Long Baseline Ranging Acoustic Positioning System

Tejaswi Gode

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Electrical and Computer Engineering

Craig A. Woolsey, Chair
Daniel J. Stilwell
Alfred L. Wicks

February 20, 2015
Blacksburg, Virginia

Keywords: Long baseline ranging, Acoustic localization, Estimation theory, Kalman filter,
Unmanned underwater glider

Long Baseline Ranging Acoustic Positioning System

Tejaswi Gode

(ABSTRACT)

A long-baseline (LBL) underwater acoustic communication and localization system was developed for the Virginia Tech Underwater Glider (VTUG). Autonomous underwater vehicles, much like terrestrial and aerial robots require an effective positioning system, like GPS to perform a wide variety of guidance, navigation and control operations. Sea and freshwater attenuate electromagnetic waves (sea water is worse due to higher conductivity) within very few meters of striking the water surface. Since radio frequency communications are unavailable, many undersea systems use acoustic communications instead. Underwater acoustic communication is the technique of sending and receiving data below water. Underwater acoustic positioning is the technique of locating an underwater object. Among the various types of acoustic positioning systems, the LBL acoustic positioning method offers the highest accuracy for underwater vehicle navigation. A system consisting of three acoustic "beacons" which are placed on the surface of the water at known locations was developed. Using an acoustic modem to excite an acoustic transducer to send sound waves from an underwater glider, the range measurements to each of the beacons was calculated. These range measurements along with data from the attitude heading and reference system (AHRS) on board the glider were used to estimate the position of the underwater vehicle. Static and dynamic estimators were implemented. The system also allowed for underwater acoustic communication in the form of heartbeat messages from the glider, which were used to monitor the health of the vehicle.

This work received support from the Office of Naval Research (ONR), under Grant #N00014-13-1-0060

Acknowledgments

I would like to take this opportunity to acknowledge the people who have contributed to my success in graduate school. They have been pillars of support in enriching my intellect and providing me with much needed emotional backing. Most importantly, I would like to thank my adviser Dr. Craig Woolsey, for being a great mentor and providing assistance every step of the way. From the first day I met him he has trusted me unconditionally, and believed in me to complete the required tasks. Coming from a different technical background, and an academic culture that is poles apart from the American one, I encountered innumerable challenges at work, mentally and emotionally. The trust and faith put in me was the biggest driving force to overcome these challenges. His professionalism, and an acute sense of what needs to be done have added tremendous value to my work ethic. At times when we hit a wall, a meeting with him would give us much needed direction and clarity. By observing his approach to problems I have picked up some important life lessons. I feel very lucky to be his mentee, and forever grateful for the opportunity to work with him.

I would also like to especially thank Artur Wolek. He gets the triple honor of being a mentor, a friend and most importantly a "partner in crime" in the Glider project. Seeing him go about his PhD has been an inspiration, and challenged me to be the best I can. He constantly guided me in my thesis as well as course work and taught me the best practices to endure graduate school. I have been able to pick up many skills in a short amount of time

due to his guidance and expertise. He helped in designing the LBL system and planned all our test activities. The "Virginia Tech Underwater Glider" for which this system was built was designed by him. His leadership has been instrumental in the success of this project. A significant factor which make these past two years memorable is the fact that work was always fun and at most times felt effortless. Those late nights at the lab fixing the glider for a test the next day, and celebrations at the Cellar will remain precious memories for life. More than anything I'm glad to say that I made a friend and that this project will always be very close to our hearts.

Also a shout out to my best friend since middle school, Riddhi, for being the beacon of light at the end of every metaphorical tunnel. She has been at the receiving end of all my whining and complaining for over a decade now, and has always been there to show me the way forward. Her love and trust has been unconditional and unwavering. Without her emotional support I would not be here today.

A huge acknowledgment to my Dad, who has been the go-to guy to make all the important decisions in my life. His ideals and philosophy have transformed my outlook of life, and he has been my spiritual mentor. My Mom, who is an embodiment of love and courage. Her blind faith in me has led me to believe in myself, and been the source of my confidence. My sister, who is also my dearest friend, for always rooting for me.

I would also like to thank Jake Quenzer for assisting with the software updates and design, as well as Jeremy Rohn for assisting in testing and fabrication.

Lastly I would like to thank all the members of NSL who supported us in the Glider project and otherwise, especially Ony Arifianto, Dave Grymin, Chris Kevorkian, Mark Palframan, Hossam Abdelgaffar and Deva Prakash.

