surveying,' of the transponder net is often the largest source of error in LBL navigation systems. [4]

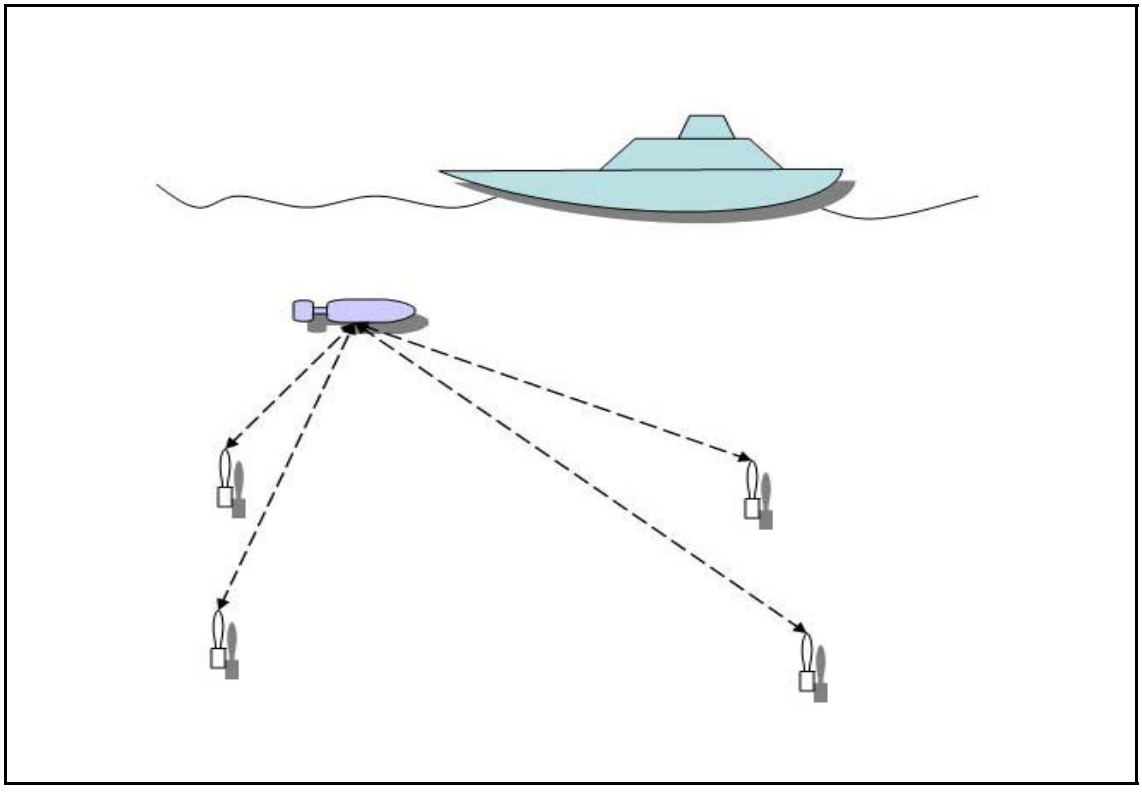

Figure 4: Long Baseline Acoustic Positioning System Geometry

Once the three-dimensional geometry of the transponder net has been surveyed, the location of each of the transponders is then communicated to the underwater vehicle. The vehicle navigates by periodically interrogating the transponders, and computing ranges to each beacon based on the time of flight of the interrogation process, as shown in Figure 4. Typically, the vehicle interrogates all the beacons on a single master frequency and they each respond at a unique frequency. The vehicle uses its best estimate of water column sound speed, multiplied by one half of the round trip time of flight for each beacon, to calculate the range to that beacon. [2] LBL systems have been designed using different frequency ranges to accomplish different missions. Although the fundamental principles and method of operation are identical, the variation in acoustic signal frequencies allows for different levels of accuracy over different effective ranges of operations. Typical high frequency systems operate on the order of 300 kHz, while low frequency systems operate on the order of 12 kHz. The differences in performance characteristics are summarized in Table 1.

Table 1: Performance Characteristics of Low and High Frequency LBL Navigation Systems [9]

SYSTEM	UPDATE RATE	TYPICAL PRECISION	EFFECTIVE RANGE
12 kHz	0.1 to 1.0 Hz	0.01 to 10 m	5 to 10 km
300 kHz	1.0 to 5.0 Hz	+/- 0.002 m	100 m

Depending on the number of ranges available at the end of each interrogation cycle, the vehicle position is calculated in different ways. Depth information from sensors onboard the vehicle and the transponders is used to reduce the triangulation problem to the two-dimensional horizontal plane. If ranges from only two external transponders are available, the vehicle calculates the two points of intersection of the range circles from these transponders. Using an *a priori* awareness of which side of the transponder baseline it is located, the vehicle can determine which of the two possible solutions represents its current position. Three or more transponder ranges allow the vehicle to uniquely fix its position using a least squares method. [10]

Once vehicle position has been 'triangulated' using ranges from the LBL transponders, the vehicle then computes its dead-reckoning track until the next set of LBL ranges is available. After each interrogation cycle, vehicle position is determined and used to reinitialize the dead-reckoning track. The update rate at which the system can be reinitialized is fundamentally limited by the speed of sound in water, which is approximately 1500 meters per second in water. Integrating information from a DVL into the dead-reckoning solution has been shown to dramatically improve LBL navigation rates, especially in lower frequency setups with slower update rates. [11] This system uses complementary linear filters to combine low-passed LBL position fixes, which are noisy but globally precise, with high-pass Doppler position fixes which are precise over short periods. [2]

An alternate computational algorithm for AUV navigation using an LBL system employs a Kalman Filter (KF). A Kalman Filter incorporates information from onboard sensors and *a priori* knowledge of the inaccuracies of these sensors with a dynamic state space model of the overall system to provide real-time state estimates. [12] The vehicle position is initially determined in the manner described above, but the position estimate updates are computed via careful application of a KF. Since underwater vehicle motion is nonlinear, an extended Kalman Filter (EKF) is used instead. However, a major limitation for the use of any kind of Kalman Filter with LBL navigation systems is due to the nature of noise in an LBL system. Systematic errors in a typical LBL setup include water speed uncertainty; initial beacon position survey errors; movement of beacons due to currents; acoustic multi-path; loss of direct acoustic path; and poor signal-to-noise ratios due to machinery and electromagnetic noise. Therefore, some of the resulting LBL fixes are inaccurate with large, non-Gaussian errors. [2, 13] These non-Gaussian errors violate a fundamental assumption of Kalman Filter algorithms that system noise is Gaussian.

At the end of the underwater vehicle operations in an area, the acoustic transponders must be recovered. In general, most sea-floor transponders incorporate an acoustic release. When a particular coded acoustic signal is sent to the transponders, a weighted

mooring tether is released and the acoustic transponder floats to the surface for recovery. [2] Despite the cost and time required for acoustic transponder handling, LBL navigation systems remain the standard for low-cost, deep-ocean vehicle operations.

Section 2.1.3: Geophysical Navigation

Another newer technique in underwater navigation is that of geophysical navigation, using vision-based or terrain mapping methods. Geophysical navigation is achieved by measuring geophysical parameters, such as bathymetry or magnetic field anomalies, and matching them to a map of the operating area. Although the idea of navigating at sea by comparing depth soundings to bathymetric maps has a long history in surface navigation, the successful application of this concept to underwater vehicles is relatively recent. The main motivation for geophysical mapping techniques is to achieve highly accurate navigation in any location without the need to first deploy an acoustic net. [1]

Geophysical navigation for AUVs has been successfully demonstrated in several instances using *a priori* underwater maps, and a plethora of research is currently ongoing within the field of concurrent mapping and localization. Concurrent mapping and localization eliminates the need for *a priori* maps by using real-time correlation of multiple images to create a bathymetric map during a mission and simultaneously using that map for navigation by periodically reinitializing the dead-reckoning solution. [1, 14] Significant research in this field has been done, *inter alia*, by John Leonard and his students at the Massachusetts Institute of Technology (MIT) and the Woods Hole Oceanographic Institution (WHOI), and by Paul Newman and his students at the University of Oxford. [1, 15-18] See, for example, the doctoral dissertations of Ryan Eustice and Christopher Roman from the Deep Submergence Lab (DSL) at WHOI for navigation algorithms which incorporate terrain mapping features to improve navigational accuracy of underwater vehicles. [19, 20] Although geophysical navigation techniques hold significant promise for the future, they are not yet at the point of reliable, wide-spread use in AUVs as a primary means of navigation. Therefore, the primary means of navigation for small, low-cost vehicles remains acoustic navigation systems.

Section 2.2: Overview of Single Beacon Navigation Research

Over the past decade, interest in the idea of using a single external beacon in acoustic positioning systems has become increasingly popular. A number of researchers have studied the issue from slightly different approaches. The fundamental issue throughout the literature is the question of observability, in other words, determining what conditions are necessary to generate an accurate position fix using only range data from a single external acoustic transponder in conjunction with information from onboard sensors.

Section 2.2.1: Least Squares Approach

One of the earliest presentations of a single beacon positioning system in the literature was by Alexander Scherbatyuk of the Institute for Marine Technology Problems, Far East Branch of the Russian Academy of Sciences in 1996. [21] The approach for his AUV Positioning Algorithm was to use a least-square root method to solve for vehicle position and current velocity in the horizontal plane. The required inputs to the system were ranges between the AUV and a single transponder from an LBL transponder net calculated in the usual way; vehicle yaw information from either a gyrocompass or magnetic course transducer; vehicle velocity information from a velocity transducer logging in either relative or absolute mode; and measured vehicle depth. (See Appendix A for a mathematical model of this least squares approach.) Scherbatyuk concluded that the vehicle needed to obtain ranges to the transponder while steady on three different straight-line trajectories in order to determine vehicle position.

Section 2.2.2: Extended Kalman Filter Approach

Early work on applying an Extended Kalman Filter (EKF) to acoustic, range-only, single transponder navigation systems for small, low-cost AUVs was done by Jerome Vaganay, Phillipe Baccou, and Bruno Jouvencel from the Laboratoire d'Informatique de Robotique et de Microélectronique de Montpellier at the Université Montpellier II in

France. They initially presented this approach in the context of a homing algorithm at the Oceans 2000 MTS/IEEE conference [22], then Baccou and Jouvencel went on to present extensions of the algorithm to vehicle navigation at robotics conferences in 2002 and 2003. [23, 24] Their approach for single transponder homing and navigation used a nonlinear least squares method for position initialization incorporated with an EKF which then provided constant updates of the vehicle estimated position. The required inputs to the system were ranges between the AUV and a single transponder from an LBL transponder net calculated in the usual way, reliable vehicle heading and depth information, and an approximation of vehicle water speed as a function of propeller shaft speed based on a priori calibration. Their analysis assumes that Doppler Velocity Logs are not available to provide vehicle velocity information. The conclusion which they drew from the results their simulations was that a single beacon navigation system using an EKF was a robust method worthy of further investigation.

A second approach for using an EKF in conjunction with a single beacon was also presented at the Oceans 2000 conference by Mikael Larsen of the Technical University of Denmark. [14] His approach, which he called Synthetic Long Baseline (SLBL) navigation, used a cascaded Kalman Filter mechanization to calculate vehicle position from the Dead Reckoning Navigation System (DRNS) outputs and the range measurement discrepancies as shown in Figure 5. The system was cascaded because a second error state Kalman Filter was used in the DRNS itself. Larsen's conclusion based on simulated and post-processed data sets was that SLBL could be used to provide sub-meter accuracy in a 1 km by 1 km survey site in the deep ocean. [14, 25]

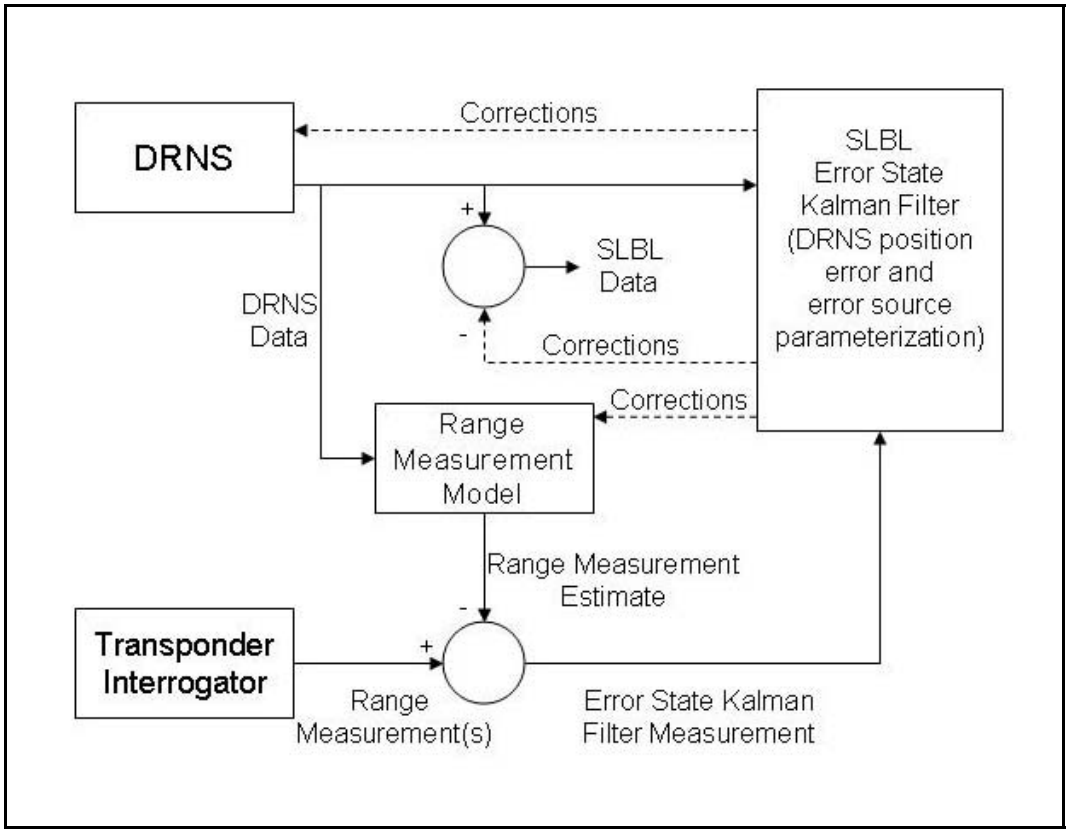

Figure 5: Synthetic Baseline Navigation Approach [14]

More recent work on single beacon navigation has been done by Aditya Gadre and Daniel Stillwell of the Virginia Polytechnic Institute and State University. At the 2004 International Conference on Robotics and Automation, they presented theoretical research on a precise observability analysis of underwater vehicle trajectories in an EKF single beacon navigation system. [26] Their analysis tested for local observability by linearizing the kinematic system model and applying the Observability Rank Test. They concluded that the all vehicle trajectories in such a system are locally observable, with the exception of straight-line trajectories traveling directly towards or away from the single beacon. (See Appendix A for a mathematical model of an EKF approach.) In 2005, they presented an extension of their research in the presence of unknown currents. Based on theoretical analysis and simulations, they confirmed their earlier conclusions on observable trajectories and asserted that in the presence of slowly varying unknown

currents, the estimation errors were negligible enough that the algorithm should be usable for real-time analysis. [27]

In 2005, at the Unmanned Untethered Submersible Technology Conference, Andrew Ross and Jerome Jouffroy of the Centre for Ships and Ocean Structures at the Norwegian University of Science and Technology presented a similar observability analysis of a single beacon, EKF navigation algorithm. The assumed input to their theoretical and simulated research was an unmanned underwater vehicle equipped with a gyro-compass, bottom-lock DVL, and a transponder. Their analysis differed from that of Gadre and Stilwell by using the Observability Rank Condition (ORC) for nonlinear systems to test for local observability, instead of linearizing the system. However, their results were similar, finding that trajectories straight towards or away from the beacon were unobservable. Furthermore, they noted that vehicle position estimates could converge on a mirror image of the actual track for other straight-line trajectories if the initial position estimates were not accurate. [28]

Chapter 3: Development of the Virtual Long Baseline Navigation Algorithm

The Virtual Long Baseline navigation algorithm was developed to allow an underwater vehicle to calculate, or 'fix', its globally referenced position using a single external acoustic transponder. Multiple asynchronous ranges from the same transponder are manipulated to create a long baseline of virtual transponders in different locations at a single point in time. Using the locations of and ranges to these virtual transponders, an underwater vehicle can then compute its global location in the same way that it would compute its location using multiple transponders in a traditional Long Baseline system.

Section 3.1: Defining the Virtual Long Baseline

The motivation for the virtual long baseline approach came from my experience as the navigator on a United States Navy Arleigh Burke Class destroyer. In surface navigation, a 'running fix' is used to determine ship's position when only one navigational aid is available. The position of the navigational aid is recorded at three separate times and then these positions are each advanced along the ship's dead-reckoning track through the appropriate time steps. Then all three positions can be compared at a single time to determine the ship's position.

A similar idea was used to develop the Virtual Long Baseline approach. The underwater vehicle interrogates the single transponder at multiple points in time. The calculation of range between vehicle and transponder is then maintained in the vehicle memory in a historical record. When a sufficient number of ranges have been recorded, the vehicle calculates the location of the virtual transponder that corresponds to each of those ranges, based on the vehicle's dead-reckoning track through the corresponding time steps.

Similarly to LBL navigation, VLBL could calculate a position fix with as few as two ranges. However, for robustness and to achieve the maximum physical separation between the virtual transponders, the VLBL algorithm was designed to use four range values to compute vehicle position. The importance of physical separation between virtual transponders will be discussed below with the subject of system observability.

Accurate depth sensors on both the vehicle and the transponder allow the navigation problem to be treated two-dimensionally in the horizontal plane. The depth values are used to geometrically transform the slant range between the vehicle and the transponder into a horizontal range.

Section 3.1.1: General VLBL Geometry

The geometry of the VLBL is based on a virtual net of acoustic transponders. In actuality there is only one acoustic transponder located in a fixed, known position. Since the VLBL algorithm uses ranges taken between the vehicle and the actual transponder position, AT, at multiple points in time, the transponder location associated with each historical range value must be adjusted accordingly so that the ranges can all be compared at the current time.

Certain assumptions and simplifications are made in this discussion of geometry development. They are discussed fully in Section 3.2.2; however, for clarity, they are stated here as well. Although only two ranges are required to generate a position estimate of the vehicle, the VLBL algorithm determines vehicle position using four ranges in a least squares calculation routine. Therefore, over the course of the time necessary to get a single position fix using VLBL, the underwater vehicle has been in four separate locations when it calculated range to the transponders, depicted in Figure 6 as X1 to X4. Furthermore, the ranges calculated at each time step undergo a series of tests in order to determine whether they are ‘good ranges’. The time steps referred to in this discussion are the time steps at which four consecutive ‘good ranges’ were recorded. Therefore, the time delays between time steps are not necessarily equal in duration.

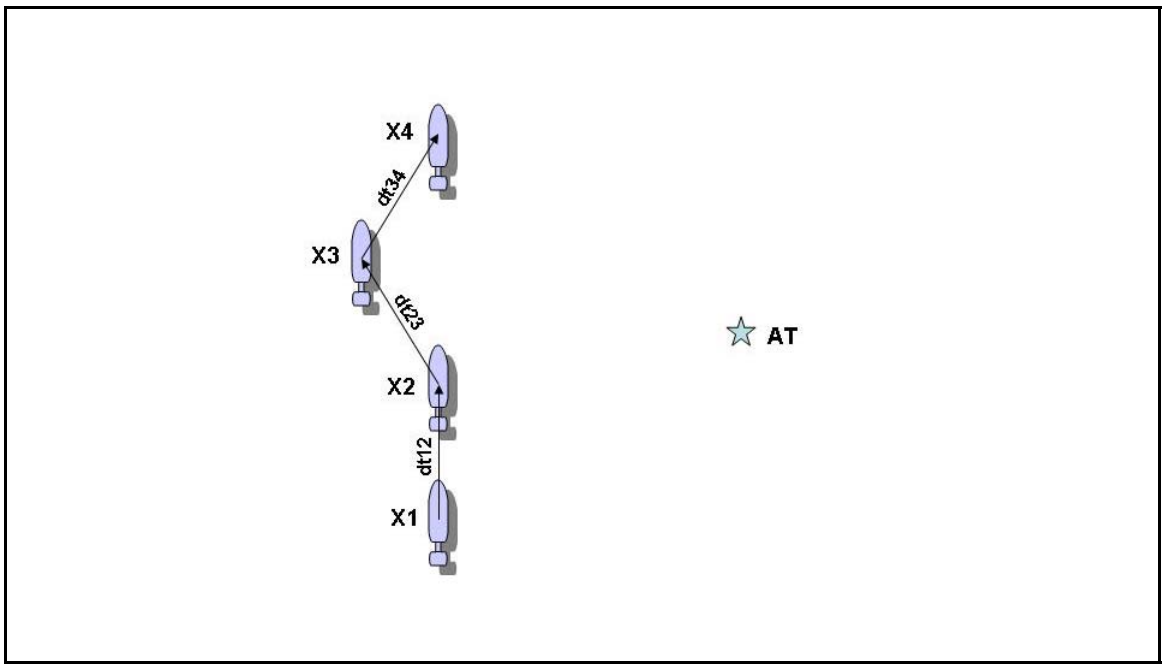

Figure 6: VLBL Vehicle Dead Reckoning Track

In order to build the virtual transponder net, one must start at the last time step, T4, corresponding to position X4 of the vehicle, and work backwards. The following series of figures shows this development of the geometry working backwards from T4, when the position fix is calculated.

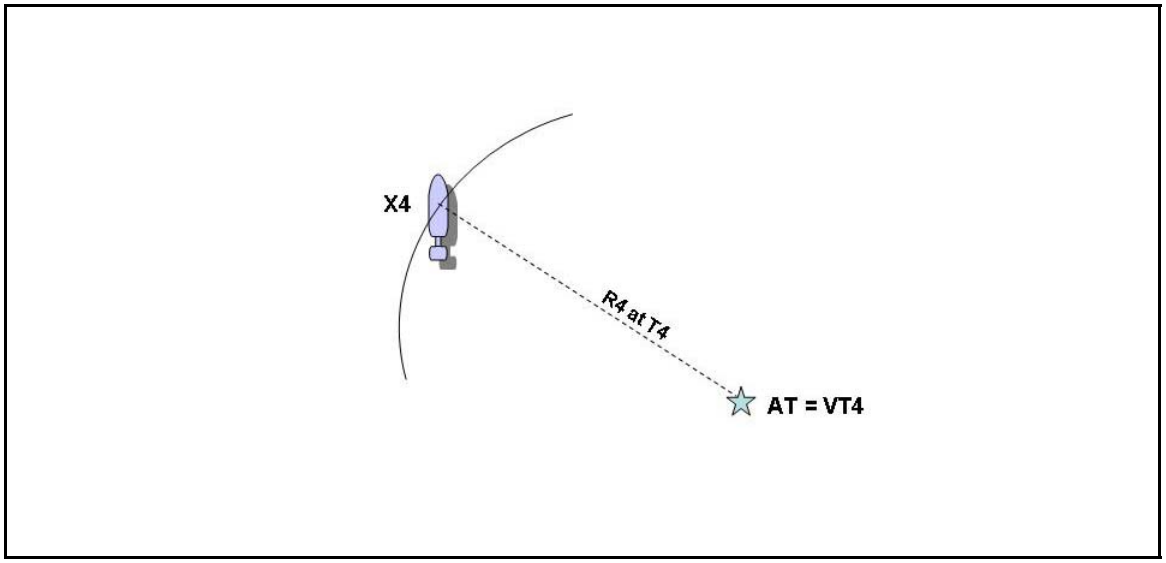

Figure 7: VLBL Geometry Development Time Step Four

At T4, the vehicle is located at X4 and it calculates a range to the beacon, R4. This is illustrated in Figure 7 as "R4 at T4". Since the position fix is actually calculated at T4, the actual transponder location, AT, is the same as the virtual transponder location, VT4, corresponding to R4.

At time step three, T3, the vehicle was at position X3 where the range R3 was recorded. This is shown in Figure 8 as "R3 at T3". In order for the vehicle to be able to use this range at T4 in the fix determination, R3 must be properly translated through time and space so as to be compatible with R4 at T4. A 'virtual transponder' location, VT3, is created by advancing the actual transponder location in the direction and distance that the vehicle traveled during the time delay between T3 to T4, notated as dt34. The best estimate of direction and distance of the vehicle during that time is along the dead reckoning track of the vehicle, as shown in Figure 8.

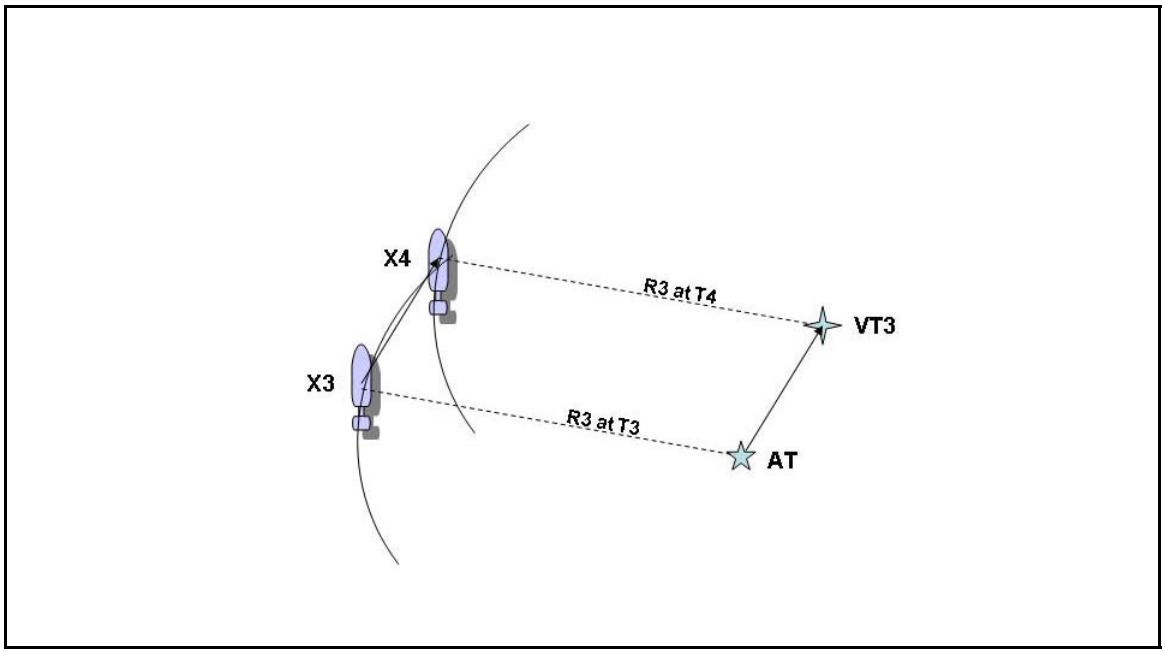

Figure 8: VLBL Geometry Development Time Step Three

The exact same process that was used to advance R3 to T4 is also used to advance R2 and R1 to T4. The only difference is that the position of each transponder must be

adjusted for the vehicle location from each additional time step. The general equation for this adjustment is Equation (3.1) as follows:

$$VT_i = AT + \sum_i (v_i * dt_{(i,i+1)}) \qquad (3.1)$$

where

$dt_{(i,i+1)}$ = Time delay between time steps i and $i+1$

v_i = Vehicle velocity at time step i

AT = Actual transponder location

VT_i = Virtual transponder location corresponding to R_i

R_i = Range between vehicle and AT at time step i

Therefore, R2 must be advanced through both dt23 and dt34, while R1 must be advanced through dt12, dt23 and dt34, as shown in Figure 9 and Figure 10, respectively.

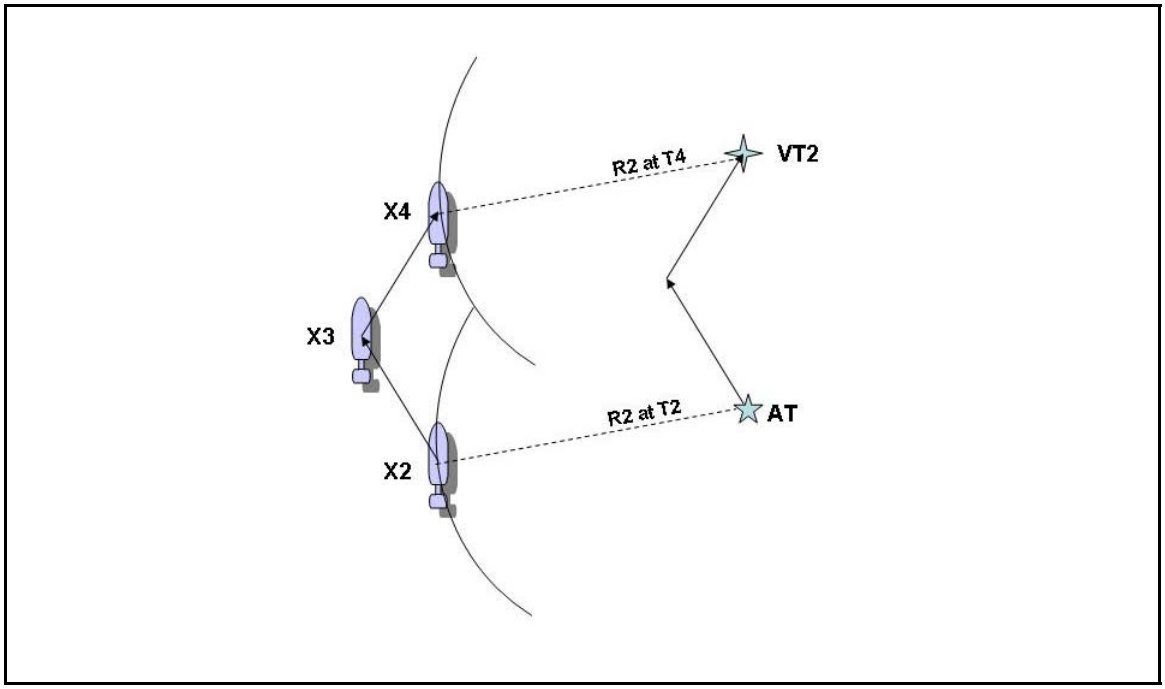

Figure 9: VLBL Geometry Development Time Step Two

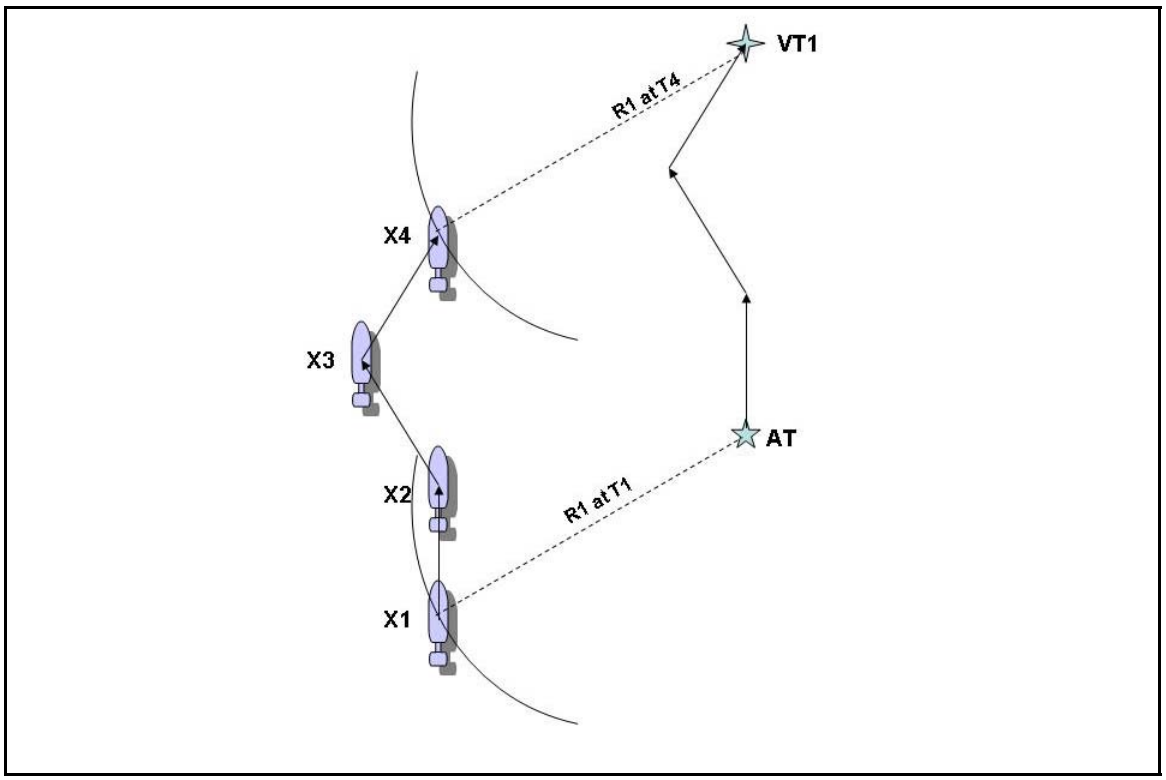

Figure 10: VLBL Geometry Development Time Step One

Once the location of each of the virtual transponders has been determined, the resulting geometry of the virtual long baseline is as shown in Figure 11. The fix can then be computed as shown in Figure 12 using a least squares computation. Again it is important to note that the choice of using four separate measurements to calculate the fix was convenient, but not necessary. A position fix could be generated by time forwarding a single range measurement to cross it with one real-time measurement.

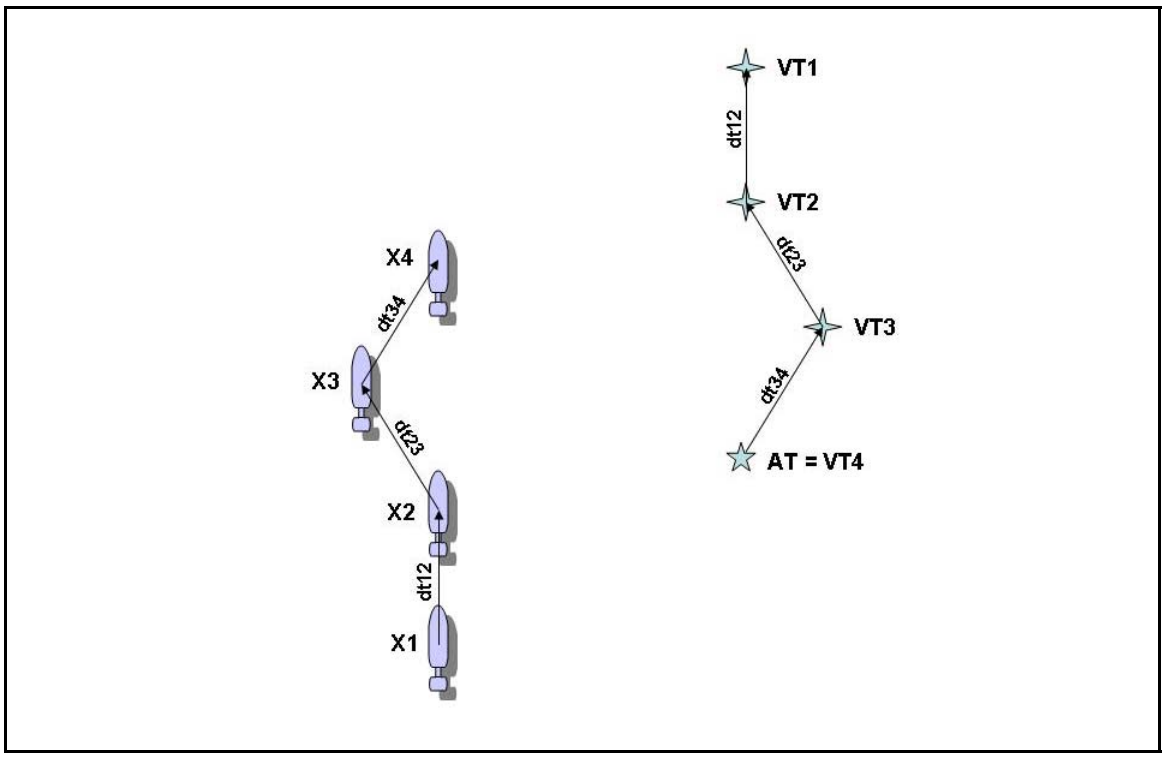

Figure 11: Resulting VLBL Geometry

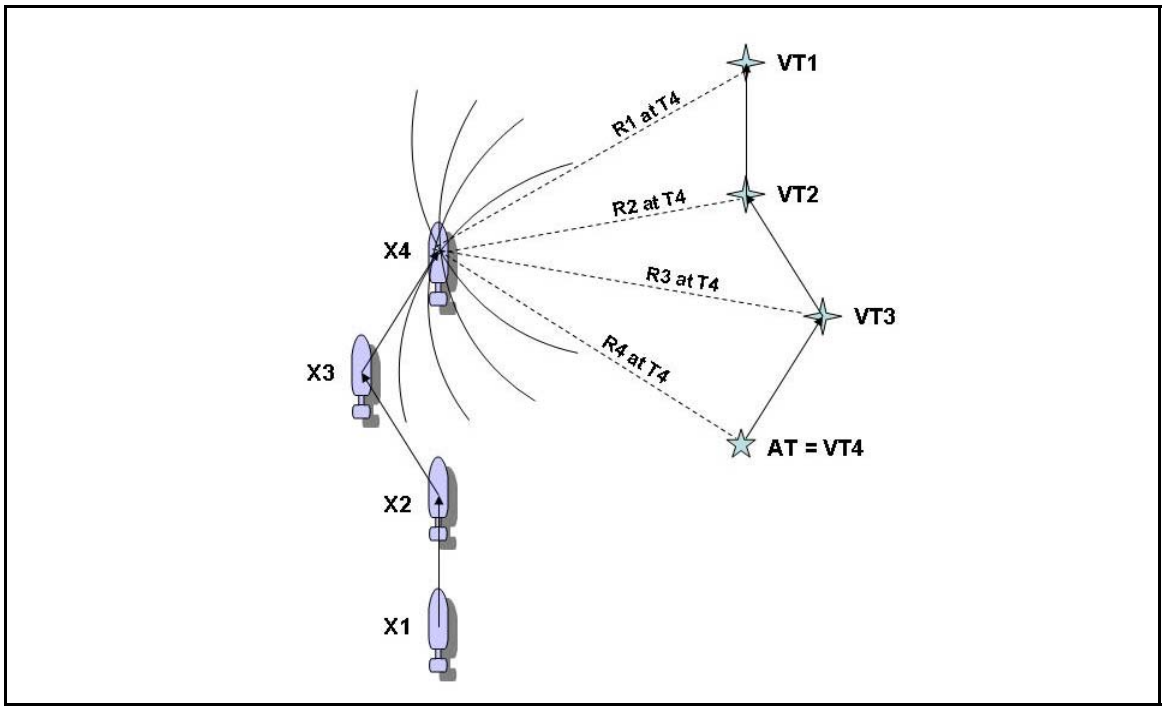

Figure 12: VLBL Fix Computation at Time Step Four

Figure 13 shows a compilation of the complete geometry of the virtual long baseline transponder net.

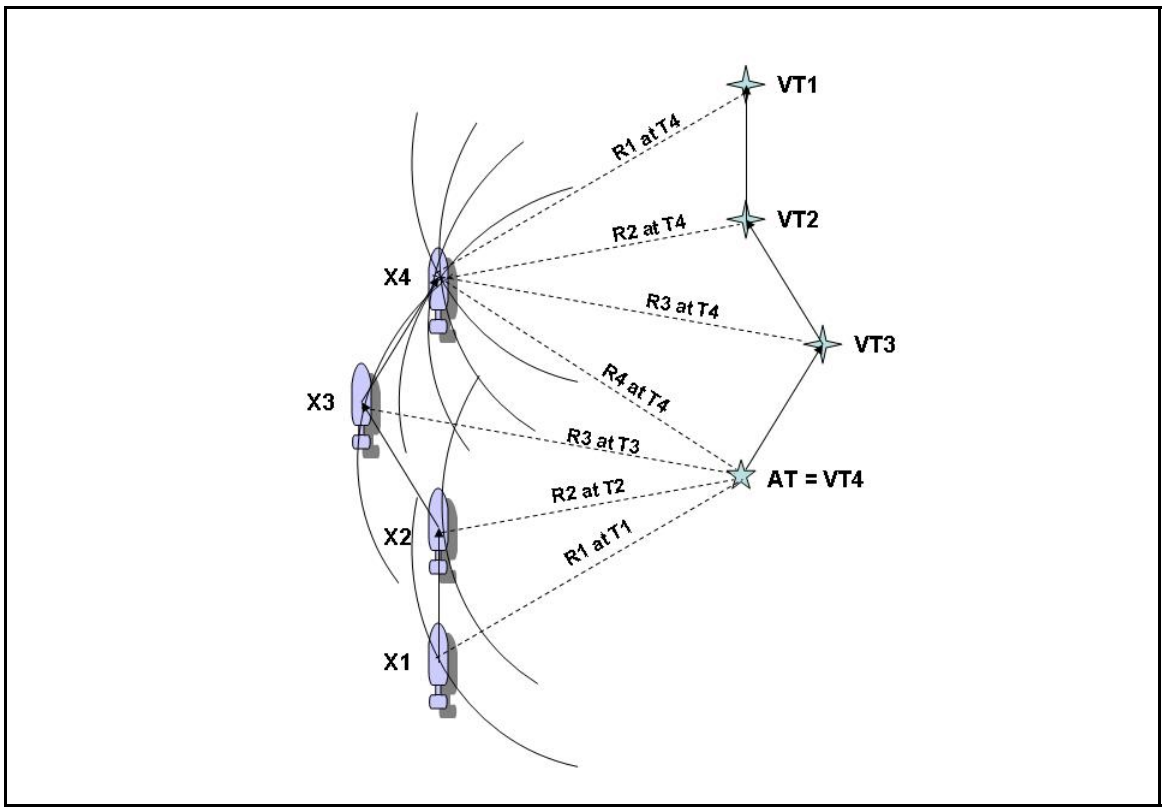

Figure 13: Complete Geometry of the VLBL Transponder Net

Section 3.1.2: Simplifications and Assumptions

The preceding discussion on VLBL transponder net geometry development includes several simplifications. The ranges used as inputs to the algorithm have all been pre-tested via minimum, maximum, and median tests from which only the 'good ranges' were kept. Furthermore, the discussion implies that the resulting geometry produces a fix using ranges taken at four consecutive time steps, which is not always true. Inadequate separation between virtual transponder locations produces a singular matrix with no solution in the least squares computation of position. Therefore, in some instances the historical range data record must be sampled less frequently in order to produce adequate

separation between the virtual transponder locations. The steps of the VLBL algorithm are discussed in Section 3.1.3 below.

The VLBL algorithm assumes that dead reckoning track is a good estimation of vehicle movement. The issue of dead reckoning accuracy is fundamental to all underwater vehicle navigation schemes, but the difference in the VLBL scheme is how the error manifests itself. In the VLBL algorithm, any error between DR track and actual vehicle movement manifests itself as error in the virtual transponder net geometry.

Section 3.1.3: Flow Chart of Approach

The VLBL algorithm is an iterative process using information from multiple time steps to determine each vehicle position fix. The sequence of processes inherent to the VLBL algorithm is laid out visually in Figure 14. The first steps concerning the interrogation of the acoustic transponder, range calculation, and outlier rejection are identical to the processes used in traditional LBL navigation, with the exception that there is only one external acoustic transponder. Furthermore, the VLBL requires the generation of a historical record of ranges from at least four distinct time steps before the first fix can be computed.

Once a record of at least four good ranges has been acquired, the position calculation process begins. The virtual transponder net geometry is determined using the ranges and segments of vehicle dead reckoning track corresponding to the appropriate time steps. Then the ranges and the virtual transponder positions are input into a least squares computation identical to that of a traditional LBL navigation system. If a least squares solution exists, the position fix is accepted and is used to reinitialize the vehicle's dead reckoning track. The cycle is then repeated continuously throughout the dive.

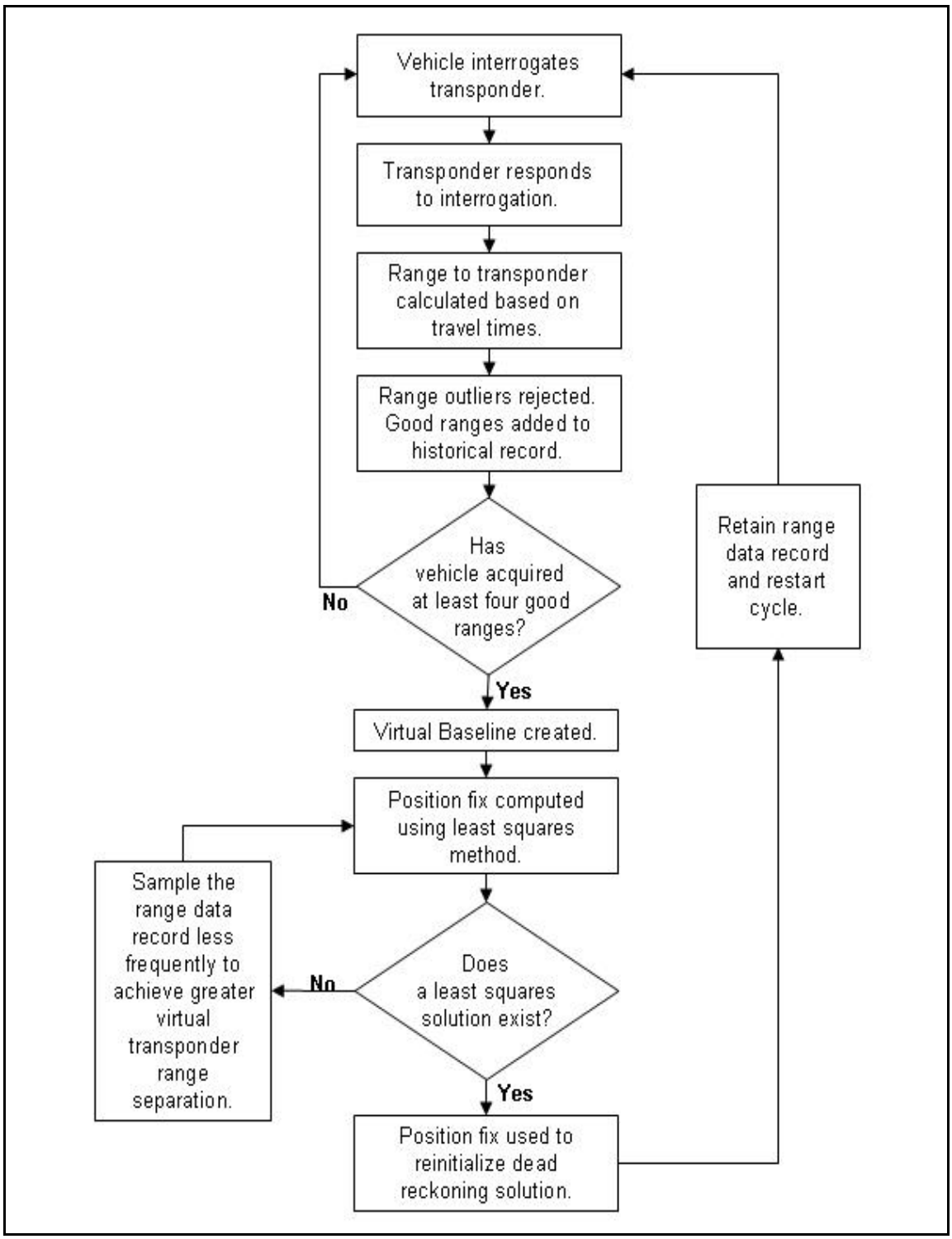

Figure 14: Flow Chart of the VLBL Navigation Algorithm

Section 3.2: Defining the Moving Virtual Long Baseline

The Moving Virtual Long Baseline is an adaptation of the VLBL navigation method to use a ship-mounted single acoustic transponder. In MVLBL navigation, the ship-mounted transponder position is communicated to the vehicle encoded in every acoustic ping, and that information is incorporated into the formation of the virtual transponder net. While the introduction of the VLBL navigation system would decrease underwater vehicle operating costs by requiring the deployment of only one acoustic transponder at each dive site, the successful introduction of MVLBL would provide additional cost savings by eliminating the need to deploy any transponders. Furthermore, MVLBL could increase host platform operational flexibility during voyages.

Section 3.2.1: General MVLBL Geometry

The Moving Virtual Long Baseline geometry is developed in the horizontal plane identically to that of the Virtual Long Baseline with the exception that the movement of the transponder platform needs to be taken into account between successive range calculations. Analogous to the development of the VLBL in Section 3.1.1 above, the following series of figures shows the development of the geometry working backwards from the final time step at which the position fix is calculated.

Although the vehicle movement between time steps is accounted for in the same way for both VLBL and MVLBL, the transponder's movement between the time steps must actually be subtracted from the actual position at the time of fix calculation. The resulting general equation for the development of the MVLBL transponder net is Equation (3.2) as follows:

$$VT_i = AT + \sum_i (v_i * dt_{(i,i+1)}) - \sum_i (V_i * dt_{(i,i+1)}) \qquad (3.2)$$

where

$dt_{(i,i+1)}$ = Time delay between time steps i and $i+1$

V_i = Transponder platform velocity at time step *i*

v_i = Vehicle velocity at time step *i*

AT = Actual transponder location at time of fix calculation

VT_i = Virtual transponder location corresponding to R_i

R_i = Range between vehicle and AT at time step *i*.

In practice, error in the MVLBL transponder geometry can be minimized if the known transponder location at each time step is used in the calculations instead of the transponder dead reckoning track. The general equation in this formulation is shown in Equation (3.3)

$$VT_i = AT_i + \sum_i (v_i * dt_{(i,i+1)}) \tag{3.3}$$

where AT_i = Actual transponder location at time step *i*.