Dedication

To my grandmother Jayamma, and my grandfather Parameshwarappa

Contents

List of Figures

List of Tables

Nomenclature

A/D:	Analog to Digital
ACOMMS:	Acoustic Communications
ACS:	Attitude Control System
AHRS:	Attitude Heading and Reference System
AUV:	Autonomous Underwater Vehicle
BCS:	Buoyancy Control System
BER:	Bit error rate
CAD:	Computer Aided Design
COMMS:	Communications
DGPS:	Differential Global Positioning System
DR:	Dead Reckoning
DVL:	Doppler Velocity Log
EKF:	Extended Kalman Filter

EMB: Electronics Main Bay

FSK: Frequency Shift Keying

GIB: GPS Intelligent Bouy

GPS: Global Positioning System

INS: Inertial Navigation System

LBL: Long Baseline

LED: Light Emitting Diode

LS: Least Squares

NMEA: National Marine Electronics Association

OWTT: One way travel time

PPS: Pulse Per Second

PSK: Phase Shift Keying

PVC: Polyvinyl Chloride

RF: Radio frequency

ROS: Robot Operating System

RPC: Remote Procedure Call

RTS: Rauch-Tung-Striebel

RTT: Round Trip Travel Time

SBL: Short Baseline

SPURV: Special Purposed Underwater Research Vehicle

TAT: Turn Around Time

TDOA: Time Difference Of Arrival

TOA: Time Of Arrival

TOF: Time Of Flight

UML: Unified Modeling Language

USBL: Ultra Short Baseline

UWSN: Underwater sensor network

VTUG: Virginia Tech Underwater Glider

WHOI: Woods Hole Oceanographic Institute

Chapter 1

Introduction

This thesis addresses the general problem of estimating the position of an underwater moving object, carrying an acoustic transducer, by measuring the one way travel times (OWTTs) of the acoustic signals from a set of surface buoys equipped with submerged acoustic transducers and GPS receivers. When compared with more conventional systems [2], this class of underwater acoustic positioning systems is far more effective, portable, and cheaper to implement and operate.

Example of an underwater sensor network (UWSNs) deployed in oceans or other water bodies consists of acoustic transducers/transponders equipped with acoustic modems that enable them to communicate wirelessly (using sound) with each other. These sensor networks are deployed on the seabed or in the water column (like in the case of this project), and they cooperate in the sensing task and send their data for real time processing to a central agent. In case of the Long Baseline (LBL) positioning system developed for the Virginia Tech Underwater Glider (VTUG), the mobile glider acts as the central agent, processing data from the static sensors, to estimate its own position, with respect to a local inertial frame. Acoustic communications is used instead of Radio frequency (RF) because RF signals

($> 1Mhz$) cannot travel beyond a few meters through the water due to severe attenuation. Underwater acoustic channels are characterized by harsh physical environments, where the available bandwidth for communications is limited, channels are impaired due to multi-path and fading problems, there is motion induced Doppler shift, high propagation delay, and high bit-error rate (BER). Sound waves travel five orders of magnitude lower than light waves, and consequently, the propagation delay is very significant and has high variance. However, acoustic communication remains the most effective mode of communication underwater.

Localization is the problem of determining location. In the specific case of LBL ranging, the object whose location is to be determined is an Autonomous Underwater Vehicle (AUV) equipped with an acoustic transducer. However, it is critical for the stationary surface nodes/buoys mentioned above, which act as anchored reference nodes, to know their own location. This is accomplished by equipping them with GPS or DGPS for higher accuracy [3]. With GPS information, range data from the nodes can be interpreted meaningfully to calculate the location of the AUV. Research and development in this area have progressed to a point where commercial products called GPS Intelligent Buoys (GIBs) are available in the market [4],[5]. The terms node, beacon, and buoy are used interchangeably in the literature in reference to fixed sensors in a communication and positioning network. This setup is analogous to the GPS used in terrestrial applications and derives its design directly from it. In fact, one of the static estimation algorithms used in this thesis [6] was originally developed for the GPS, however is modified for UWSNs. The GIBs which are equipped with acoustic transducers act like satellites in the GPS constellation, providing independent measurements from which position may be triangulated.

1.1 Thesis Outline

This section presents a road map to the reader to aid in navigating through the document. The first part of the thesis (Chapters 2,3, and 4) presents theoretical concepts in underwater vehicle localization. The second part (Chapters 4, 5 and 6) presents an overview of an LBL positioning system developed for the VTUG, and results from an experiment conducted at Claytor Lake, VA.

Chapter 2 provides a brief overview and background of the various methods for underwater vehicle localization with emphasis on Long Baseline acoustic localization.

Chapter 3 presents two static estimation algorithms for 3D localization of an AUV. The first is a simple least squares estimator and the second is called the Bancroft algorithm, which provides a direct algebraic solution to the LBL localization problem. Some Matlab simulations, implementing these algorithms are also shown.