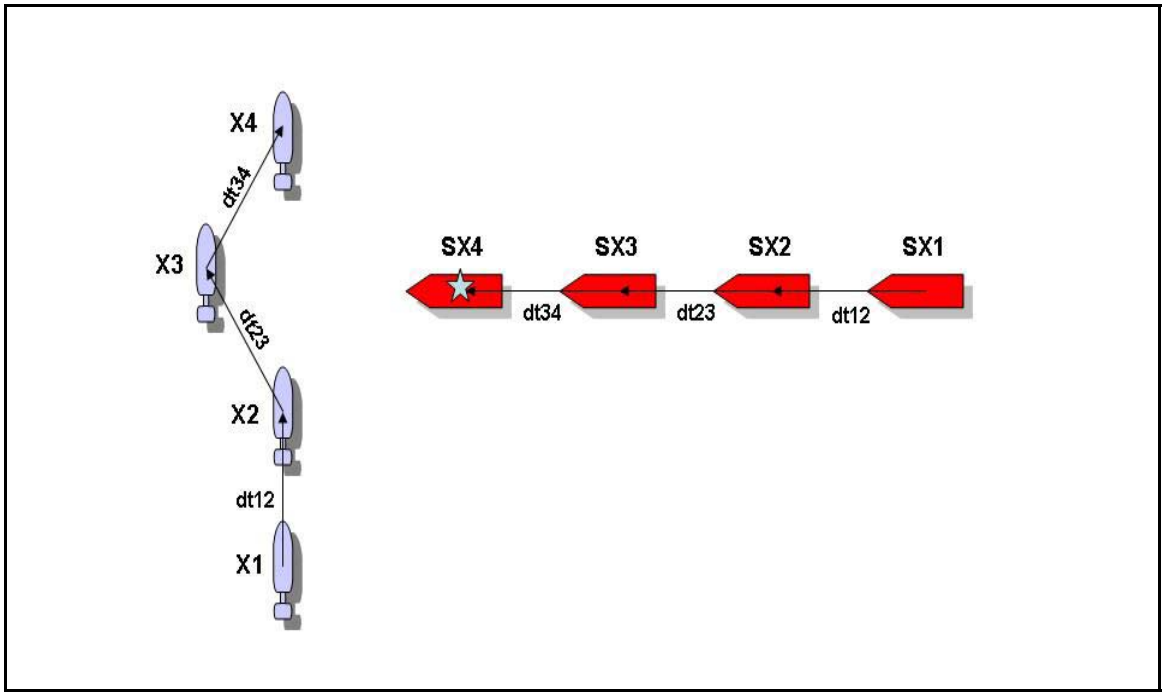

Figure 15: MVLBL Dead Reckoning Track of Vehicle and Transponder Platform

The positions of both the vehicle and the transponder platform at time steps one through four are shown in Figure 15. Although this analysis assumes that the moving transponder is located on the ship from which the underwater vehicle has been launched, the transponder could be located on any moving platform with accurate positioning information. Once again, the vehicle's positions at times T1 through T4 are annotated as X1 through X4, respectively. Additionally, the locations of the ship-mounted transponder at the corresponding times are designated SX1 to SX4.

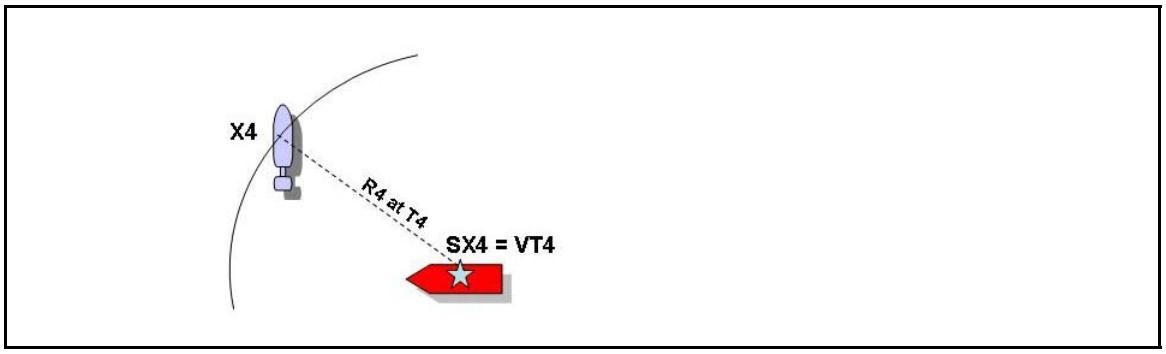

Figure 16: MVLBL Geometry Development Time Step Four

At T4, the range, R4, is calculated between the vehicle, at location X4, and the transponder, at location SX4, as shown in Figure 16. Virtual transponder locations VT3, VT2 and VT1 are calculated using Equation (3.2), as shown in Figure 17, Figure 18, and Figure 19, respectively.

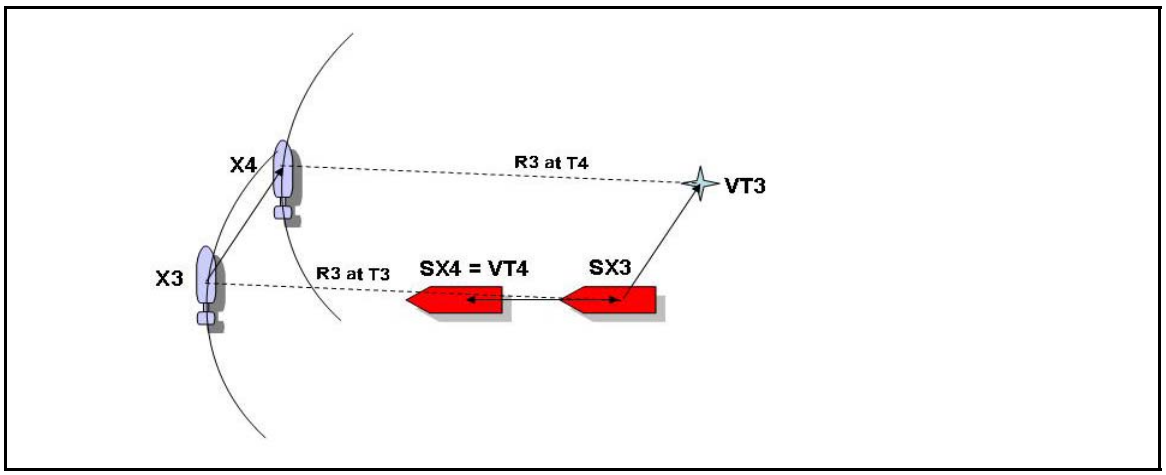

Figure 17: MVLBL Geometry Development Time Step Three

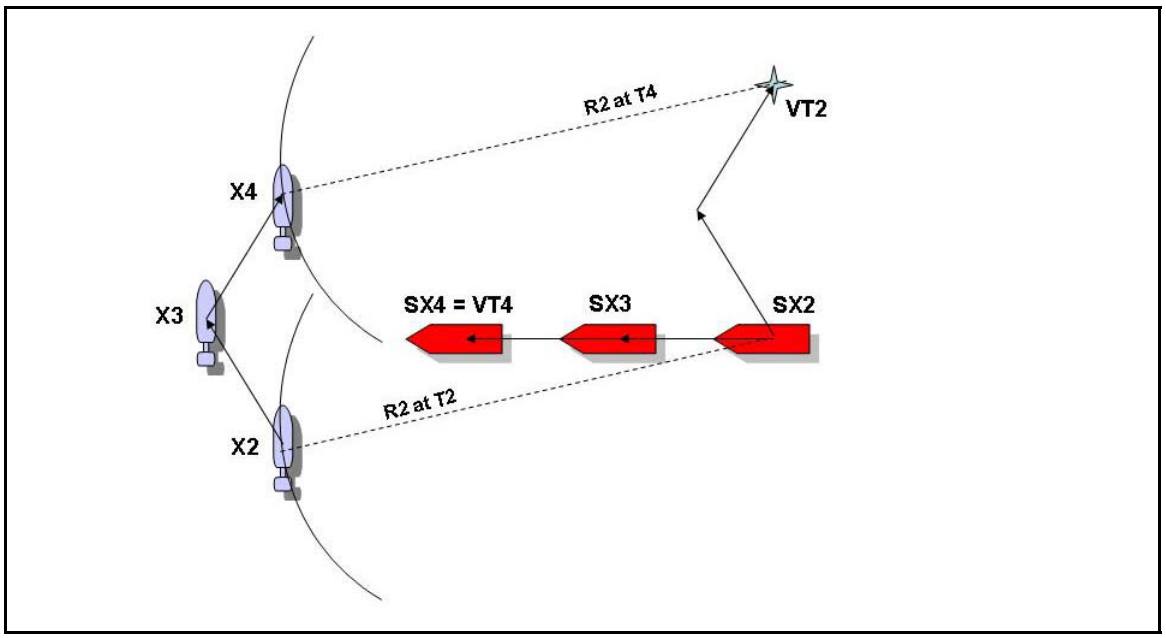

Figure 18: MVLBL Geometry Development Time Step Two

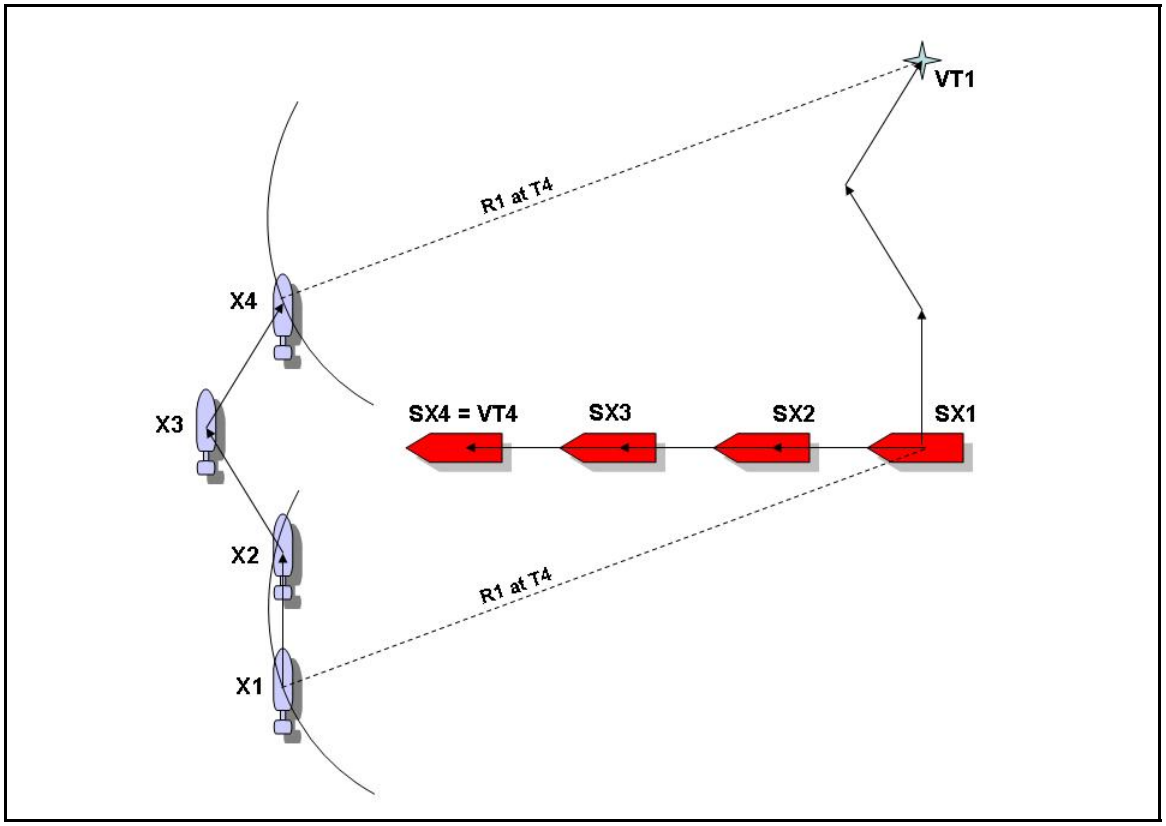

Figure 19: MVLBL Geometry Development Time Step One

Finally, the resulting virtual transponder net geometry is shown in Figure 20 and the computation of the position fix at T4 is shown in Figure 21.

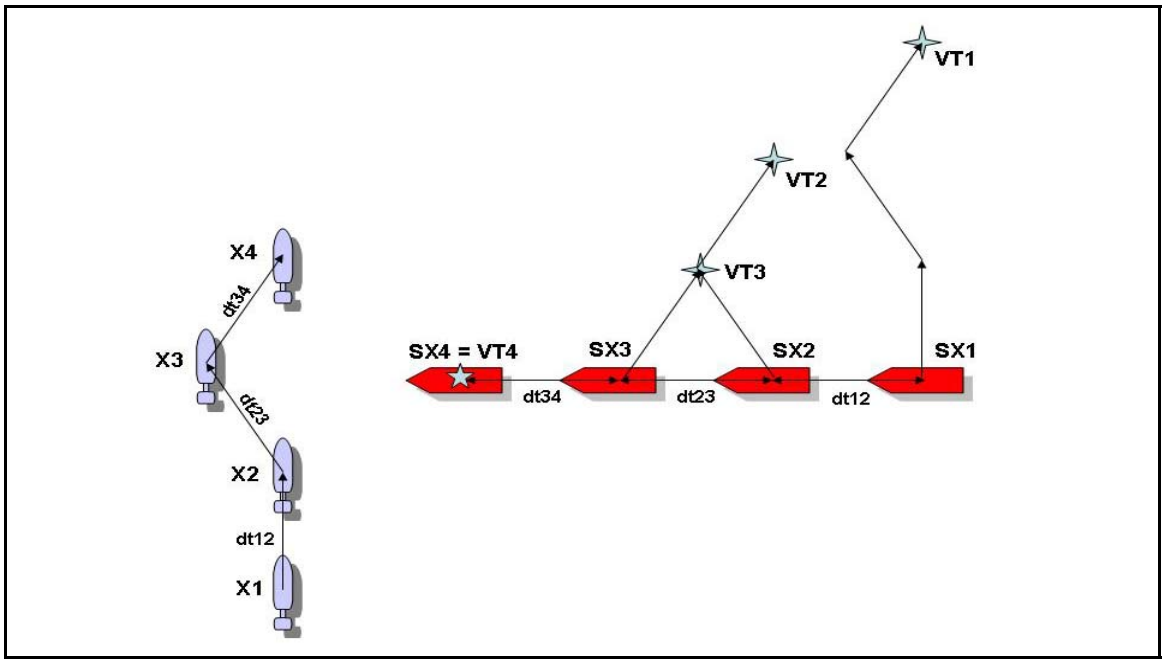

Figure 20: Resulting MVLBL Geometry

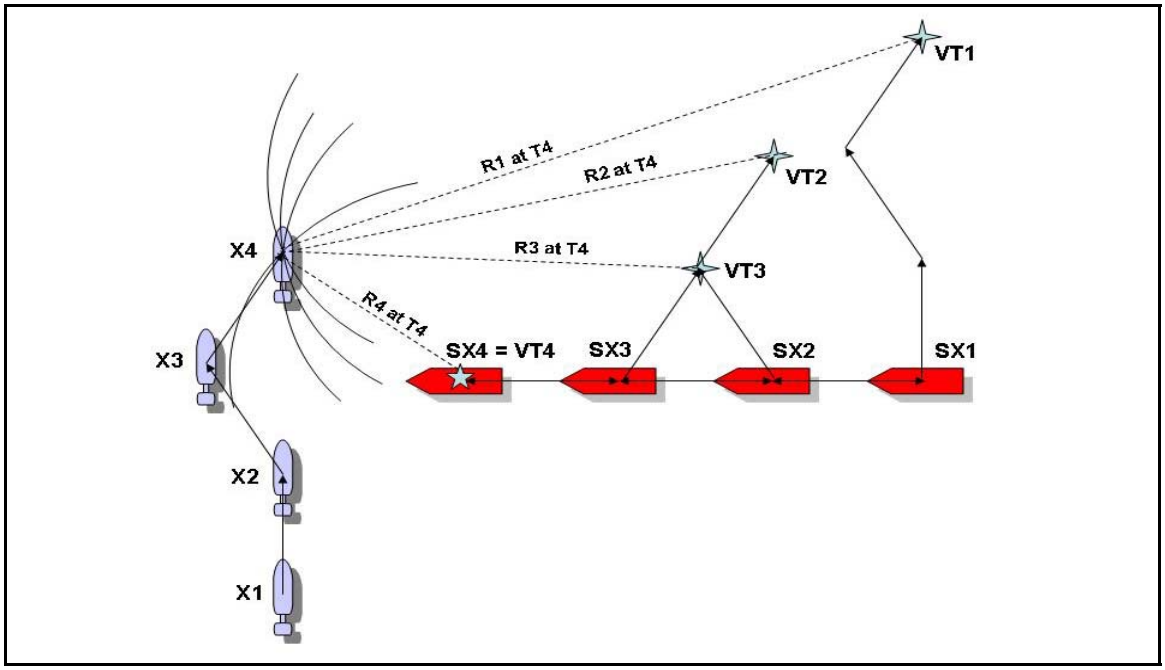

Figure 21: MVLBL Fix Computation at Time Step Four

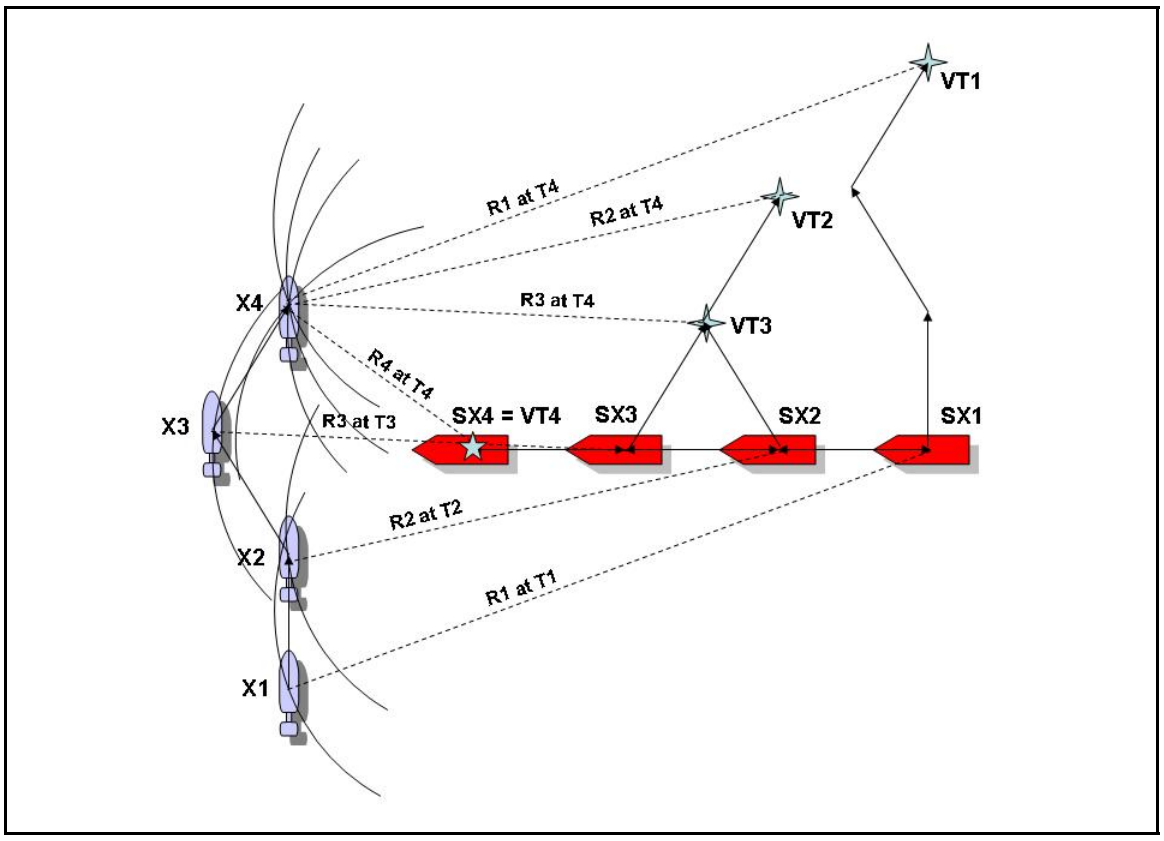

Figure 22: Complete Geometry of the MVLBL Transponder Net

Section 3.2.2: Simplifications and Assumptions

All of the simplifications and assumptions discussed above in Section 3.2.2 still apply for the discussion on MVLBL geometry with the addition of a few more considerations. A critical assumption for the MVLBL navigation scheme is that the transponder platform would have to relay position or DR track information to the vehicle at each time step. Other than gathering this additional information, MVLBL follows the same flow chart as VLBL as seen in Figure 14.

Furthermore, depending on the relative motion between the vehicle and the ship-mounted transponder, the MVLBL algorithm can either increase or decrease the separation between the locations of the virtual transponders as compared to VLBL. This separation change can affect the observability of the vehicle position, i.e. whether or not there is a solution to the least squares computation of position.

Chapter 4: Virtual Long Baseline Navigation Results

Section 4.1: Virtual Long Baseline Algorithm Performance Characteristics

Using the methods described in the previous chapter, the VLBL navigation algorithm was developed in two different forms. The first form was a basic VLBL algorithm that did not include dead reckoning or outlier rejection, included in Appendix B in MATLAB script form. The second form is an expanded VLBL algorithm including both dead reckoning and outlier rejection. This expanded algorithm is included in Appendix C in MATLAB script form. The purpose of the basic VLBL algorithm was to examine the performance characteristics of the VLBL method by varying the inputs to the algorithm using a simulated data set with no noise. The inputs to both VLBL algorithms include beacon location, sampling rate of range data, and acceptable least squares residual, while the expanded VLBL algorithm also requires an outlier rejection factor.

Section 4.1.1: Simulated Data Set and Geometry

A simulated data set was developed in order to test some basic characteristics of the VLBL method. The geometry of the simulated data set emulates that of a typical deep ocean survey pattern, as shown in Figure 23. Since the purpose of the initial simulations was to illustrate basic performance characteristics of the VLBL system, no noise was included in the simulated data set. Similar to a traditional LBL system, four beacons were included the simulated data set in the configuration shown in Figure 23. Range data from only one of these transponders is used in each trial of the VLBL navigation algorithm, so beacon choice is an input parameter to the system.

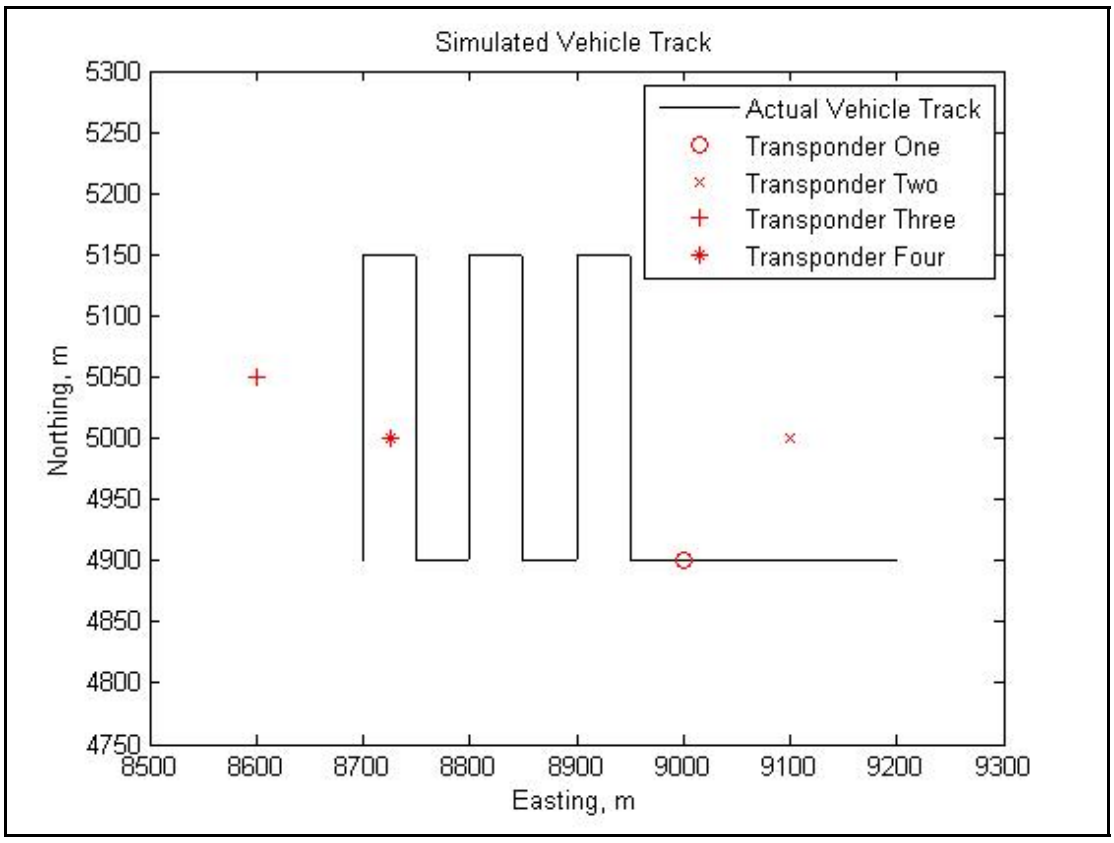

Figure 23: Simulated Dive Track and Transponder Locations

Section 4.1.2: Effect of Transponder Location on Observability

The effect of beacon location on system observability was analyzed using the simulated data set with the basic VLBL algorithm. All other input parameters to the system were held constant while each transponder was chosen in turn. The results are shown in the following series of figures. In the first scenario, shown in Figure 24, the vehicle track was completely observable with the exception of at the corners of the survey pattern. Immediately following each course change, the vehicle lost the actual track and fixed its position elsewhere on the range arc originating from the transponder. In the complete VLBL algorithm, some of these outliers would be rejected, but they are retained here for illustration purposes.

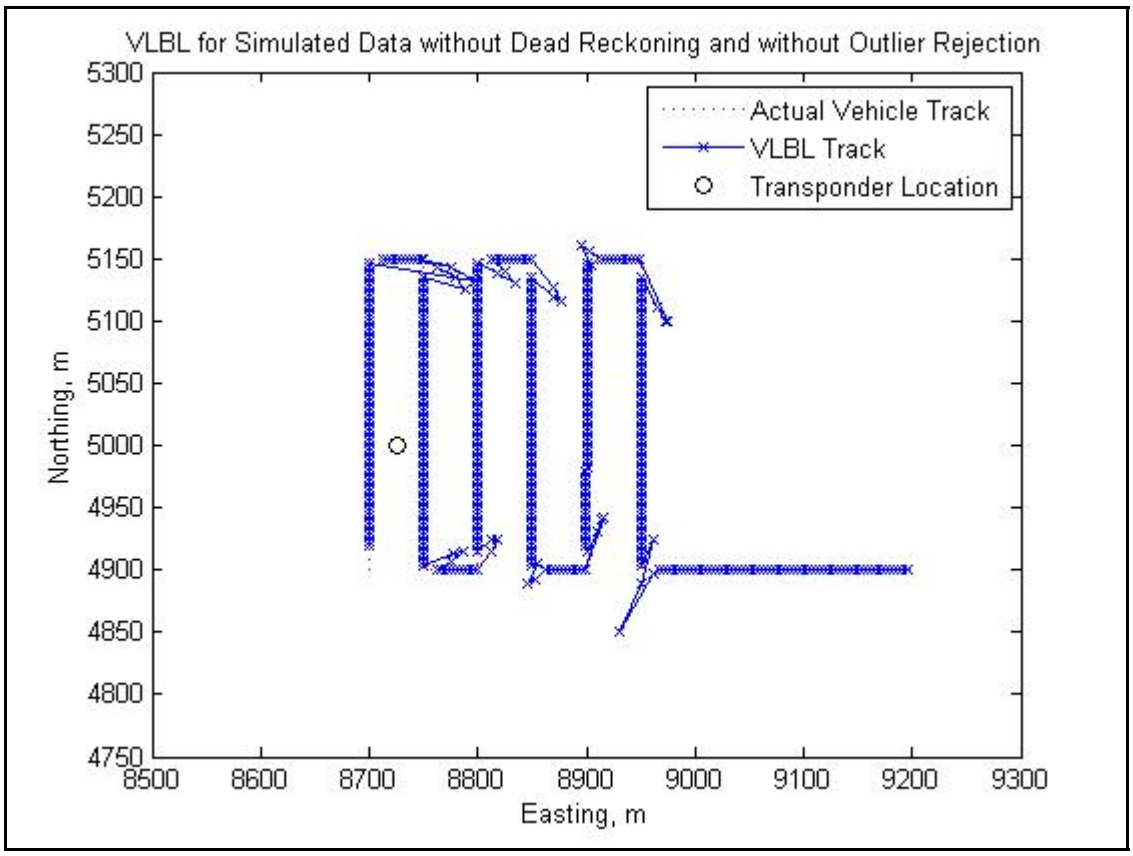

Figure 24: Basic VLBL System using a Sampling Rate of 1 in 4 Ranges with Transponder Four

The next scenario, illustrated in Figure 25, exhibited similar observability characteristics of the VLBL system using ranges from Transponder Three. However, this scenario highlighted one other observability issue in the form of mirror pathline tracking. The vehicle was able to correctly fix its position on all straight pathlines with the exception of the fifth eastbound leg of the survey. On this leg, the vehicle fixed its position on a mirror pathline south of the transponder. This is consistent with the issue of local versus global observability raised by both Gadre et al. and Ross et al. [27, 28] Using single source range information, vehicle position is locally observable, but not globally observable. In other words, there are two solutions to the triangulation problem and the vehicle cannot distinguish between the two without additional information. However, with proper initialization, the vehicle can distinguish between the two solutions

and fix its position on the actual track instead of the mirror path. However, in this illustration using the basic VLBL algorithm, dead reckoning was not used to provide system initialization in the form of updated position estimates. Furthermore, outlier rejection was not included in the process, therefore the vehicle had no way of distinguishing between the actual and mirror tracklines. After the vehicle changed course again, the position fixes reacquired the actual trackline.

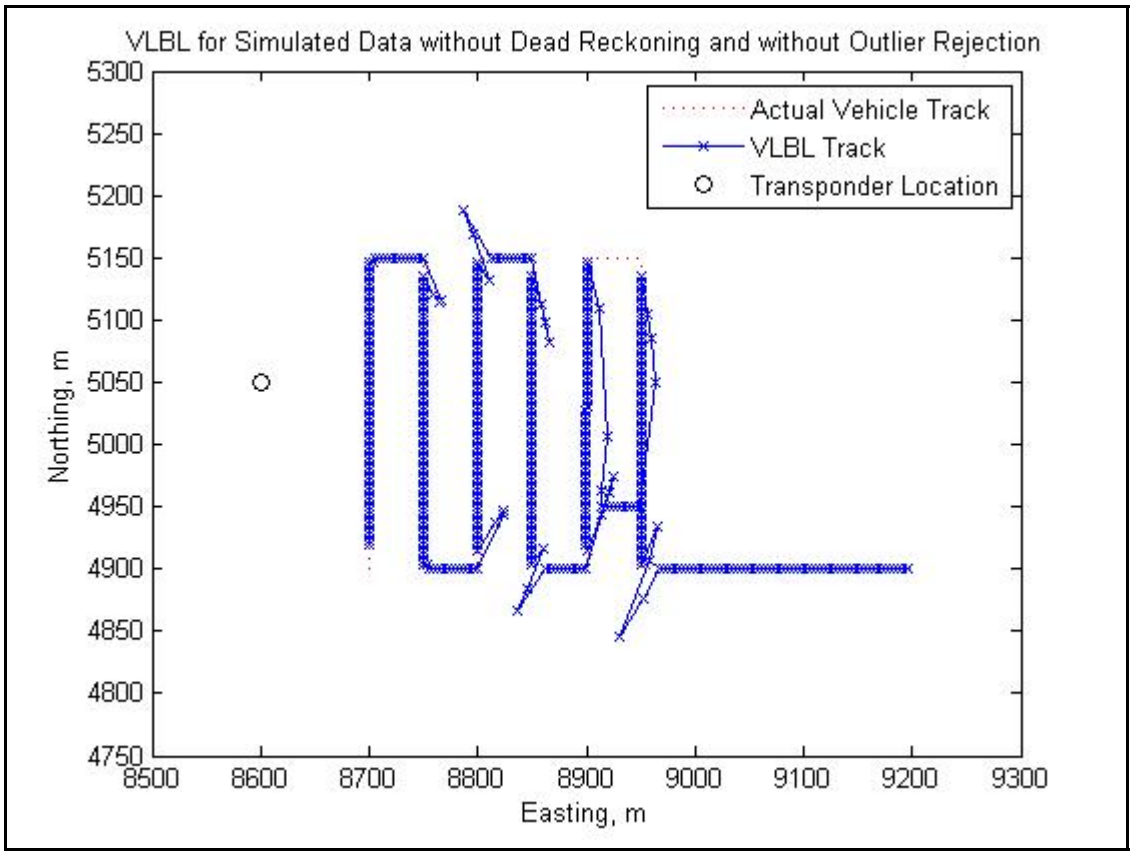

Figure 25: Basic VLBL System using a Sampling Rate of 1 in 4 Ranges with Transponder Three

The results of the basic VLBL algorithm using Transponder Two are shown in Figure 26. In this scenario, the system behaved similarly to the Transponder Four scenario discussed above in that the instances of system unobservability occurred immediately following vehicle course changes. Although the divergences between the position estimates and the actual track were larger in the Transponder Two scenario, they still

followed the same pattern of forming a sweeping arc around the actual position of the transponder.

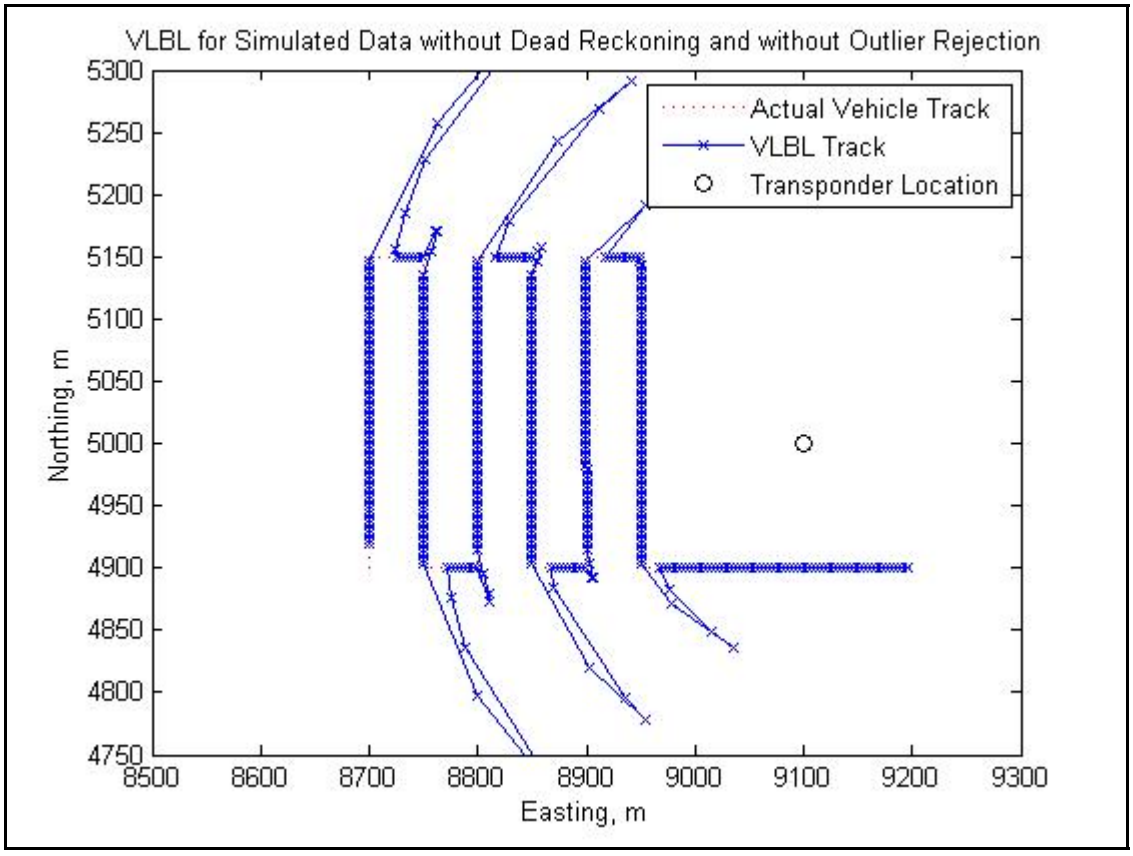

Figure 26: Basic VLBL System using a Sampling Rate of 1 in 4 Ranges with Transponder Two

The final scenario, using Transponder One, added an additional element to the analysis of observability because Transponder One was located directly on one of the tracklines. As shown in Figure 27, this system exhibited dramatic divergence between the vehicle estimated position and the actual track immediately after turns, similar to the other scenarios. Furthermore, on all three eastbound tracklines at the southern end of the survey pattern, the system exhibited more erratic position estimates. These errors were due to the unobservability of the system when the vehicle was traveling directly towards or away from the transponder, as predicted by the work of Gadre et al. and Ross et al. [26, 28]

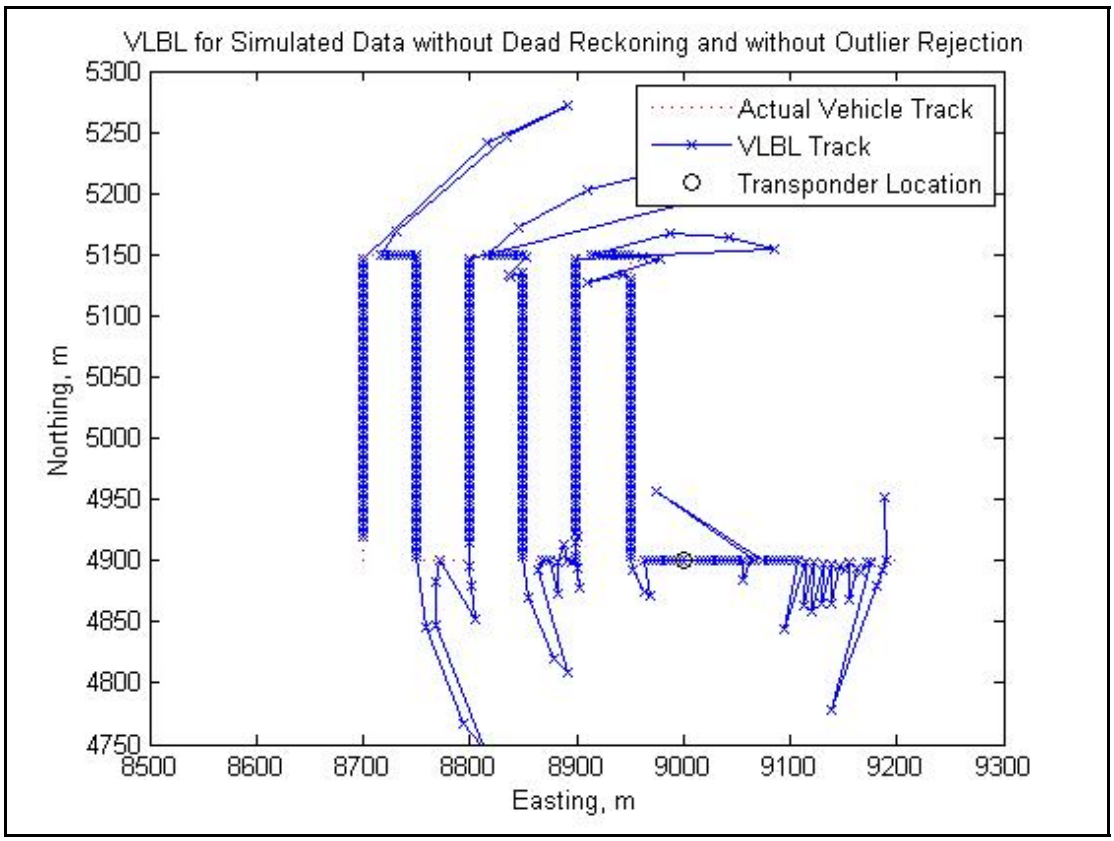

Figure 27: Basic VLBL System using a Sampling Rate of 1 in 4 Ranges with Transponder One

Therefore, it was apparent from using the simulated data in the basic VLBL algorithm that beacon placement with regard to vehicle survey path is an important input to an effective VLBL system.

Section 4.1.3: Effect of Range Sampling Rate on Observability

Another input parameter to the VLBL navigation system is the range sampling rate. Range data to the transponder is recorded by the vehicle at every navigation cycle. If the observed range passes minimum, maximum and median tests, then it is added to the historical record of range data. In some instances, the distance the vehicle travels between recording consecutive range data points is negligible compared to its range to the transponder. In these cases, the mathematical computation of a least squares solution

is not possible because the matrix generated in the calculations is not invertible. Therefore, instead of using four consecutive ranges in the formation of the VLBL transponder net, the historical range data can be sampled less frequently. For example, a sampling rate of 1 in 10 ranges means that every tenth range data point is used in the formation of the VLBL net. The advantage of using a less frequent sampling rate is to achieve geometric separation between the virtual transponders, therefore improving the 'health' of the matrices. However, the choice of sampling rate directly and significantly affects the update rate of the VLBL system. A sampling rate of 1 in 10 means that the opportunity to develop each position estimate requires ten times as many navigation cycles, but there is a greater likelihood that the system will be observable and a position estimate will be calculated. Therefore, there is an important, nonlinear relationship between the sampling rate of range data and VLBL navigation system update rate.

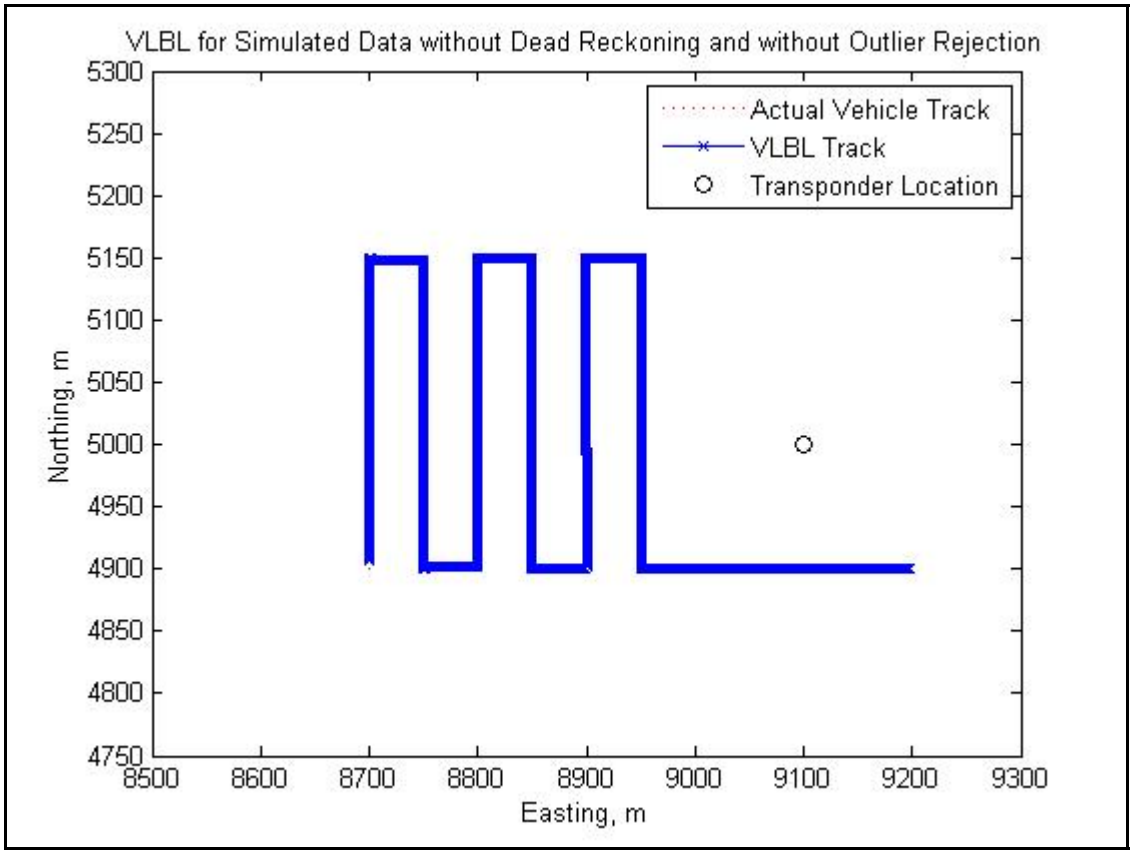

Figure 28: Basic VLBL System using Transponder Two with a Sampling Rate of 1 in 1 Ranges

The simulated data set was again used with the basic VLBL algorithm to examine the effects of sampling rate on system performance. All other input parameters were held constant while the sampling rate was varied. Since there is no noise in the simulated data set and the transponder is located close to the vehicle track, it is important to note that these results do not adequately illustrate the potential unobservability which results from too frequent data sampling. That issue will be discussed in more detail in the context of real-world data VLBL processing. However, this series of figures does show the potential disadvantage of sampling the range data too infrequently. Figure 28 shows the basic VLBL system sampled at a rate of 1 to 1. As previously discussed, the model setup does not reflect the true potential for position estimate unobservabilities. Therefore, the VLBL fixes are exceeding accurate with respect to actual vehicle track.

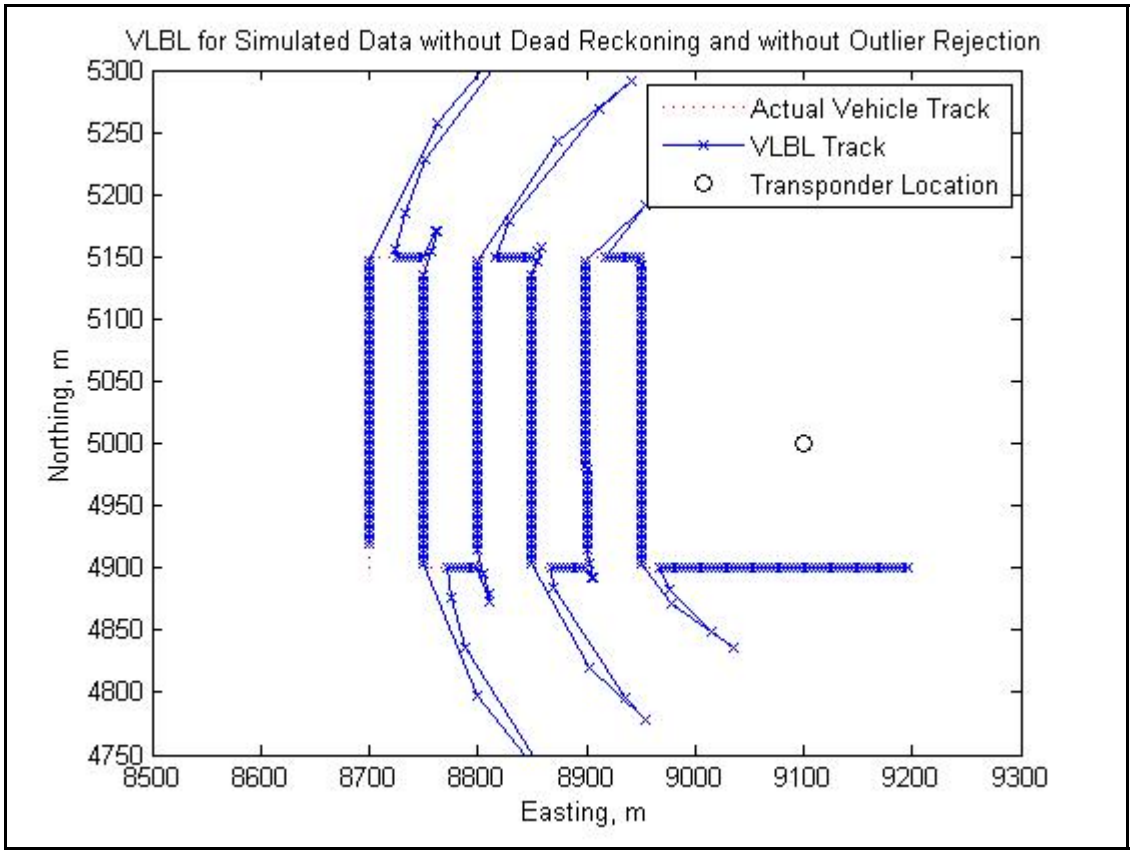

Figure 29: Basic VLBL System using Transponder Two with a Sampling Rate of 1 in 4 Ranges

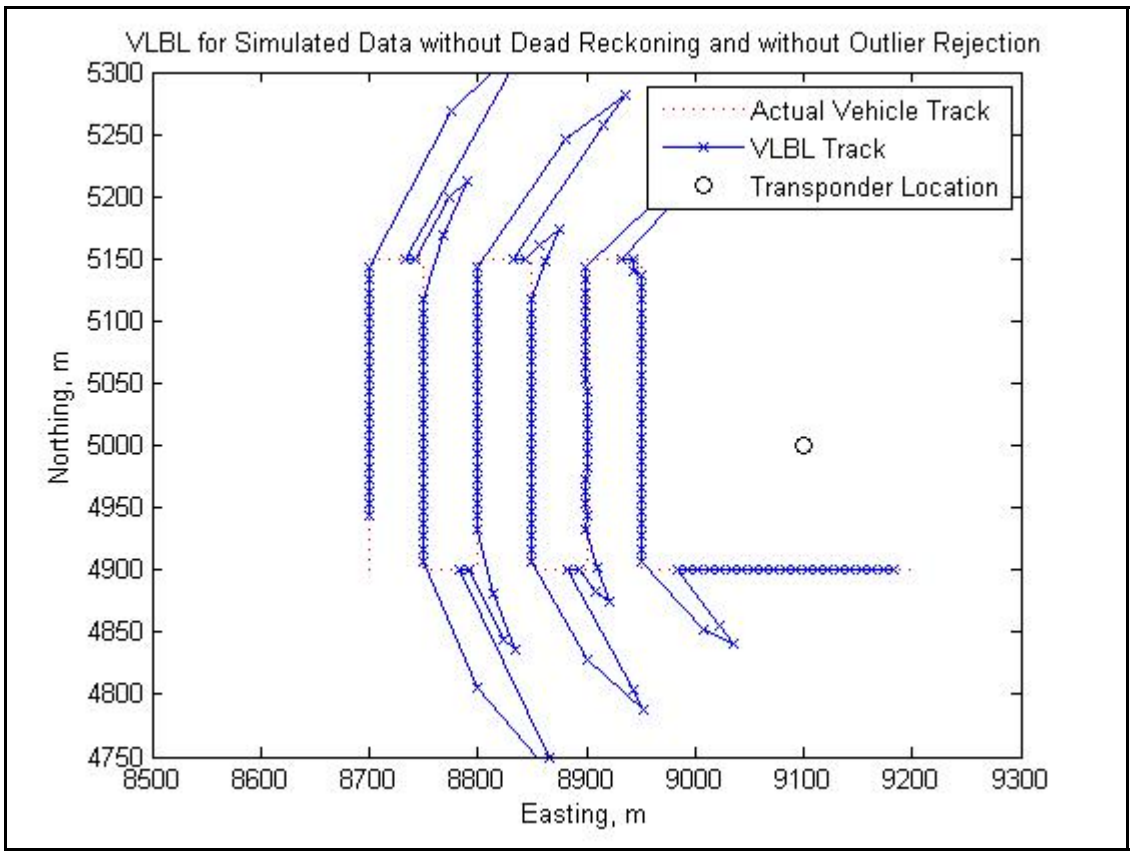

Figure 30: Basic VLBL System using Transponder Two with a Sampling Rate of 1 in 10 Ranges

Figure 29, Figure 30, and Figure 31 illustrate the effects of reducing the sampling rate to 1 in 4, 1 in 10, and 1 in 25, respectively. The vehicle position estimates began to diverge dramatically from the actual vehicle track immediately following course changes. Since the basic algorithm includes neither dead reckoning nor outlier rejection, these outliers were retained for illustration purposes. As the sampling rate became less frequent, it took the vehicle longer to reacquire the track following each turn. The sampling rate in the final scenario, 1 in 25, is so low that the vehicle never reacquired the actual track on the shorter transects of the survey pattern, as illustrated in Figure 31. Furthermore, whereas it had taken the vehicle only four meters, corresponding to four navigation cycles, in the first scenario with a 1 in 1 sampling rate, it took the vehicle 100 meters to acquire its first position fix with a 1 in 25 sampling rate. Therefore, it has been

shown that sampling rate has a major effect on VLBL system performance as well. The effects of sampling rate will be discussed further with respect to real-world data later on as well.

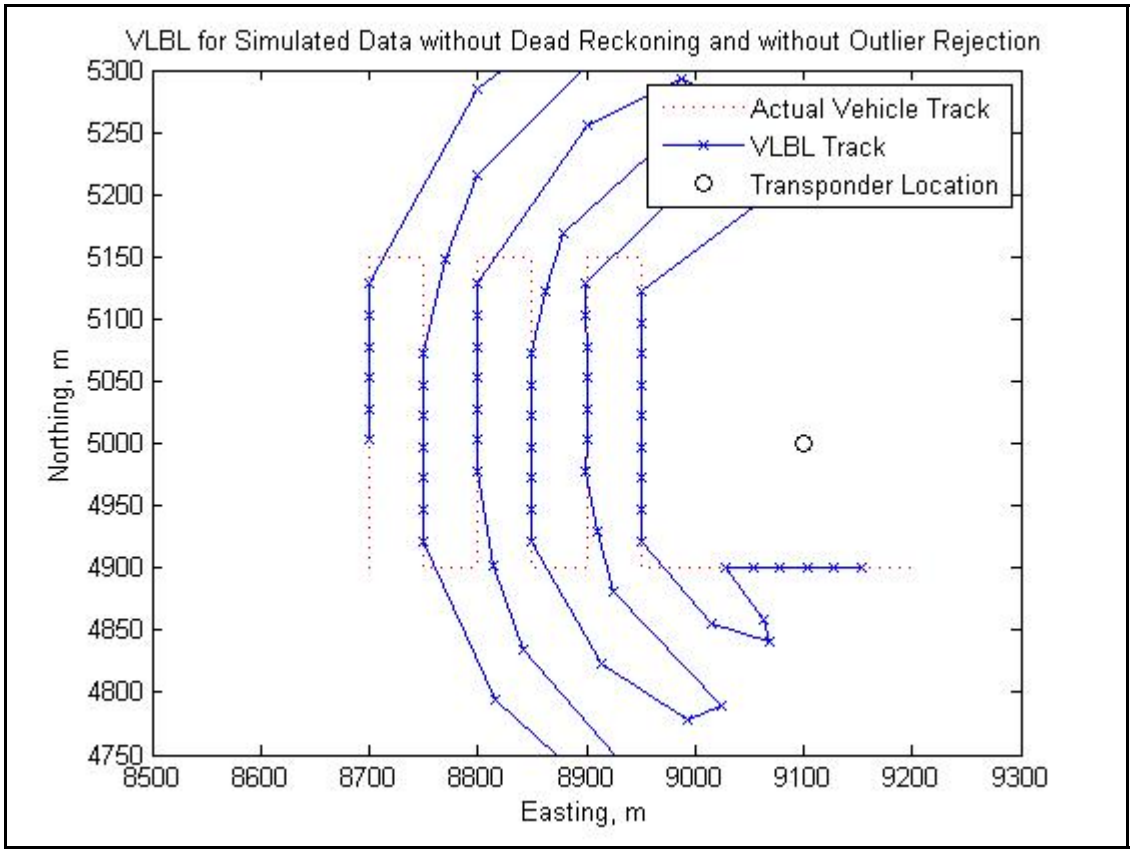

Figure 31: Basic VLBL System using Transponder Two with a Sampling Rate of 1 in 25 Ranges

Section 4.2: Virtual Long Baseline Performance using Real-World Data

Data from deployments by the underwater vehicle called the Autonomous Benthic Explorer (ABE) were used to demonstrate the performance of the VLBL algorithm developed in this thesis.

Section 4.2.1: The Autonomous Benthic Explorer (ABE)

The ABE is a deep ocean autonomous underwater vehicle developed at the Deep Submergence Lab (DSL) of the Woods Hole Oceanographic Institution (WHOI). One of ABE's primary missions has been to conduct deep ocean bottom survey missions to find hydrothermal vents in support of scientific research objectives. These survey missions are often carried out progressively along midocean ridges in a number of adjacent dive sites.

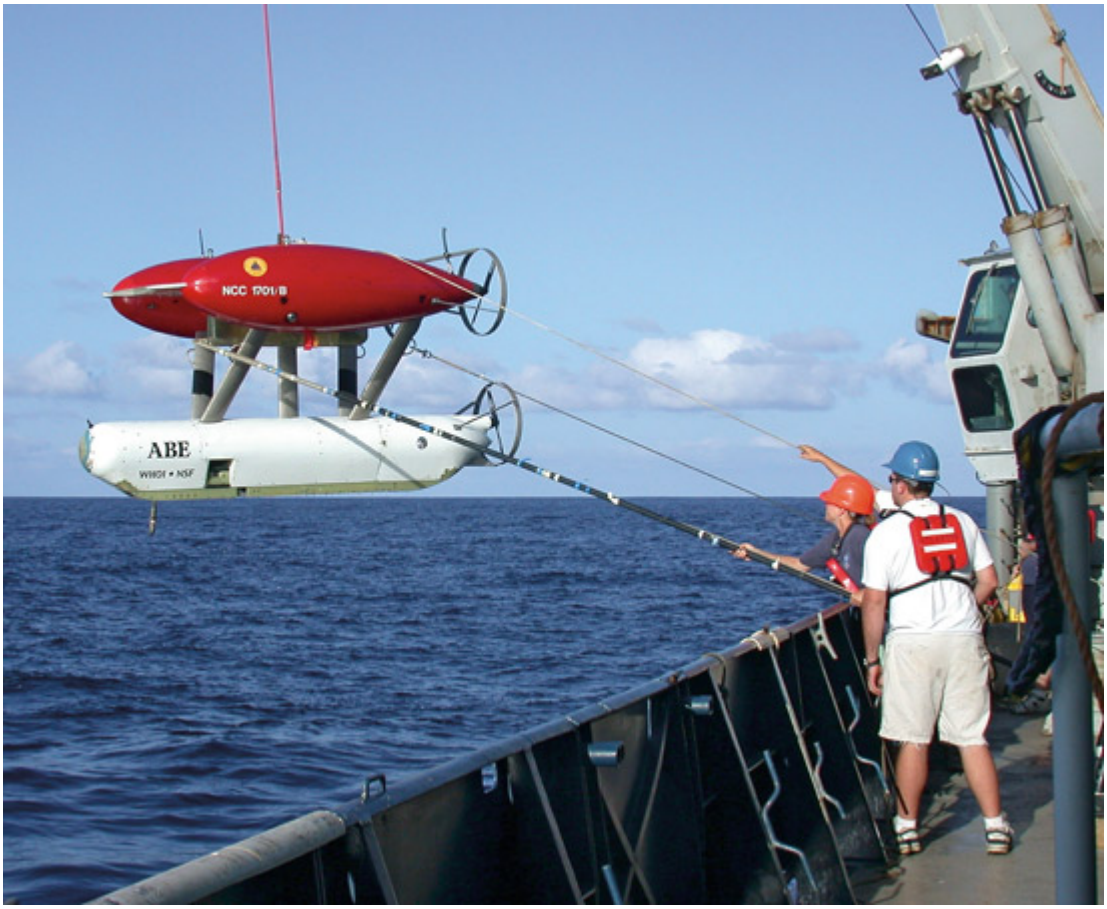

Figure 32: The Autonomous Benthic Explorer. (Dana Yoerger) [29]

Data from two ABE dives were used with the expanded VLBL algorithm in order to assess the performance of the VLBL navigation method with real-world data. These dives, ABE162 and ABE163, were both bottom survey operations done in the Juan de Fuca region in September 2005. Traditional LBL navigation systems were used for each

dive with four external acoustic transponders. During each trial of the VLBL algorithm, one transponder was chosen from which to use the range data. The VLBL algorithm performance was judged against the best real-time estimates of vehicle track made by ABE using the traditional combined LBL and DVL navigation system. The figures in the following sections label the best real-time track as 'Actual Vehicle Track,' but this terminology is actually a misnomer because post-processing of navigation data using Kalman Smoothing techniques is able to correct the best real-time track estimate. However, since the goal of VLBL is to replicate the real-time results of traditional VLBL, in this thesis, the term actual track refers to the real-time track estimate.

In addition to the range data to each of the transponders, readings from ABE's heading, depth, and Doppler Velocity Log (DVL) sensors were used as inputs to the VLBL algorithm. Each sensor and instrument onboard ABE has particular issues associated with it which affect the quality of its output. Ample literature exists discussing these issues in detail. However, since the goal of this thesis is to compare the performance of the VLBL navigation system to the traditional LBL system using the same input data, the accuracy of ABE's instrumentation is not explicitly addressed herein. [30-39]

Section 4.2.2: Effect of Sampling Rate on Virtual Long Baseline Navigation Performance using Real-World Data

The issue of the effect of sampling rate on VLBL system performance was revisited using real-world data from dive ABE162. The following series of figures shows the effect of decreasing the range data sampling rate. With sampling rates of 1 in 1 ranges and 1 in 4 ranges respectively, Figure 33 and Figure 34 illustrate that a decrease in sampling rate actually produced better track following on several tracklines of the survey, holding all other input parameters constant. However, Figure 35 and Figure 36 show that VLBL performance was degraded with a further decrease of sampling rate to 1 in 10 and 1 in 25 ranges, respectively.

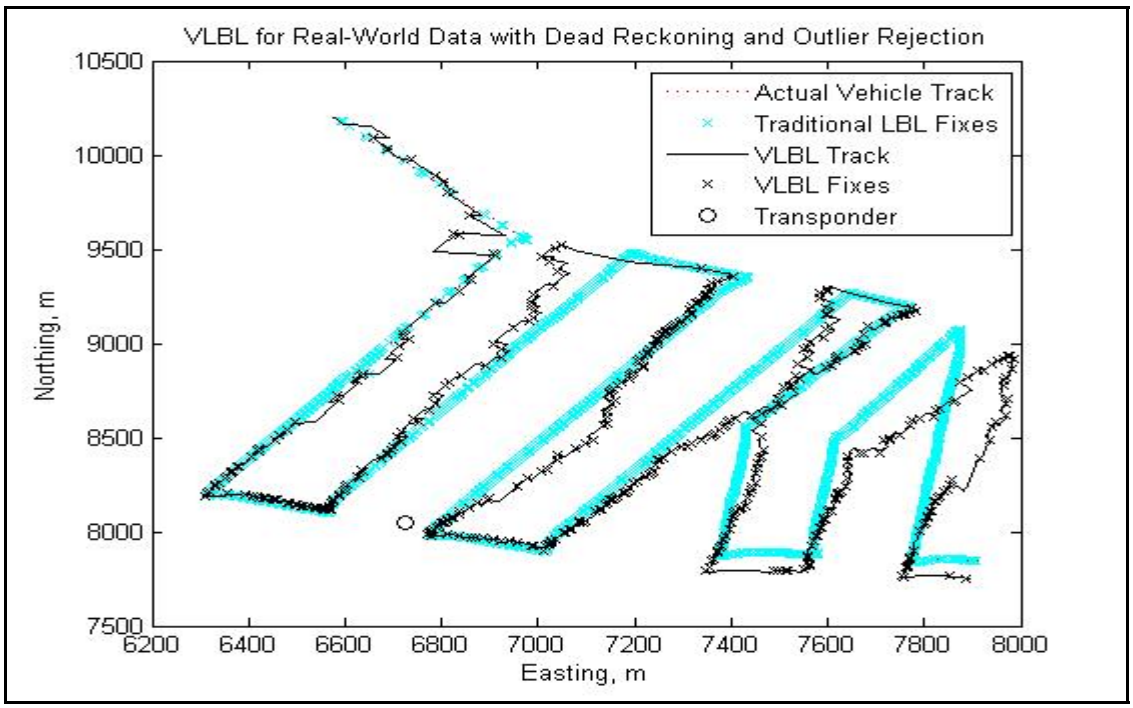

Figure 33: Expanded VLBL Algorithm with ABE162 using Transponder Two and an Outlier Rejection Factor of 1.8 with a Sampling Rate of 1 in 1 Ranges

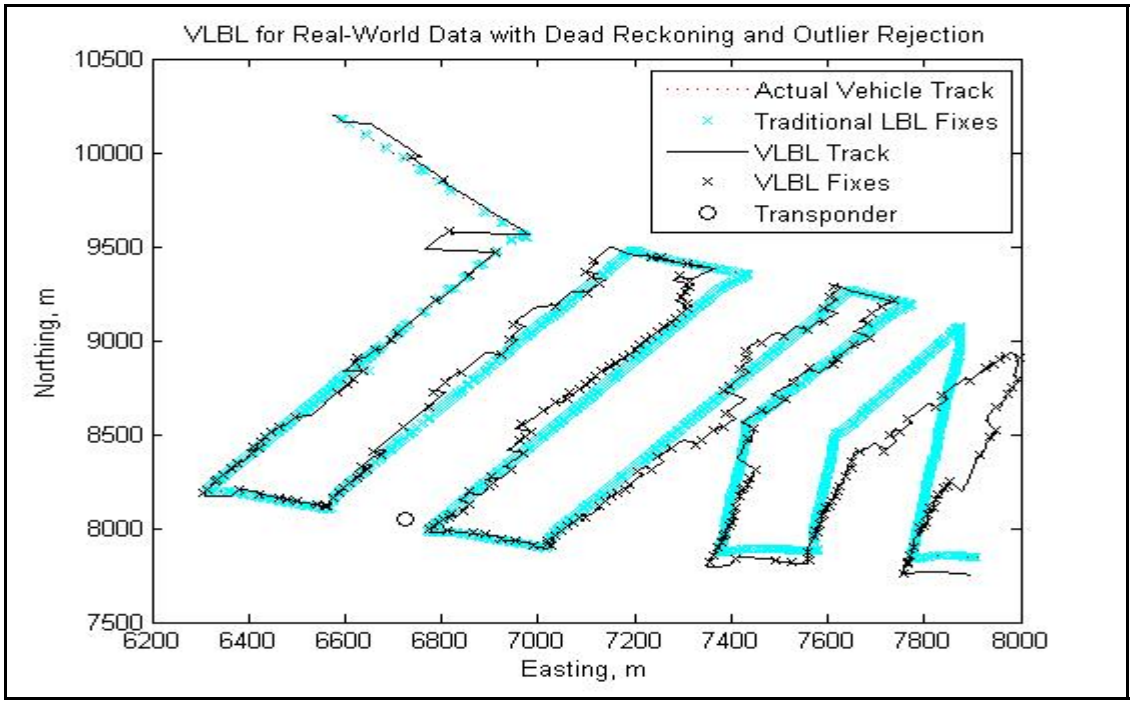

Figure 34: Expanded VLBL Algorithm with ABE162 using Transponder Two and an Outlier Rejection Factor of 1.8 with a Sampling Rate of 1 in 4 Ranges

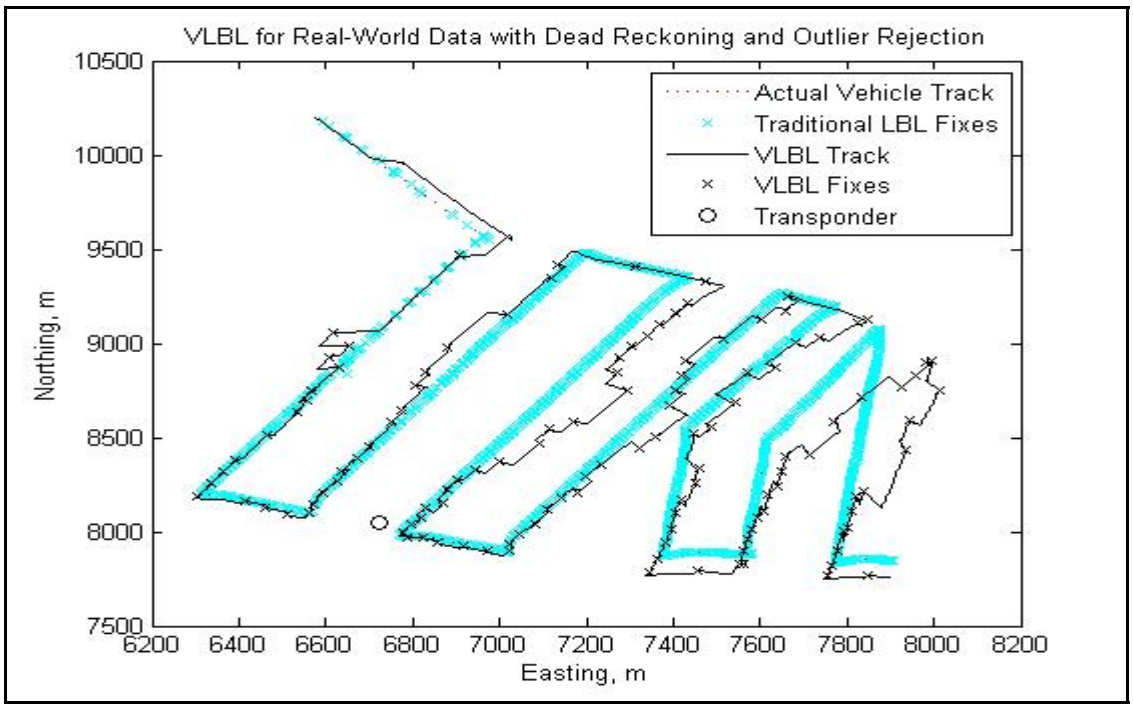

Figure 35: Expanded VLBL Algorithm with ABE162 using Transponder Two and an Outlier Rejection Factor of 1.8 with a Sampling Rate of 1 in 10 Ranges

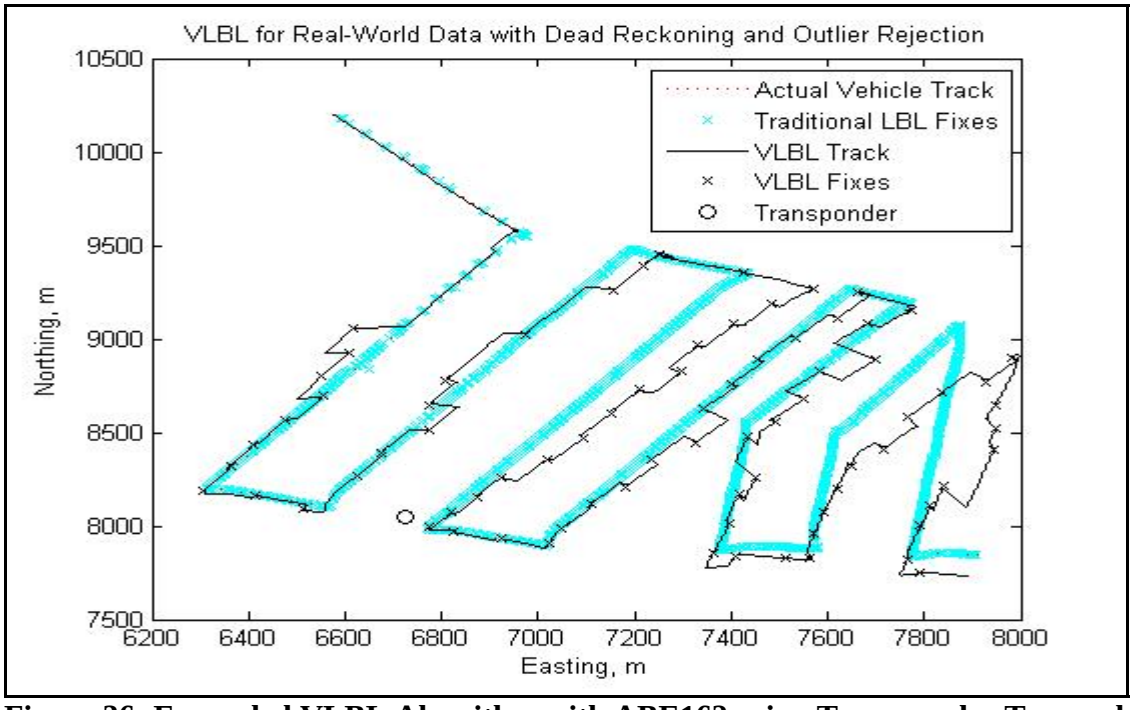

Figure 36: Expanded VLBL Algorithm with ABE162 using Transponder Two and an Outlier Rejection Factor of 1.8 with a Sampling Rate of 1 in 25 Ranges

Section 4.2.3: Effect of Outlier Rejection on Virtual Long Baseline Navigation Performance using Real-World Data

The effect of outlier rejection on the performance of VLBL was analyzed using ABE162 in the expanded VLBL algorithm. The outlier rejection factor is used to eliminate any position estimates that fall outside of a circle emanating from the last fixed position. The radius of that circle is the expected travel distance of the vehicle based on DVL speed times the outlier rejection factor. Therefore, an outlier rejection factor of 1.5 corresponds to the elimination of any position estimates that are more than 150% of the estimated travel distance away from the last fixed position. Figure 37 through Figure 40 show the progressive increase in outlier rejection factor from 1.1 to 2.5. If the factor is set too low, such as in the examples with 1.1 and 1.5, too few position estimates are accepted and the vehicle position estimates begin to diverge from the actual track due to compass heading error over long periods of dead-reckoning track keeping.

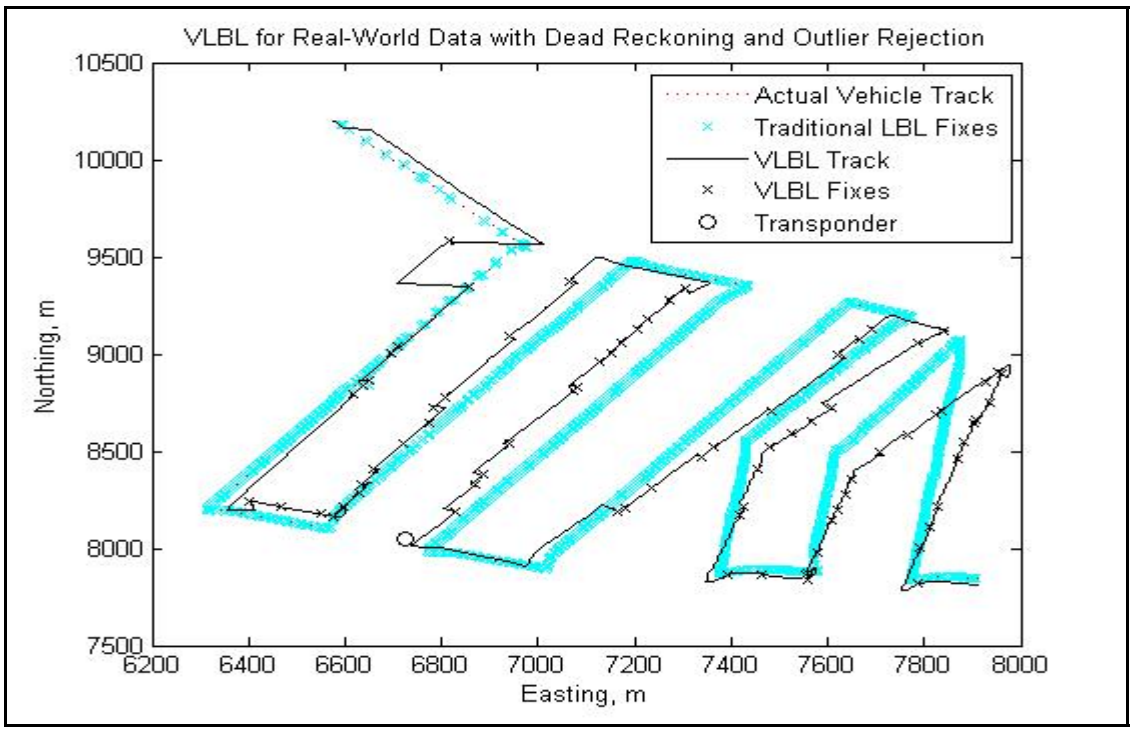

Figure 37: Expanded VLBL Algorithm with ABE162 using Transponder Two and a Sampling Rate of 1 in 1 Ranges with an Outlier Rejection Factor of 1.1

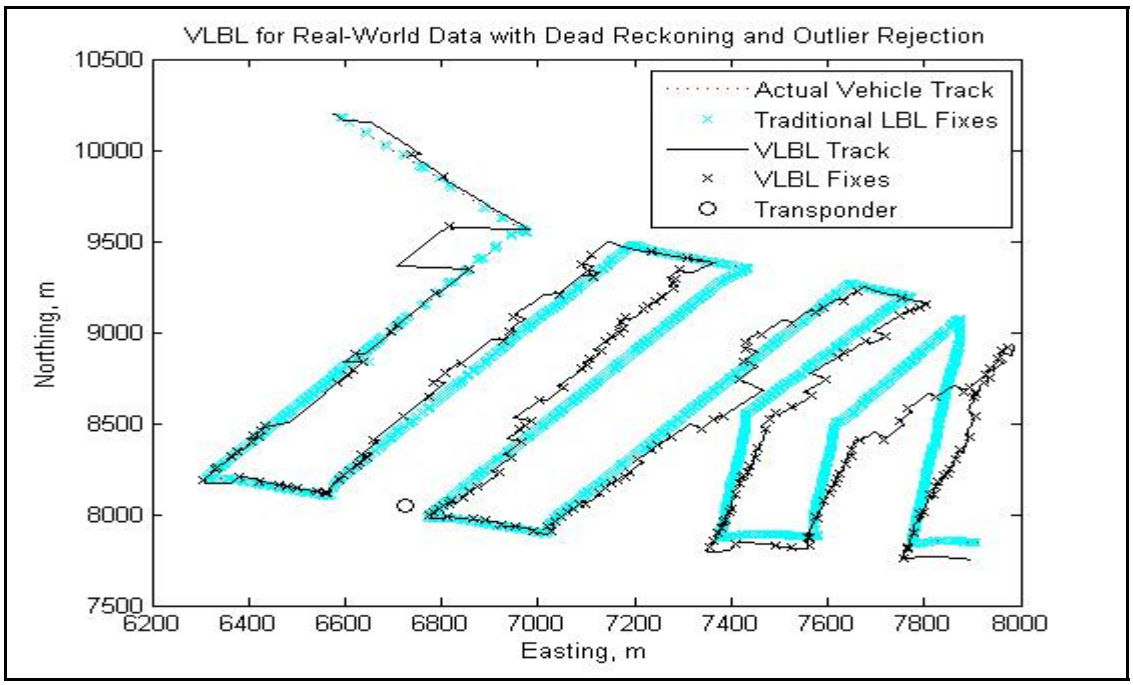

Figure 38: Expanded VLBL Algorithm with ABE162 using Transponder Two, and a Sampling Rate of 1 in 1 Ranges with an Outlier Rejection Factor of 1.5

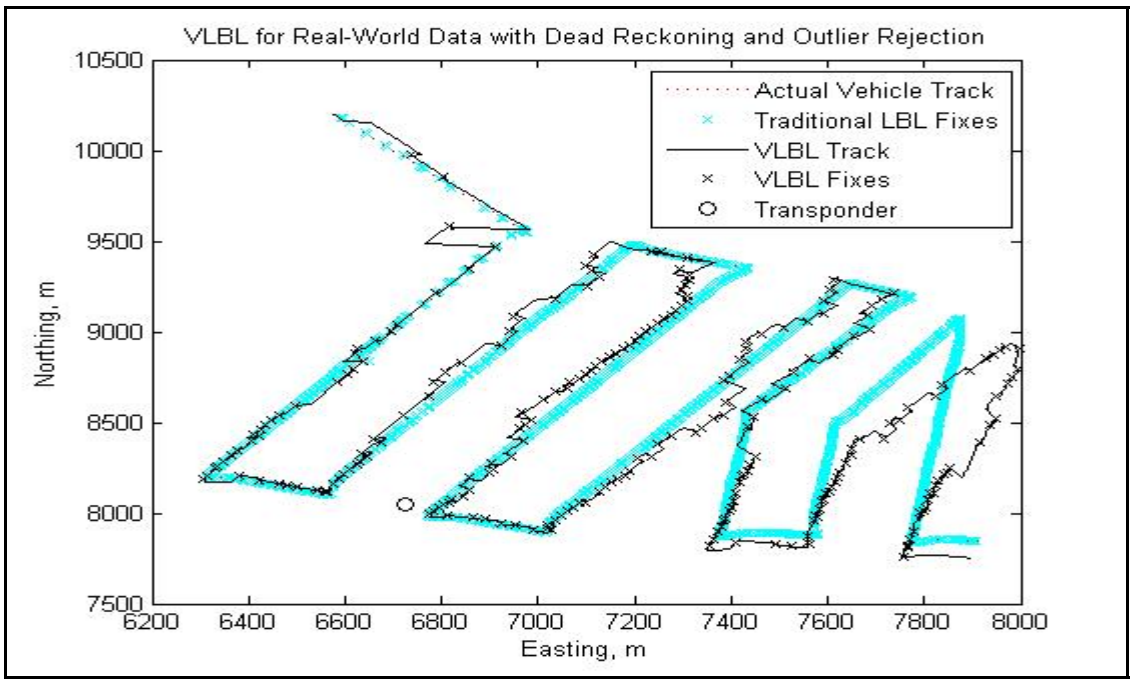

Figure 39: Expanded VLBL Algorithm with ABE162 using Transponder Two, and a Sampling Rate of 1 in 1 Ranges with an Outlier Rejection Factor of 1.8

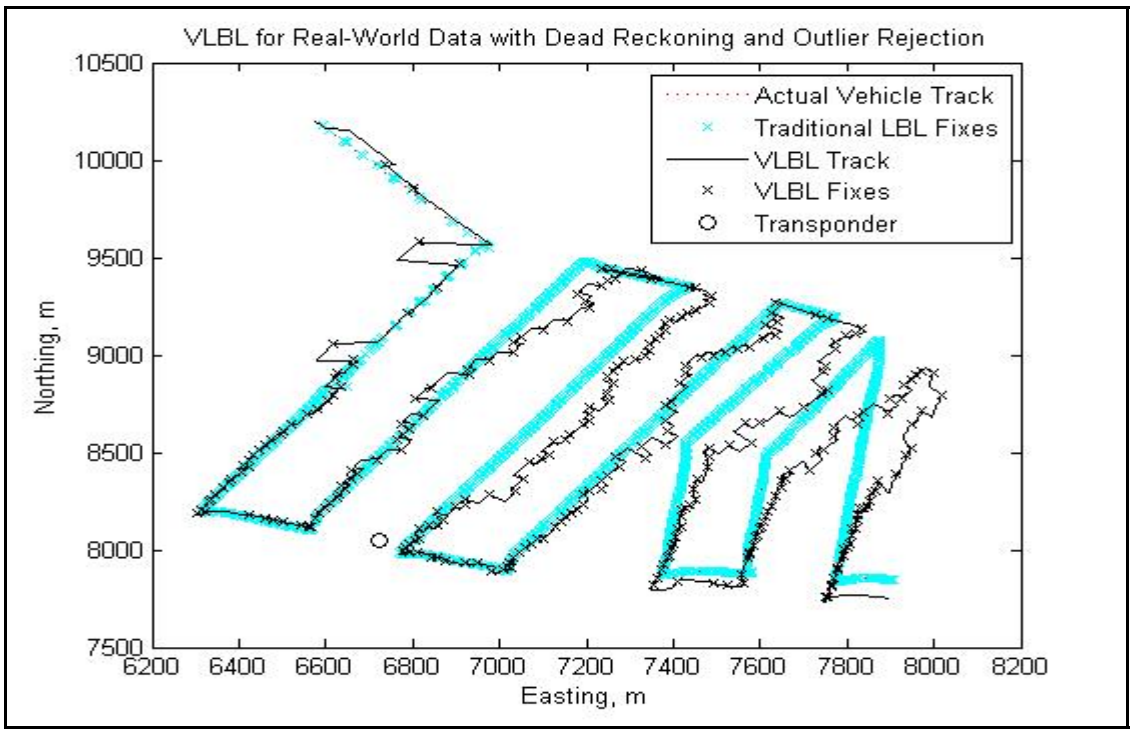

Figure 40: Expanded VLBL Algorithm with ABE162 using Transponder Two, and a Sampling Rate of 1 in 1 Ranges with an Outlier Rejection Factor of 2.5

The best track keeping occurred where an outlier rejection factor of 1.8 was used. When the factor was increased to 2.5, the track began to diverge again because too many fixes were accepted too far from the actual track. These results illustrate the nonlinear relationship between outlier rejection limits and quality of track keeping.

Section 4.2.4: Effect of Transponder Location on Virtual Long Baseline Navigation Performance using Real-World Data

The following series of figures shows the dramatic effect of transponder location on effectiveness of the VLBL system using data from ABE163 in the expanded VLBL algorithm. In Figure 41 and Figure 42, the VLBL performance is comparable to that of the traditional LBL system with some degradation in accuracy using Transponders Three and Four, respectively. However, when the same parameters are used in the VLBL algorithm using range data from Transponders Two and One, the VLBL system completely fails, as shown in Figure 43 and Figure 44. Therefore, the relative geometries

of the external transponder and the vehicle survey patterns are crucial to the effective operation of the VLBL navigation system.

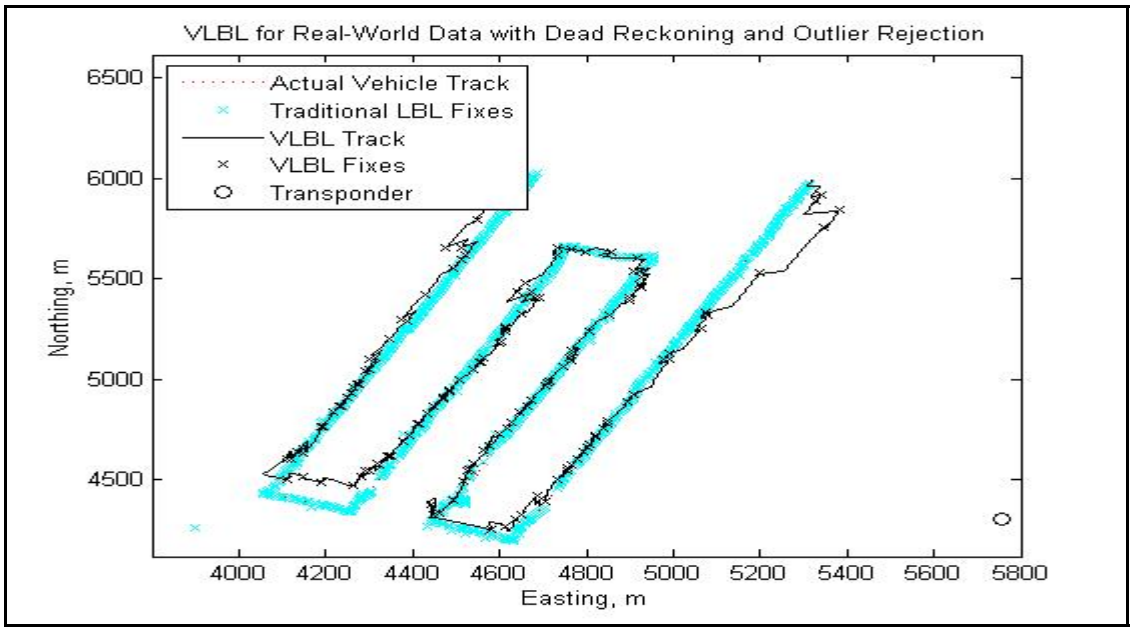

Figure 41: Expanded VLBL Algorithm with ABE163 using a Sampling Rate of 1 in 4 Ranges, and an Outlier Rejection Factor of 2.2 with Transponder Three

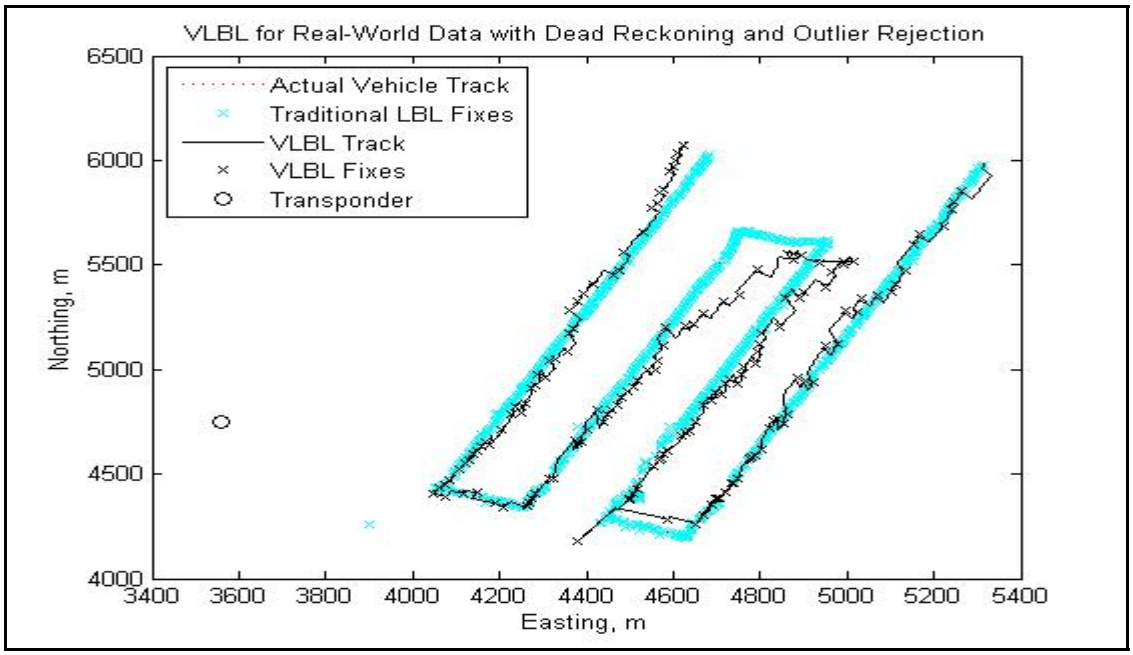

Figure 42: Expanded VLBL Algorithm with ABE163 using a Sampling Rate of 1 in 4 Ranges, and an Outlier Rejection Factor of 2.2 with Transponder Four

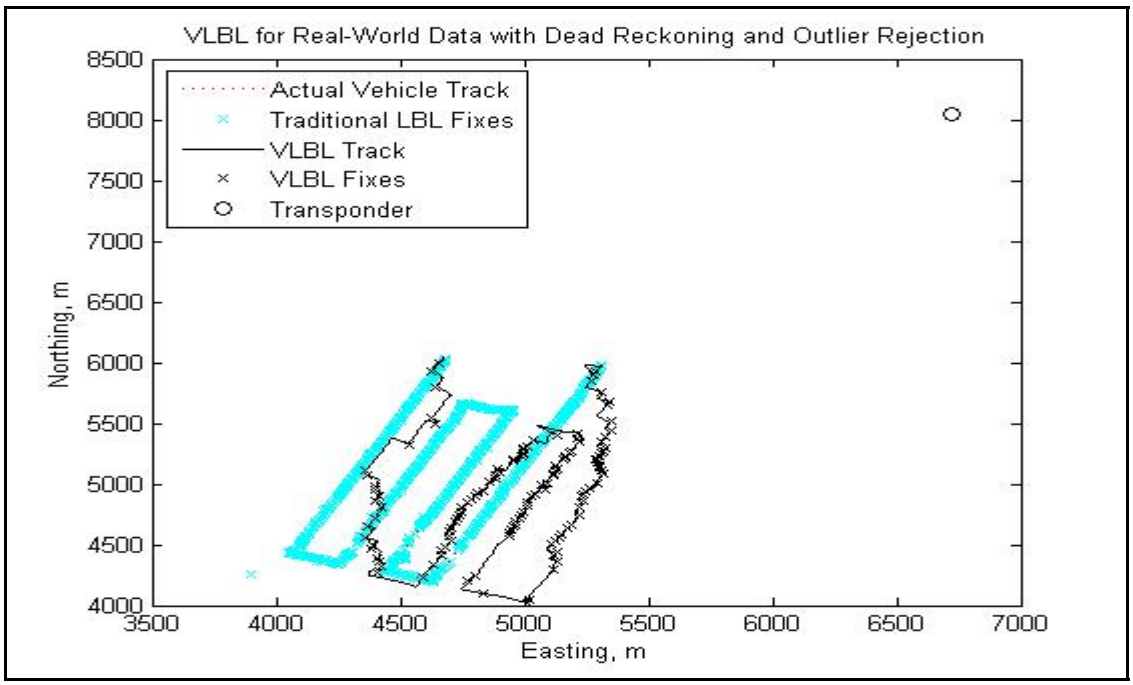

Figure 43: Expanded VLBL Algorithm with ABE163 using a Sampling Rate of 1 in 4 Ranges, and an Outlier Rejection Factor of 2.2 with Transponder Two

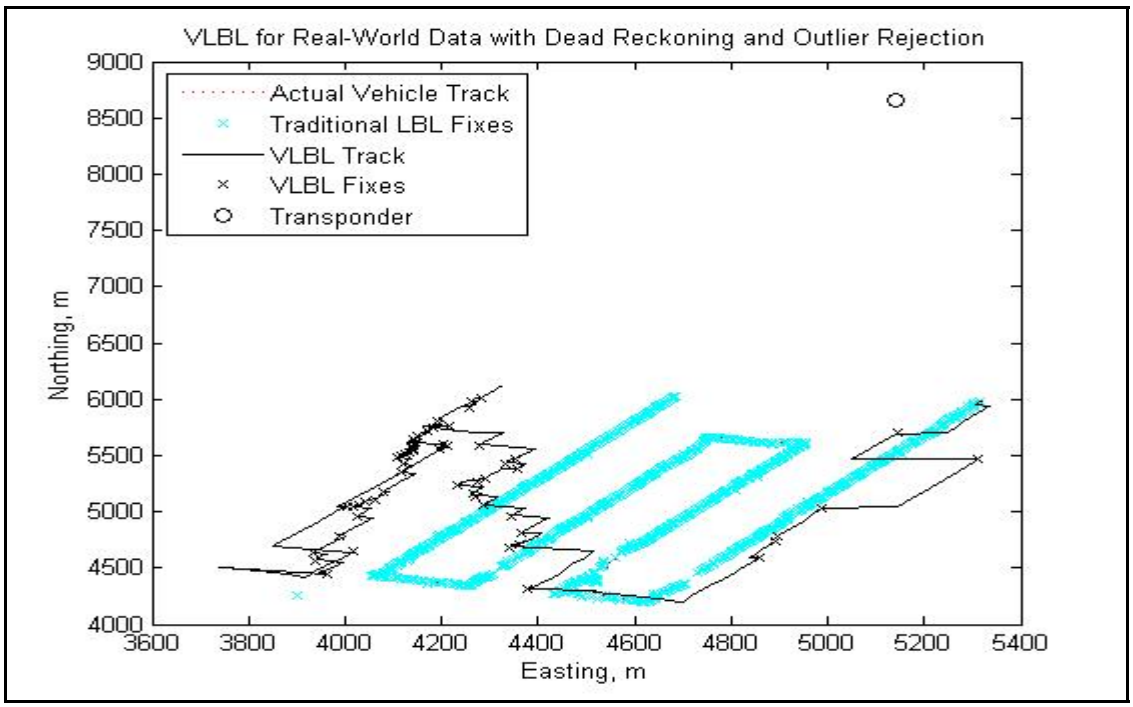

Figure 44: Expanded VLBL Algorithm with ABE163 using a Sampling Rate of 1 in 4 Ranges, and an Outlier Rejection Factor of 2.2 with Transponder One

Section 4.2.5: Error Budget

An important characteristic of any navigation system is the error budget. The error budget as a percentage of distance traveled is the drift rate of the navigation system. When the VLBL algorithm was optimally tuned for each of the ABE dives analyzed, the system achieved drift rates on the order of two percent over the long term. As shown in Figure 45, the VLBL algorithm achieved and maintained drift rates on this order for dive ABE162 after the vehicle had traveled one kilometer.

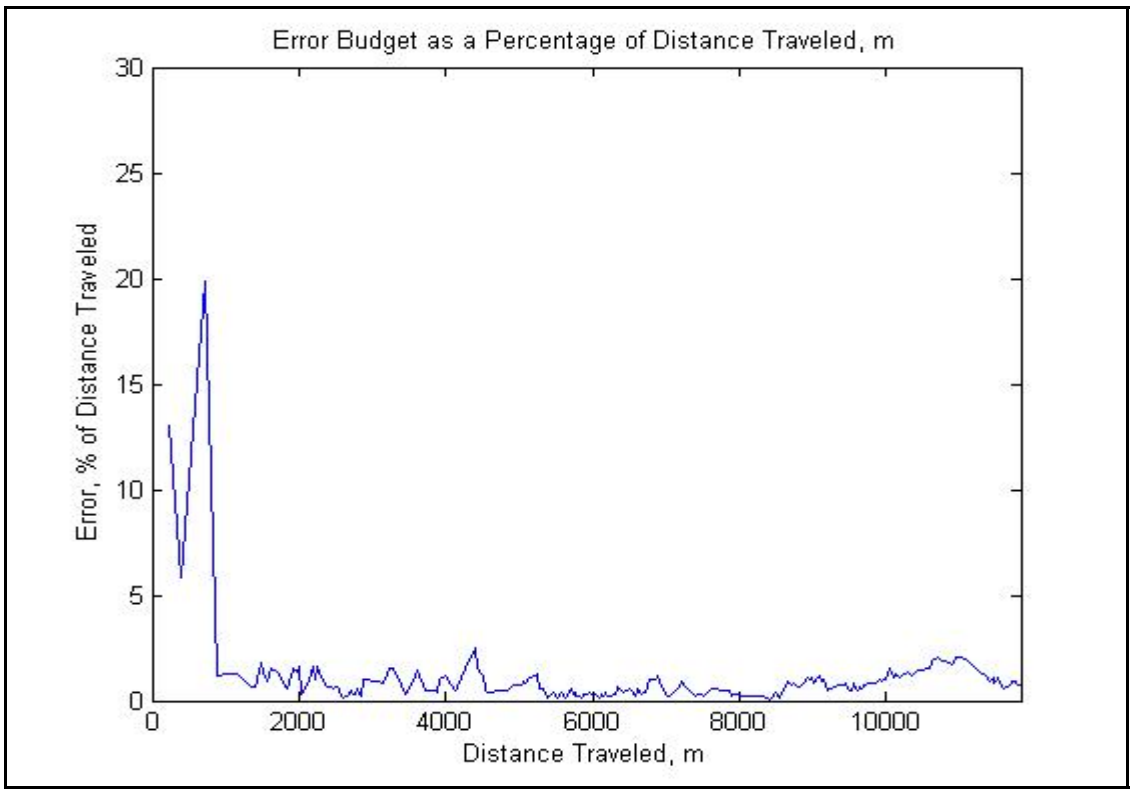

Figure 45: Expanded VLBL Algorithm with ABE162 using Transponder Two, a Sampling Rate of 1 in 4 Ranges, and an Outlier Rejection Factor of 1.8

However, as shown in Figure 46, during dive ABE163 the VLBL algorithm produced fluctuations in drift rate up to five percent during the first eight kilometers of the dive. After that, the drift rate held steady on the order of one percent for the remainder of the dive track.

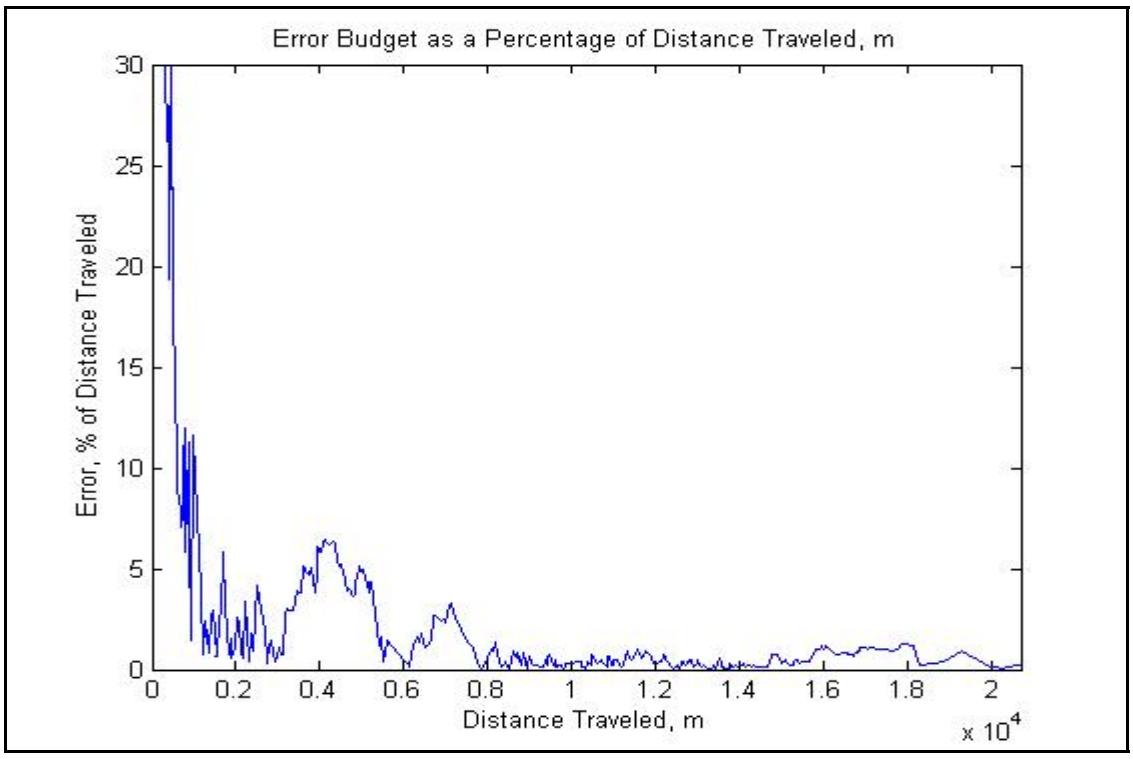

Figure 46: Expanded VLBL Algorithm with ABE163 using Transponder Three, a Sampling Rate of 1 in 4 Ranges, and an Outlier Rejection Factor of 2.2

Section 4.3: Applications and Extensions

Section 4.3.1: Implementing Single Beacon Navigation in Real-Time

The preceding analysis of the performance of the VLBL navigation system highlights both the potential and the limitations of this system. In certain scenarios, the VLBL algorithm was able to approach the performance of the traditional LBL system with some losses in accuracy and update rate. However, the input parameters of sampling rate and outlier rejection factor needed to be tuned in each scenario to achieve optimum performance. Therefore, any real-time implementation of VLBL would require the development of a dynamic method for tuning these parameters in real-time.

Furthermore, the VLBL navigation system completely broke down, wildly diverging from actual vehicle track, in several scenarios. The prime factor in the complete

unobservability of these systems was transponder location with respect to survey pattern geometry. Therefore, any use real-time application of VLBL would require proper planning of transponder location and survey geometry. This requirement would significantly reduce the flexibility of the AUV to further investigate any areas outside of the planned survey.

Given these limitations, the real value in VLBL may be in implementing it as a secondary system within an LBL system using a reduced number of beacons in applications where precision is not vital. Current LBL systems can produce fix estimates using range data from only two transponders, but more are used for redundancy as well as increased accuracy. For large scale surveys, two transponders could be deployed for navigation. Then, in any given navigation cycle, the vehicle would implement VLBL if it only collected one range measurement at that time, thereby providing redundancy in the two beacon system. This scenario also addresses the important issue of direct path blockage between the transponder and the vehicle. If there is only one transponder in the water, careful planning must done to ensure that the direct path to the vehicle from the transponder is never blocked. In a two transponder system, VLBL provides the required navigational redundancy in case one of the transponders is blocked from vehicle view.

Section 4.3.2: Ship-Mounted Single Beacon Navigation

The VLBL algorithm could also be extended to ship-mounted beacon applications via the method as described earlier for MVLBL geometry. MVLBL in this thesis is treated only in the context of cost analysis. However, for a thorough analysis of the real-world application of an algorithm similar to the MVLBL approach described in this thesis, refer to recent work done at WHOI by Carl Hartsfield. [40] Although this scenario is similar to USBL in that there is a single assembly mounted each on the underwater vehicle and on the ship, there is a key distinction. The MVLBL method has the ability to create distance between the virtual transponders at the sacrifice of update rate. Since the ship motion is known accurately through GPS positioning, only the same concerns with dead reckoning track accuracy apply to MVLBL as with VLBL.

Chapter 5: Cost Savings Analysis

Section 5.1: Cost Savings Analysis

The process of deploying, surveying into location and retrieving acoustic transponders can be time intensive and costly. Therefore, reducing or eliminating the need to deploy transponders could translate into significant cost savings. An analysis of the potential savings in operating costs of a VLBL navigation system versus a traditional LBL system was undertaken using real-world data from ABE operations on the R/V Atlantis in the Juan de Fuca Straits in 2004.

Section 5.1.1: Method of Analysis

Data concerning the exact duration of launch, survey and recovery operations were taken from the Deck Log of the R/V Atlantis on voyage number 11-14. [41] During this voyage, ABE was deployed for a series of bottom survey dives in two general locations. Upon arriving at each location of interest, the R/V Atlantis deployed an LBL network of three acoustic transponders. The location of the transponders was subsequently surveyed into position through a series of ship maneuvers while interrogating the beacons. Multiple ABE dives were undertaken at each location before the acoustic transponders were recovered. At the first dive site, equipment malfunctions on several occasions necessitated the retrieval and redeployment of a single transponder. The time data recorded from the deck logs was averaged for all of the transponder operations to determine average handling times per beacon as shown in Equations (5.1) and (5.2).

The potential cost savings associated with the real-world application of VLBL and MVLBL were determined based on the average transponder handling time calculations.

In order to determine the associated costs of transponder operations, the 2006 day rate for the R/V Atlantis of $30,466 was used. The operating costs of the ship incurred during transponder handling were calculated using this day rate as shown in Equation (5.3).

$$\overline{THT} = \overline{T_{deploy}} + \overline{T_{survey}} + \overline{T_{recover}} \quad (5.1)$$

$$VTHT(nt, nd) = \overline{THT} * nt * nd \quad (5.2)$$

$$VTHC(nt, nd) = VTHT(nt, nd) * DayRate \quad (5.3)$$

where

$\overline{T_{deploy}}$ = Average time to deploy one transponder

$\overline{T_{survey}}$ = Average time to survey the location of one transponder

$\overline{T_{recover}}$ = Average time to recover one transponder

$\overline{THT}$ = Average total handling time for one transponder

nt = Number of transponders used at each dive site

nd = Number of dive sites on a given voyage

$VTHT(nt, nd)$ = Voyage Transponder Handling Time

$DayRate$ = R/V Atlantis CY-06 Day Rate

$VTHC$ = Voyage Transponder Handling Cost

Section 5.1.2: Critical Assumptions

Several critical assumptions were made during this analysis. The most important assumption was that the bulk of the savings associated with VLBL would be realized in terms of operating costs. Therefore, any cost differences associated with initial expenditures on equipment procurement or vehicle software modification were excluded from the analysis. There were also assumptions made in the data collection from ship

deck logs. Since the initiation of transponder deployment and recovery operations were not recorded in the deck logs, it was assumed that transponder deployment began when the first transponder was listed as streaming and that the transponder recovery began when ABE was on deck after the last dive on site. Furthermore, it was assumed that transponder handling time increased linearly with an increased number of transponders.

Section 5.2: Results

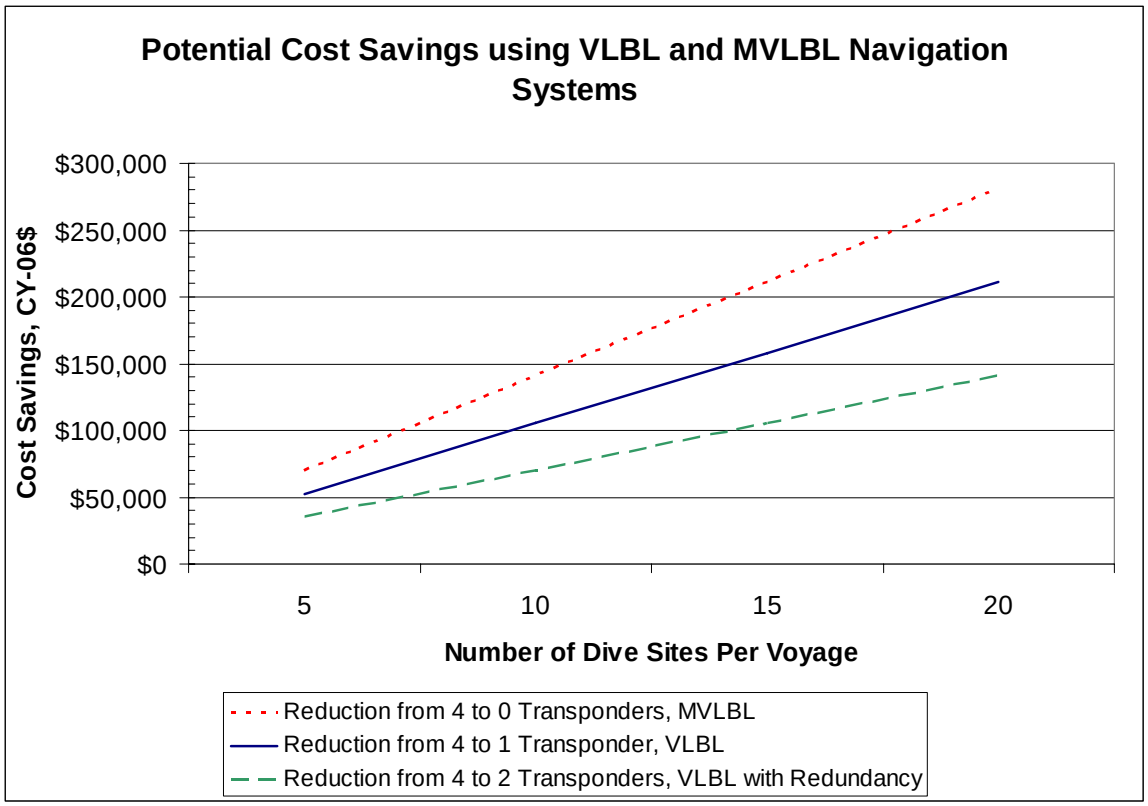

Figure 47: Voyage Costs Associated with Various Operating Profile Assumptions

The cost savings potential associated with the VLBL and MVLBL were calculated for a range of representative voyage lengths. Figure 47 shows these cost savings for voyages covering between five and twenty dive sites. As listed in Table 2, this corresponds to savings of $10,560 per dive site for implementation of VLBL using only one transponder

and $14,080 for implementation of MVLBL with no transponders. For a voyage including twenty distinct dive sites, the accumulated savings of VLBL over the entire voyage is $211,199 and that of MVLBL is $281,599. Even if two transponders are deployed to provide redundancy for the VLBL navigation system, the savings over twenty dive sites is still $140,799. The complete calculations are included in Appendix D.

Table 2: Summary of Cost Analysis Results

	PER DIVE SITE
Reduction from 4 to 2 Transponders, VLBL with Redundancy	$7,040
Reduction from 4 to 1 Transponder, VLBL	$10,560
Reduction from 4 to 0 Transponders, MVLBL	$14,080

It is also important to note that the cost savings here were calculated only in terms of actual dollars saved based on time saved. However, there is another advantage that has not adequately been captured by this cost savings model, namely the increased availability of a critical asset. The number of research vessels suitable for deep ocean research operations is limited. Therefore, time saved through more efficient underwater vehicle evolutions corresponds to increased availability of this limited asset and the potential to accomplish more research.

Chapter 6: Conclusions

Section 6.1: Contributions

This thesis has presented a methodology for single beacon underwater navigation and applied that methodology to the post-processing of real-world navigation data from a deep ocean AUV. The development of a Virtual Long Baseline acoustic transponder net theoretically allows an underwater vehicle to navigate in a method similar to a traditional long baseline method with significant potential for reduced operational costs. The effectiveness of the VLBL method has been studied in the framework of an analysis of observability. The impact on observability of three main variable inputs to the VLBL was studied in depth.

Since the virtual transponders are created based on the vehicle's movement between range calculations, the geometry between vehicle track and transponder location has been shown to be critically important to the issue of observability. In certain relative geometries between the transponder and the vehicle track, the VLBL system is not effective at all. These geometries involve numerous long tracklines that are oriented either directly towards or away from the beacon, or nearly so. Careful permission planning would be required for real-time operations in order to avoid this catastrophic situation. Either the vehicle survey pattern would need to be planned with the long tracklines oriented tangent to range circles emanating from the transponders or the vehicle would need to adopt a meandering track when traveling toward or away from the beacon, similar to tacking a sailboat into the wind.

In order to achieve adequate separation between virtual transponders in comparison to the overall distance between the vehicle and the transponder, the sampling rate of the historical range data is also a key factor in observability. Furthermore, the choice of outlier rejection factor also plays an important role. Too restrictive factors preclude the vehicle from obtaining any position fixes while too lenient outlier rejection factors allow

spurious fixes to be accepted by the system. Both sampling rate and outlier rejection factor exhibit a nonlinear relationship with the quality of path following by the VLBL algorithm for different operational geometries. Therefore, a dynamic method of real-time tuning of these parameters would be required in order for the VLBL algorithm to be adapted for real-time operations.

Finally, even a tuned VLBL navigation system does not achieve the operational precision of a traditional LBL system. Therefore, its real value may be in providing redundancy for a traditional LBL system using a reduced number of transponders. Another potential application of the VLBL system is doing initial surveys for large-scale underwater phenomena in which navigational accuracy and precision may be sacrificed in order to achieve the operational cost savings. Additional transponders could then be placed into the water when the phenomena are detected to use traditional LBL navigation for more precise operations.

Section 6.2: Future Work

Future work with VLBL could apply it to real-time navigation applications. As discussed earlier, there are significant challenges associated with this extension. For example, dynamic methods for adjusting sampling methods and outlier rejection factors must be developed for VLBL to function as a primary navigation system. In the immediate future, however, VLBL could be implemented as a redundant system to traditional LBL algorithms for use in any navigational cycle in which range data from only one external acoustic transponder is available.

Bibliography

[1] J. J. Leonard, A. A. Bennett, C. M. Smith, and H. J. S. Feder, "Autonomous Underwater Vehicle Navigation," Massachusetts Institute of Technology, MIT Marine Robotics Laboratory Technical Memorandum 98-1, 1998.

[2] L. Whitcomb, D. Yoerger, H. Singh, and D. Mindell, "Towards Precision Robotic Maneuvering, Survey, and Manipulation in Unstructured Undersea Environments," presented at Robotics Research- The Eighth International Symposium, 1998.

[3] "http://www.sonardyne.co.uk/Support/PositioningTechniques/index.html#sbl," Sonardyne.

[4] P. H. Milne, *Underwater Acoustic Positioning Systems*. London: E. & F. N. Spon, 1983.

[5] J. Vaganay, J. Leonard, and J. G. Bellingham, "Outlier Rejection for Autonomous Acoustic Navigation," presented at IEEE International Conference on Robotics and Automation, Minneapolis, Minnesota, 1996.

[6] S. M. Smith and D. Kronen, "Experimental Results of an Inexpensive Short Baseline Acoustic Positioning System for AUV Navigation," presented at Oceans 1997 MTS/IEEE, Halifax, NS, 1997.

[7] "http://www.dosits.org/gallery/tech/n/at1.htm," Office of Marine Programs, University of Rhode Island.

[8] "http://www.ifremer.fr/fleet/systemes_sm/positionnement/posidonia.htm," French Research Institute for Exploitation of the Sea.

[9] L. L. Whitcomb, D. R. Yoerger, and H. Singh, "Combined Doppler/LBL Based Navigation of Underwater Vehicles," in *11th International Symposium on Unmanned Untethered Subersible Technology*. Durham, New Hampshire, 1999.

[10] B. S. Bingham, "Precision Autonomous Underwater Vehicle Navigation," in *Mechanical Engineering*, vol. Doctor of Philosophy. Cambridge, Massachusetts: Massachusetts Institute of Technology, 2003, pp. 217.

[11] L. Whitcomb, D. Yoerger, H. Singh, and J. Howland, "Advances in Underwater Robot Vehicles for Deep Ocean Exploration: Navigation, Control, and Survey Operations," presented at Ninth International Symposium of Robotics Research, Snowbird, Utah, 1999.

[12] D. E. Leader, "Kalman Filter Estimation of Underwater Vehicle Position and Attitude Using a Doppler Velocity Aided Inertial Motion Unit," in *Joint Program in Oceanographic Engineering*, vol. Ocean Engineer. Cambridge, Massachusetts: Massachusetts Institute of Technology and Woods Hole Oceanographic Institution, 1994, pp. 106.

[13] L. Whitcomb, D. Yoerger, and H. Singh, "Advances in Doppler-based Navigation of Underwater Robotic Vehicles," presented at IEEE International Conference on Robotics and Automation, 1999.

[14] M. B. Larsen, "Synthetic Long Baseline Navigation of Underwater Vehicles," presented at Oceans 2000 MTS/IEEE Conference and Exhibition, 2000.

[15] P. Newman and J. Leonard, "Pure Range-Only Sub-Sea SLAM," presented at IEEE International Conference on Robotics and Automation, 2003.
[16] C. M. Smith and J. J. Leonard, "A Multiple Hypothesis Approach to Concurrent Mapping and Localization for Autonomous Underwater Vehicles," presented at International Conference on Field and Service Robotics, Sydney, Australia, 1997.
[17] K. Ho and P. Newman, "Multiple Map Intersection Detection using Visual Appearance," presented at 3rd International Conference on Computational Intelligence, Robotics and Autonomous Systems, Singapore, 2005.
[18] P. M. Newman, J. J. Leonard, and R. R. Rikoski, "Towards Constant-Time SLAM on an Autonomous Underwater Vehicle Using Synthetic Aperture Sonar," presented at International Symposium on Robotics Research, Vienna, 2003.
[19] R. M. Eustice, "Large-Area Visually Augmented Navigation for Autonomous Underwater Vehicles," in *Joint Program in Oceanography/Applied Ocean Science and Engineering*, vol. Doctor of Philosophy. Cambridge, Massachusetts: Massachusetts Institute of Technology and Woods Hole Oceanographic Institution, 2005, pp. 187.
[20] C. Roman, "Self Consistent Bathymetric Mapping From Robotic Vehicles in the Deep Ocean," in *Joint Program in Oceanography/Applied Ocean Science and Engineering*, vol. Doctor of Philosophy. Cambridge, Massachusetts: Massachusetts Institute of Technology and Woods Hole Oceanographic Institution, 2005, pp. 129.
[21] A. P. Scherbatyuk, "The AUV Positioning Using Ranges from One Transponder LBL," presented at OCEANS 1995, 1995.
[22] J. Vaganay, P. Baccou, and B. Jouvencel, "Homing by Acoustic Ranging to a Single Beacon," presented at OCEANS 2000 MTS/IEEE Conference and Exhibition, Providence, RI, 2000.
[23] P. Baccou and B. Jouvencel, "Simulation results, post-processing experimentations and comparisons results for navigation, homing and multiple vehicles operations with a new positioning method using on transponder," presented at 2003 IEEE/FSJ International Conference on Intelligent Robots and Systems, Las Vegas, NV, 2003.
[24] P. Baccou and B. Jouvencel, "Homing and navigation using one transponder for AUV, postprocessing comparisons results with long base-line navigation," presented at IEEE Conference on Robotics and Automation, 2002.
[25] M. B. Larsen, "High Performance Autonomous Underwater Navigation," *Hydro International*, vol. 6, 2002.
[26] A. S. Gadre and D. J. Stilwell, "Toward Underwater Navigation Based on Range Measurements from a Single Location," presented at 2004 IEEE Conference on Robotics and Automation, New Orleans, LA, 2004.
[27] A. S. Gadre and D. J. Stilwell, "Underwater Navigation in the Presence of Unknown Currents Based on Range Measurements from a Single Location," presented at 2005 American Control Conference, 2005.

[28] A. Ross and J. Jouffroy, "Remarks on the Observability of Single Beacon Underwater Navigation," presented at 14th International Symposium on Unmanned Untethered Submersible Technology, Durham, NH, 2005.
[29] D. Yoerger, "The Autonomous Benthic Explorer (ABE)," 2002.
[30] D. Yoerger, A. Bradley, P. Bachmayer, R. Catanach, A. Duester, S. Liberatore, H. Singh, and B. Walden, "Near-Bottom Magnetic Surveys of the Coaxial Ridge Segment Using the Autonomous Benthic Explorer Survey Vehicle," *RIDGE Events*, pp. 5-9, 1996.
[31] D. R. Yoerger, "Robotic Undersea Technology," *Oceanus*, vol. 34, pp. 32-37, 1991.
[32] D. R. Yoerger, A. M. Bradley, M.-H. Cormier, W. B. F. Ryan, and B. B. Walden, "High Resolution Mapping of a Fast Spreading Mid Ocean Ridge with Autonomous Benthic Explorer," in *11th International Symposium on Unmanned Untethered Submersible Technology*. Durham, NH, 1999.
[33] D. R. Yoerger, A. M. Bradley, H. Singh, B. B. Walden, M.-H. Cormier, and W. B. F. Ryan, "Multisensor Mapping of the Deep Seafloor with the Autonomous Benthic Explorer," *IEEE ??*, pp. 248-253, 2000.
[34] D. R. Yoerger, A. M. Bradley, and B. B. Walden, "The Autonomous Benthic Explorer (ABE): An AUV Optimized for Deep Seafloor Studies," presented at 7th International Symposium on Unmanned Untethered Submersible Technology, Durham, New Hampshire, 1991.
[35] D. R. Yoerger, A. M. Bradley, and B. B. Walden, "Dynamic Testing of the Autonomous Benthic Explorer," presented at 8th International Symposium on Unmanned Untethered Submersible Technology, Durham, New Hampshire, 1993.
[36] D. R. Yoerger, A. M. Bradley, and B. B. Walden, "An AB(L)E Bodied Vehicle Proves Its Worth," *Oceanus*, vol. 39, pp. 20-21, 1996.
[37] D. R. Yoerger, A. M. Bradley, and B. B. Walden, "Scientific Survey with the Autonomous Benthic Explorer," presented at 10th International Symposium on Unmanned Untethered Submersible Technology, Durham, New Hampshire, 1997.
[38] D. R. Yoerger, A. M. Bradley, B. B. Walden, H. Singh, and R. Bachmayer, "Surveying a subsea lava flow using the Autonomous Benthic Explorer (ABE)," *International Journal of Systems Science*, vol. 29, pp. 1031-1044, 1998.
[39] D. R. Yoerger, M. Jakuba, A. M. Bradley, and B. Bingham, "Techniques for Deep Sea Near Bottom Survey Using an Autonomous Underwater Vehicle," *International Journal of Robotics Research*, vol. Submitted. Awaiting publication., 2005.
[40] J. C. Hartsfield, Jr., "Single Transponder Range Only Navigation Geometry (STRONG) Applied to REMUS Autonomous Underwater Vehicles," in *Joint Program in Oceanography/Applied Ocean Science and Engineering*, vol. Master of Science Cambridge, Massachusetts: Massachusetts Institute of Technology and Woods Hole Oceanographic Institution, 2005, pp. 125.
[41] "R/V Atlantis Deck Logs Voyage 11-14," 2004.

Appendices

Appendix A: Mathematical Models from the Single Beacon Navigation Literature Review

A.1: Least Squares Model

The mathematical least squares model set up by Scherbatyuk based in a Cartesian coordinate system was given by the following system of equations:

$$[x_a(0) + (v_x + w_x)k\tau - x_t]^2 + [y_a(0) + (v_y + w_y)k\tau - y_t]^2 = \tilde{d}^2(k\tau),$$

$$\tilde{d}^2(k\tau) = d^2(k\tau) - [z_a - z_t]^2, \qquad \text{(A.1)}$$

$$v_x = v * \sin\varphi, v_y = v * \cos\varphi,$$

where:

$d(k\tau)$ = Measured ranges from transponder to AUV,

(x_t, y_t, z_t) = Known transponder position,

(x_a, y_a, z_a) = AUV position coordinates,

(v, φ) = AUV relative velocity and yaw on a straight-line path,

(w_x, w_y) = Current velocity on a straight-line path,

τ = Period of LBL operations,

$(x_a(0), y_a(0))$ = Initial AUV position on a straight-line path.

Using this mathematical model, Scherbatyuk reformulated the set of Equations (A.1) into the following quadratic relationship for solution via a least squares method.

$$\tilde{d}^2(k\tau) = A * (k\tau)^2 + B * k\tau + C, \qquad \text{(A.2)}$$

where:

$$A = (v_x + w_x)^2 + (v_y + w_y)^2,$$

$$B = 2[(x_a(0) - x_t)(v_x + w_x) + (y_a(0) - y_t)(v_y + w_y)], \quad \text{(A.3)}$$

$$C = (x_a(0) - x_t)^2 + (y_a(0) - y_t)^2 .$$

The least squares method is applied to estimate the *A*, *B*, and *C* values by minimizing the sum of differences *J* between the ranges $\tilde{d}^2(k\tau)$ observed in each time step $k\tau$, (*k* = 1,…,*K*), and the curve defined by Equation(A.1), where:

$$J = \sum_{k=1}^{K} Q(k) * E(k), \quad \text{(A.4)}$$

$$Q(k) = A*(k\tau)^2 + B*k\tau + C - \tilde{d}^2(k\tau), \quad \text{(A.5)}$$

E(k) = 1 for the transponder response received at the time $k\tau$ and *E(k)* = 0 otherwise. Mathematically, in order for a solution to the least squares minimization problem to exist, i.e. for the system to be observable, Scherbatyuk found that the following condition must be true:

$$\begin{vmatrix} (v_{x2} - v_{x1}) & (v_{x3} - v_{x2}) \\ (v_{y2} - v_{y1}) & (v_{y3} - v_{y2}) \end{vmatrix} \neq 0, \quad \text{(A.6)}$$

Where the AUV relative velocities on each of the three straight-line paths are given by (v_{x1}, v_{y1}), (v_{x2}, v_{y2}) and (v_{x3}, v_{y3}). This is equivalent to saying that the AUV must travel on straight-line paths in three distinct directions before an initial fix can be computed.

A.2: Extended Kalman Filter Model

Much of the existing literature on single beacon range only navigation is based on an Extended Kalman Filter concept. Although some of the authors differ in the details, they are based on the same general concepts. The model described below is a representative example. Set in a North-East coordinate system, the mathematical model used by Vaganay, Baccou and Jouvencel with an Extended Kalman Filter obtains system initialization with a Levenberg-Marquardt non-linear least squares algorithm is as follows:

$$\Delta x_{i,n} = \sum_{k=i}^{k=n} C\theta_k C\psi_k u_k \Delta t - du \sum_{k=i}^{k=n} C\theta_k C\psi_k \Delta t + v_{cn} \sum_{k=i}^{k=n} \Delta t \,,$$

$$\Delta y_{i,n} = \sum_{k=i}^{k=n} C\theta_k S\psi_k u_k \Delta t - du \sum_{k=i}^{k=n} C\theta_k S\psi_k \Delta t + v_{ce} \sum_{k=i}^{k=n} \Delta t \,, \tag{2.7}$$

$$d_i = [(x_b - x + \Delta x_{i,n})^2 + (y - y + \Delta y_{i,n})^2]^{1/2} \,,$$

where:

$(\Delta x_{in}, \Delta y_{in})$ = Vehicle North and East displacements between ranges i and *n*,

θ = Vehicle pitch,

ψ = Vehicle heading,

u = Vehicle calibrated water referenced speed,

Δt = Sampling period,

du = Speed bias,

(v_{cn}, v_{ce}) = Underwater North (x) and East (y).

A non-linear least squares optimization method was then used to determine vehicle initial position.

$$P = \left[\left(\frac{\partial f(X)}{\partial X} \right)^t (HP_e H_t + R)^{-1} \left(\frac{\partial f(X)}{\partial X} \right) \right]^{-1} \,, \tag{2.8}$$

where:

P = Covariance matrix used to initialize the estimation error covariance matrix of the EKF,

X = $(x, y, v_{cn}, v_{ce}, du)^t$,

H = Jacobian of the ranges with respect to ψ, θ, and u,

P_e = Covariance matrix of ψ, θ, and u,

R = Variance matrix of the Gaussian noise on the measured ranges.

The Extended Kalman Filter's equation of state was defined as follows:

$$\begin{bmatrix} x \\ y \\ z \\ v_{cn} \\ v_{ce} \\ du \end{bmatrix}_{k+1} = \begin{bmatrix} x \\ y \\ z \\ v_{cn} \\ v_{ce} \\ du \end{bmatrix}_{k} + \begin{bmatrix} C\theta C\psi(u-du)+v_{cn} \\ C\theta S\psi(u-du)+v_{ce} \\ -S\theta(u-du) \\ 0 \\ 0 \\ 0 \end{bmatrix}_{k} \Delta t + v_k \qquad (2.9)$$

The Extended Kalman Filter works continuously by determining a state prediction and then correcting it every time a new range is available.

Appendix B: Basic VLBL Algorithm

```
% Load the correct data file
% Start and End Time Indices refer to the lbl structure
S = input('Choose which dive to use (153,162 or 163) or enter ...
      100 for test data:');
if S == 162;
    FileLoad162;
    Start_Index = 1500; % 1
    End_Index = 4000; % 3000
elseif S == 163;
    FileLoad163;
    Start_Index = 1000;
    End_Index = 5000;
elseif S == 153;
    FileLoad153;
    Start_Index = 1200; % 1100
    End_Index = 4000; % 5000
elseif S == 100;
    FileLoadTest2;
    Start_Index = 1;
    End_Index = 2001;
else
    fprintf('Invalid choice.\n');
    return;
end;

% Create a path to the dsl utility functions in the dslutils folder
path(path,'./dslutils');

% Plot all four transponder positions
figure;
plot3(xp_x(1),xp_y(1),-xp_z(1),'ro',xp_x(2),xp_y(2),-xp_z(2),'rx',...
    xp_x(3),xp_y(3),-xp_z(3),'r+',xp_x(4),xp_y(4),-xp_z(4),'r*')
grid on
title('Transponder Positions');
xlabel('Easting, m');
ylabel('Northing, m');
zlabel('Depth, m');
legend('Transponder 1','Transponder 2','Transponder 3','Transponder...
      4');

% Choose a transponder for the single beacon calculations
i = input('Choose which transponder to use (1 to 4):');

% Isolate the x,y,z position of the chosen transponder
xpx = xp_x(i);
xpy = xp_y(i);
xpz = xp_z(i);
```

```
% Define the start and end times of the analysis
Start_Time = lbl.t(Start_Index);
End_Time = lbl.t(End_Index);

% Extract the data which is needed from the lbl structure for the
% indicated time range
lbl_sb = sext(lbl,find(lbl.t > Start_Time & lbl.t < End_Time));

% Take only the lbl data points from the chosen transponder which
% correspond to status 5, which means good data
lbl_SB = sext(lbl_sb,find(lbl_sb.status(:,i)==5));

% Create new time reference starting with zero at first good data pt:
lbl_SB.T = lbl_SB.t - Start_Time;      % Time wrt new reference, sec

% Run a function to test horizontal ranges for exceeding maximum,
% minimum and median tests
[Time,Range] = RangeTest(lbl_SB.T,lbl_SB.r(:,i));
L = length(Time);

% Extract the data which is needed from the state structure
state_SB   = sext(state,find(state.t > Start_Time & state.t ...
      <  End_Time));

% Put state data with respect to new time reference
state_SB.T = state_SB.t - Start_Time;

% Find the necessary data points from the state structure at the time
% of each good range data point
I = zeros(L,1);
Heading = zeros(L,1);
Depth = zeros(L,1);
U = zeros(L,1);
V = zeros(L,1);
W = zeros(L,1);
StateX = zeros(L,1);
StateY = zeros(L,1);

for i = 1:L;
    I(i)       = find(abs(state_SB.T-Time(i))==min(abs(state_SB.T-
      Time(i))));
    Heading(i) = state_SB.hdg(I(i));        % Vehicle hdg, 0:2pi rad
    Depth(i)   = state_SB.depth(I(i));      % Vehicle depth, m
    U(i)       = state_SB.u(I(i));          % Vehicle forward vel, m/s
    V(i)       = state_SB.v(I(i));          % Vehicle sideways vel, m/s
    W(i)       = state_SB.w(I(i));          % Vehicle vertical vel, m/s
    StateX(i)  = state_SB.x(I(i));          % Vehicle state position -
                                                              goal
    StateY(i)  = state_SB.y(I(i));          % Vehicle state position -
                                                              goal
end
```

```
% Transform the vehicle's velocity vectors from local (body oriented)
% coordinates to global coordinates, and plot them vs time
VelX    = U.*sin(Heading) + V.*cos(Heading);   % Vehicle east vel, m/s
VelY    = U.*cos(Heading) - V.*sin(Heading);   % Vehicle north vel, m/s

% Plot velocities and headings
figure;
plot(Time,U,'b',Time,V,'r',Time,W,'g',Time,VelX,'c',Time,VelY,'k')
title('Vehicle Velocity Components');
xlabel('Time, sec');
ylabel('Vehicle Component Velocity, m/s');
legend('Forward','Sideways','Vertical','X Direction','Y Direction');

figure;
plot(Time,Heading)
title('Vehicle Headings');
xlabel('Time, sec');
ylabel('Vehicle Heading, rad');

% Choose sampling rate of data points to use
I_skip = input('Choose I_skip to use:');
count = 1;
for i = 1:L;
    if mod(i,I_skip) == 0;
        time(count)  = Time(i);
        range(count) = Range(i);
        depth(count) = Depth(i);
        velX(count)  = VelX(i);
        velY(count)  = VelY(i);
        w(count)     = W(i);
        stateX(count)= StateX(i);
        stateY(count)= StateY(i);
        count = count + 1;
    end;
end;
L = count - 1;
time = time';
range = range';
depth = depth';
velX = velX';
velY = velY';
w = w';

% Find the four virtual transponder locations by advancing locations
% of a single transponder through 3 time steps in the past
[X,Y,Z,Z0,R] = RunningRange(time,xpx,xpy,xpz,velX,velY, ...
      w,depth,range,L);

% Plot the horizontal ranges and differences between successive ranges
Diff34 = R(:,4) - R(:,3);
Diff23 = R(:,3) - R(:,2);
```

```
Diff12 = R(:,2) - R(:,1);

figure;
subplot(211);
plot(R);
title('Horizontal Ranges');
ylabel('Ranges, m');
legend('VT1','VT2','VT3','VT4');
subplot(212);
plot(time,Diff34,'b',time,Diff23,'r',time,Diff12,'g');
title('Differences in Horizontal Range between Virtual Transponder...
      Positions')
xlabel('Time, sec');
ylabel('Offset, m');
legend('VT3 and VT4','VT2 and VT3','VT1 and VT2')

% Least Squares solution for spherical positioning using lbl_ls_method
Xstart = stateX(4);
Ystart = stateY(4);
poseInitial = [Xstart Ystart];

PC = 1;
D_Trav(1) = 0;
% Choose an acceptance level for residuals
Accepted_Residual = input('Choose acceptable LS residual (e.g. 50):');

for ind = 1:L;
    PosX(PC) = poseInitial(1);
    PosY(PC) = poseInitial(2);
    if ind >= 2
        dt(ind) = time(ind) - time(ind-1);
        D_Trav(ind) = D_Trav(ind-1)+((velX(ind)^2+velY(ind)^2)^0.5) ...
                * dt(ind);
    end
    ranges = R(ind,:);
    beacons = [X(ind,1) Y(ind,1); X(ind,2) Y(ind,2);X(ind,3) ...
            Y(ind,3); X(ind,4) Y(ind,4)];

    [ePosition,Eresidual,Covx,iCount,debug] = lbl_ls_method(ranges, ...
            beacons,poseInitial);

    Position(ind,:) = ePosition;
    Residual(ind) = Eresidual;
    if Residual(ind) < Accepted_Residual
        poseInitial = Position(ind,:);
        PC = PC + 1;
        TTime(PC) = time(ind);
    end
end;

fprintf(1,'Fix Count is %d.\n',PC-1)
```

```
figure;
plot(time,Residual)
xlabel('Time, s');
ylabel('VLBL Least Squares Residual');

% Plot the result of the vehicle location
if S == 100;
    figure;
    plot(state.x,state.y,'r:',PosX,PosY,'b-x',xpx,xpy,'ko');
    axis([8500 9300 4750 5300]);
    title('VLBL for Simulated Data without Dead Reckoning and ...
           without Outlier Rejection');
    xlabel('Easting, m');
    ylabel('Northing, m');
    legend('Actual Vehicle Track','VLBL Track','Transponder Location')
end;

% Plot the result of the vehicle location for real-world data
if S ~= 100;
    figure;
    plot(StateX,StateY,'r:',lbl_SB.x,lbl_SB.y,'cx',PosX,PosY, ...
           'b-x',xpx,xpy,'ko');
    title('VLBL for Real-World Data without Dead Reckoning and ...
           without Outlier Rejection');
    xlabel('Easting, m');
    ylabel('Northing, m');
    legend('Actual Vehicle Track','Traditional LBL Fixes', ...
           'VLBL Track','Transponder Location')
end

% Calculate and plot the error budget
for pc = 1:PC-1
    Ind = find(abs(time - TTime(pc)) == min(abs(time - TTime(pc))));
    Error(pc) = ((PosX(pc) - stateX(Ind))^2 + (PosY(pc) - ...
           stateY(Ind))^2) ^ 0.5;
    DistTrav(pc) = D_Trav(Ind);
    Percent_Error(pc) = Error(pc) * 100 / D_Trav(Ind);
end

figure;
plot(DistTrav(5:end),Error(5:end));
title('Error Budget');
xlabel('Distance Traveled, m');
ylabel('Error, m');

figure;
plot(DistTrav(5:end),Percent_Error(5:end));
title('Error Budget as a Percentage of Distance Traveled, m');
xlabel('Distance Traveled, m');
ylabel('Error, % of Distance Traveled');

fprintf(1,'Done.\n');
```

Appendix C: Expanded VLBL Algorithm

```
% Load the correct data file, update FileLoad.m to the correct dive
% Start and End Time Indices refer to the state structure

S = input('Choose which dive to use (162 or 163) or 100 for test
      data:');
if S == 162;
    FileLoad162;
    Start_Time_Index = 15000;
    End_Time_Index = 33000;
elseif S == 163;
    FileLoad163;
    Start_Time_Index = 22000;
    End_Time_Index = 32000;
elseif S == 100;
    FileLoadTest2;
    Start_Time_Index = 1;
    End_Time_Index = 2001;
else
    fprintf('Invalid choice.\n');
    return;
end;

% Create a path to the dsl utility functions in the dslutils folder
path(path,'./dslutils');

% Choose a transponder for the single beacon calculations
Beacon_Index = input('Choose which transponder to use (1 to 4):');

% Isolate the actual x,y,z position of the chosen transponder
ATx = xp_x(Beacon_Index);
ATy = xp_y(Beacon_Index);
ATz = xp_z(Beacon_Index);

% Preprocess data in the sensor_data_preprocess function
fprintf(1,'Preprocessing... ');

% Find the indices of the analysis subset in the state.t structure
Interval.Trange = state.t([Start_Time_Index End_Time_Index]);
% Extract only data in the subset interval from lbl & state structures
% Make sure state comes first even after accounting for LBL cycle delay
Interval.lbl = sextt(lbl,Interval.Trange + [10 0]');
Interval.state = sextt(state,Interval.Trange);

% Find good ranges to the chosen beacon and the corresponding Unix
% times corrected for the 7 second navigation cycle.
index     = find(Interval.lbl.status(:,Beacon_Index) == 5);
range     = Interval.lbl.r(index,Beacon_Index); % m
RangeTime = Interval.lbl.t(index)-7 + range/1500; % s
RL        = length(RangeTime);
```

```
% Extract the fixes from the lbl structure for comparison to VLBL fixes
lblS.x = Interval.lbl.x;
lblS.y = Interval.lbl.y;
lblS.t = Interval.lbl.t;

% Create a master time series based on the time series of the state
% data which will be used for each dead-reckoning update.
MasterTime = Interval.state.t;
L = length(MasterTime);

% Extract the necessary parameters for this interval
Depth    = Interval.state.depth(:);       % Vehicle depth, m
Heading  = Interval.state.hdg(:);         % Vehicle hdg, 0:2pi rad
StateX   = Interval.state.x(:);      % Vehicle X position- comparison
StateY   = Interval.state.y(:);      % Vehicle Y position- comparison
     U   = Interval.state.u(:);      % Vehicle forward velocity, m/s
     V   = Interval.state.v(:);      % Vehicle starboard velocity, m/s

fprintf(1,'Done.\n');

% Transform the vehicle's velocity vectors from local (body oriented)
% coordinates to global coordinates, and plot them versus time
fprintf(1,'Transforming velocities into global reference frame...');

VelX  = U.*sin(Heading) + V.*cos(Heading);   % Vehicle east vel, m/s
VelY  = U.*cos(Heading) - V.*sin(Heading);   % Vehicle north vel, m/s

fprintf(1,'Done.\n');

% Choose a sampling interval for the range data for creating the VLBL
I_skip = input('Choose I_skip to use:');

fprintf(1,'Starting the dead reckoning process...\n');
EPosX(1) = state.x(Start_Time_Index);
EPosY(1) = state.y(Start_Time_Index);
poseInitial = [EPosX(1) EPosY(1)];

% Choose an acceptance level for residuals
Accepted_Residual = input('Choose acceptable LS residual (e.g. 50):');

% Choose an outlier rejection factor
OR_factor = input('Choose outlier rejection factor (e.g. 1.5):');

% Initialize the following parameters
DiffTolerance = 0.55 * mean(diff(MasterTime));  % Used 55% of mean diff
D_Trav = 0;             % Distance Traveled
I = 0;                  % # of RangesTimes lining up with MasterTimes
VC = 0;                 % VLBL Net Generation Count
PC = 1;                 % VLBL Position Count
FC = 0;                 % VLBL Fix Count = PC - 1
```

```
% Start the dead-reckoning and VLBL process
for ii = 2:L;
    % The old position is the vehicle position from the previous time
            step
    x_old(ii) = EPosX(ii-1);
    y_old(ii) = EPosY(ii-1);

    % The time difference is from the previous to the current time step
    dt(ii) = MasterTime(ii) - MasterTime(ii-1);

    % The vehicle velocity used for the DR is from the previous time
            step
    SpeedX(ii) = VelX(ii-1);
    SpeedY(ii) = VelY(ii-1);

    % The distance traveled since the previous time step is velocity *
            time
    DeltaX(ii) = SpeedX(ii) * dt(ii);
    DeltaY(ii) = SpeedY(ii) * dt(ii);

    % Calculate total distance traveled thus far
    D_Trav(ii) = D_Trav(ii-1) + (DeltaX(ii)^2 + DeltaY(ii)^2)^0.5;

    % The new DR position estimate is the old position + distance
            traveled
    x_new(ii) = x_old(ii) + DeltaX(ii);
    y_new(ii) = y_old(ii) + DeltaY(ii);

    % If there is a beacon range available, update the position
    % estimate using a least squares solution with the VLBL geometry
    Indicator = [];
    Indicator = find((abs(MasterTime(ii) - RangeTime) <=
      DiffTolerance),1);

    % Note that Indicator is in reference to the RangeTime vector
    if ~isempty(Indicator)
        I = I + 1;
        Index(I) = ii;
        Range(I) = range(Indicator);

        % Once four ranges have been accumulated, begin building VLBL
        % geometry
        if (I >= 4 && mod(I,I_skip)==0);
            VC = VC + 1;

            % Calculate locations of the virtual transponders, TX1-TX4
            % Virtual Transponder 4 is the actual transponder location
            VTx(VC,4) = ATx;
            VTy(VC,4) = ATy;

            % Virtual Transponder 3 corresponds to the range taken at
            % Index(I-1).  The position is that of actual transponder +
```

```
% the entire DR track from time Index(I-1) to Index(I)-1
VTx(VC,3) = VTx(VC,4) + sum(DeltaX(Index(I-1):Index(I)-1));
VTy(VC,3) = VTy(VC,4) + sum(DeltaY(Index(I-1):Index(I)-1));

% Virtual Transponder 2 corresponds to the range taken at
% Index(I-2).  The position is that of actual transponder +
% the entire DR track from time Index(I-2) to Index(I-1)-1
VTx(VC,2) = VTx(VC,3) + sum(DeltaX(Index(I-2):Index(I-1)-
      1));
VTy(VC,2) = VTy(VC,3) + sum(DeltaY(Index(I-2):Index(I-1)-
      1));

% Virtual Transponder 1 corresponds to the range taken at
% Index(I-3).  The position is that of actual transponder +
% the entire DR track from time Index(I-3) to Index(I-2)-1
VTx(VC,1) = VTx(VC,2) + sum(DeltaX(Index(I-3):Index(I-2)-
      1));
VTy(VC,1) = VTy(VC,2) + sum(DeltaY(Index(I-3):Index(I-2)-
      1));

% Determine the four ranges, R1 to R4
R(VC,4) = Range(I);
R(VC,3) = Range(I-1);
R(VC,2) = Range(I-2);
R(VC,1) = Range(I-3);

% Least Squares solution for spherical positioning
ranges = R(VC,:);
beacons = [VTx(VC,1) VTy(VC,1); VTx(VC,2) VTy(VC,2); ...
    VTx(VC,3) VTy(VC,3); VTx(VC,4) VTy(VC,4)];

poseInitial = [x_new(ii) y_new(ii)];

[ePosition,Eresidual,Covx,iCount,debug] =
      lbl_ls_method(ranges,beacons,poseInitial);

Position(ii,:) = ePosition;
Residual(ii) = Eresidual;

% If the LS residual is small enough, reset the position
% on the least squares solution.  If it is not, then use
% estimated position from the DR track at this time step.
if Residual(ii) < Accepted_Residual
    if PC >= 2
        if abs(((Position(ii,1) - ATx)^2 + (Position(ii,2)
            - ATy)^2)^0.5 - R(VC,4))<=1 ...
               && abs(Position(ii,1) - PosX(PC)) <=
           abs(OR_factor * sum(DeltaX(PCIndex(PC) + 1:ii)))
               && abs(Position(ii,2)- PosY(PC)) <=
           abs(OR_factor * sum(DeltaY(PCIndex(PC) + 1:ii)))

        %(((Position(ii,1) - PosX(PC))^2 + (Position(ii,2)
        % - PosY(PC))^2)^0.5) <= OR_factor *
```

```
                    % ((SpeedX(PC)^2 + SpeedY(PC)^2)^0.5) * dt(ii)
                        poseInitial = Position(ii,:);
                        PC = PC + 1;
                        PosX(PC) = poseInitial(1);
                        PosY(PC) = poseInitial(2);
                        VTime(PC) = MasterTime(ii);
                        PCIndex(PC) = ii;
                        x_new(ii) = Position(ii,1);
                        y_new(ii) = Position(ii,2);
                        FC = FC + 1;
                        vlbl.x(FC) = Position(ii,1);
                        vlbl.y(FC) = Position(ii,2);
                        fprintf(1,'Updating DR track with VLBL fix
                              number %d.\n', FC);
                    end
                else
                    poseInitial = [x_new(ii) y_new(ii)];
                    PC = PC + 1;
                    PosX(PC) = poseInitial(1);
                    PosY(PC) = poseInitial(2);
                    PCIndex(PC) = ii;
                    x_new(ii) = Position(ii,1);
                    y_new(ii) = Position(ii,2);
                    fprintf(1,'Updating DR track with VLBL fix number
                        %d.\n', FC);
                end
            end
        end
    end
    EPosX(ii) = x_new(ii);
    EPosY(ii) = y_new(ii);
end

fprintf(1,'Number of VLBL transponder nets created is %d.\n', VC);
fprintf(1,'Number of VLBL fixes is %d.\n', FC);

fprintf(1,'Done.\n');

fprintf(1,'Creating plots...');

% Plot all four transponder positions
figure;
plot3(xp_x(1),xp_y(1),-xp_z(1),'ro',xp_x(2),xp_y(2),-xp_z(2),'rx',...
    xp_x(3),xp_y(3),-xp_z(3),'r+',xp_x(4),xp_y(4),-xp_z(4),'r*')
grid on
title('Transponder Positions');
xlabel('Easting, m');
ylabel('Northing, m');
zlabel('Depth, m');
legend('Transponder 1','Transponder 2','Transponder 3','Transponder
      4');
```

```
% Plot velocities
figure;
plot(MasterTime,U,'b-',MasterTime,V,'r:',MasterTime,VelX,'g-.',...
    MasterTime,VelY,'k--')
title('Vehicle Velocity Components');
xlabel('Time, sec');
ylabel('Vehicle Component Velocity, m/s');
legend('Forward','Sideways','X Direction','Y Direction');

% Plot the LS Residuals
figure;
plot(Residual)
xlabel('Time, s');
ylabel('VLBL Least Squares Residual');

% Plot the difference between successive horizontal ranges
Diff34 = R(:,4) - R(:,3);
Diff23 = R(:,3) - R(:,2);
Diff12 = R(:,2) - R(:,1);
[Diff34,Diff23,Diff12] = denan(Diff34,Diff23,Diff12);

figure;
subplot(211);
plot(R);
title('Horizontal Ranges');
ylabel('Ranges, m');
legend('VT1','VT2','VT3','VT4');
subplot(212);
plot([Diff34 Diff23 Diff12]);
title('Differences in Horizontal Range between Virtual Transponder
Positions')
xlabel('Time, sec');
ylabel('Offset, m');
legend('VT3 and VT4','VT2 and VT3','VT1 and
VT2','Location','NorthEast')

% Plot the result of the vehicle location for simulated data
if S == 100;
    figure;
    plot(StateX,StateY,'r:',EPosX,EPosY,'k-
      ',vlbl.x,vlbl.y,'kx',ATx,ATy,'ko');
    axis([8500 9300 4750 5300]);
    title('VLBL for Real-World Data with Dead Reckoning and Outlier
      Rejection');
    xlabel('Easting, m');
    ylabel('Northing, m');
    legend('Actual Vehicle Track','VLBL Track','VLBL Fixes',...
        'Transponder','Location','NorthEast');
end;
```

```
% Plot the result of the vehicle location for real-world data
if S ~= 100;
    figure;
    plot(StateX,StateY,'r:',lblS.x,lblS.y,'cx',EPosX,EPosY,'k-',...
        vlbl.x,vlbl.y,'kx',ATx,ATy,'ko');
    title('VLBL for Real-World Data with Dead Reckoning and Outlier
      Rejection');
    xlabel('Easting, m');
    ylabel('Northing, m');
    legend('Actual Vehicle Track','Traditional LBL Fixes','VLBL
      Track','VLBL Fixes','Transponder','Location','NorthEast');
end;

% Calculate and plot the error budget
for pc = 1:PC
    Ind = find(abs(MasterTime - VTime(pc)) == min(abs(MasterTime -
      VTime(pc))));
    Error(pc) = ((PosX(pc) - StateX(Ind))^2 + (PosY(pc) -
      StateY(Ind))^2) ^ 0.5;
    DistTrav(pc) = D_Trav(Ind);
    Percent_Error(pc) = Error(pc) * 100 / D_Trav(Ind);
end

figure;
plot(DistTrav(2:end),Error(2:end));
title('Error Budget');
xlabel('Distance Traveled, m');
ylabel('Error, m');
axis([DistTrav(2) DistTrav(end) 0 1000]);

figure;
plot(DistTrav(2:end),Percent_Error(2:end));
title('Error Budget as a Percentage of Distance Traveled, m');
xlabel('Distance Traveled, m');
ylabel('Error, % of Distance Traveled');
axis([DistTrav(2) DistTrav(end) 0 30]);

fprintf(1,'Done.\n');
% End.
```

Appendix D: Cost Analysis Data

Economic Analysis of Cost Savings Potential of VLBL and MVLBL
Based on R/V Atlantis Voyage 11-14, Straits of Juan de Fuca

Assumptions:

Number of Transponders	0	1	2	4
Number of Dives Per Dive Site	1	2	3	4
Number of Dive Sites Per Voyage	5	10	15	20

Real World Transponder Statistics:

Number of Transponders Per Data Point	3	1	1	2	1
Time to Deploy these Transponders, min	82	55	36	34	26
Time to Survey these Transponders, min	214	35	132	180	60
Time to Recover these Transponders, min	179	89	36	147	26

Time to Deploy a Single Transponder, min	29
Time to Survey a Single Transponder, min	78
Time to Recover a Single Transponder, min	60
Total Handing Time for a Single Transponder, min	**166**

Real World Cost Data for R/V Atlantis:

Provisional Day Rate of Atlantis, 2006, CY-06$	30466
Provisional Hour Rate of Atlantis, 2006, CY-06$	1269
Provisional Min Rate of Atlantis, 2006, CY-06$	21

Transponder Handling Time per Voyage, min

# of Dive Sites per Voyage/ # of Transponders	***0***	***1***	***2***	***3***	***4***
5	0	832	1664	2496	3328
10	0	1664	3328	4991	6655
15	0	2496	4991	7487	9983
20	0	3328	6655	9983	13310

Transponder Handling Time per Voyage, hr

# of Dive Sites per Voyage/ # of Transponders	***0***	***1***	***2***	***3***	***4***
5	0	14	28	42	55
10	0	28	55	83	111
15	0	42	83	125	166
20	0	55	111	166	222

Transponder Handling Cost per Voyage, CY-06$:

# of Dive Sites per Voyage/ # of Transponders	***0***	***1***	***2***	***3***	***4***
5	$0	$17,600	$35,200	$52,800	$70,400
10	$0	$35,200	$70,400	$105,600	$140,799
15	$0	$52,800	$105,600	$158,399	$211,199
20	$0	$70,400	$140,799	$211,199	$281,599

Cost Savings Potential, CY-06$

# of Transponders Reduced/ # of Dive Sites per Voyage	***5***	***10***	***15***	***20***
Reduction from 4 to 2 Transponders, VLBL w/Redundancy	$35,200	$70,400	$105,600	$140,799
Reduction from 4 to 1 Transponder, VLBL	$52,800	$105,600	$158,399	$211,199
Reduction from 4 to 0 Transponders, MVLBL	$70,400	$140,799	$211,199	$281,599