Chapter 4 explains the theory behind dynamic estimation algorithms, where a dynamic model is used to describe the movement of the AUV. Since the vehicle model is nonlinear, the Extended Kalman Filter (EKF) is used to estimate the vehicle position. The RTS (Rauch–Tung–Striebel) smoothing algorithm, which further improves the EKF estimate is also described. Finally, Matlab simulations which compare static and dynamic position algorithms are presented.

Chapter 5 illustrates the experimental setup to perform LBL ranging. The mechanical, electrical and software design are explained in fair detail. A section is dedicated to the principle and working of the WHOI acoustic modems and transducers which are the pivotal components in implementing underwater localization.

Chapter 6 presents results of an experiment conducted at Claytor Lake, VA, in November

2014 with the experimental setup described in Chapter 5.

Chapter 7 concludes the thesis by highlighting the contributions of the author, and some possibilities for future work.

Chapter 2

Background and Motivation

2.1 A brief review of Underwater Vehicle Navigation

Autonomous underwater vehicles have been in development since the late 1950's. There are different types of AUV's based on their size, endurance, propulsion system, and the kind of energy used to power them. Propeller based thrusters are the most common propulsion systems used. Underwater gliders, such as the VTUG however, do not directly propel themselves. As seen later in Section 5.1, they use a buoyancy control system to repeatedly descend and ascend.

The first AUV ever developed was the SPURV (Special Purposed Underwater Research Vehicle) in 1957, at the University of Washington's Applied Physics Laboratory [7]. It was a self-propelled underwater research vehicle with a depth rating of 3000m.The first autonomous underwater glider which is still being manufactured is the *Slocum*, whose design was proposed by Henry Stommel in 1989. The design of the VTUG is based on legacy gliders such as the *Seaglider*, *Spray* and *Slocum*. However, a key difference in the VTUG, is the design of the

buoyancy control system which uses compressed air instead of oil. This makes the VTUG highly maneuverable, but limits the depth to which it can operate (rated to 100m) [1]. A detailed review of underwater glider systems can be found in [8].

During the last few decades, a growing interest in UWSNs and AUVs has been observed. Together they can be used for a multitude of underwater applications including: oceanographic data collection, warning systems for natural disasters (Eg. seismic and tsunami monitoring), ecological applications (Eg. pollution, water quality and biological monitoring), military underwater surveillance, assisted navigation, industrial applications (offshore exploration of crude oil), etc. For example, the highly publicized search and rescue effort for the MH370 (Malaysian Airlines) Boeing-777 aircraft, that is presumed to have crashed in the middle of the Indian ocean on 8 March 2014, used the Bluefin-21 AUV along with a wide range of UWSNs. As of 17 December 2014 a total of $11,000\text{km}^2$ of the seafloor has been searched, and the bathymetry survey mapped over $200,000\text{km}^2$ of seafloor.

A large number of review papers have been written over the last decade detailing the different types of UWSNs and localization schemes used for underwater vehicle navigation. [9, 10, 11, 12, 13, 14, 15, 16, 17, 18] and the references therein provide a detailed survey, and challenges of underwater vehicle navigation.

LaPointe et al. [19] describe the main types of navigation systems for underwater vehicles, they are:

1. **Inertial navigation systems:** Inertial navigation uses accelerometers and gyroscopes for increased accuracy to propagate the current state. Nevertheless, all of the methods in this category have position error growth that is unbounded.

2. **External acoustic systems:** Techniques in this category are based on measuring the time-of-flight (TOF) of signals from acoustic beacons or modems to perform navigation.

3. **Geophysical navigation:** Techniques in this category use external environmental information as references for navigation. This is done with sensors and processing that are capable of detecting, identifying, and classifying some environmental features.

2.2 Inertial Navigation Systems

The simplest navigational technique for underwater vehicles is that of Dead Reckoning (DR) using Inertial Navigation Systems (INS). Dead reckoning is defined as a process of positioning in which an AUV positions itself autonomously with no support from a ship or external acoustic networks [20]. In such systems only on-board sensors are used to determine the position of the vehicle and no external references are used. Before diving underwater, the vehicle position is initialized using GPS or a predefined local inertial frame. Throughout the duration of the vehicle's underwater activity on-board sensors are used to repeatedly update its position estimate since GPS signals are heavily attenuated once underwater. The basic principle of inertial navigation is the ability to measure the acceleration of an underwater vehicle using inertial measurement units mounted inside them. Given the ability to measure acceleration, the change in velocity and position can be calculated by performing successive integrations of the acceleration with respect to time.

Along, with the ability to measure acceleration, it is necessary to also keep track of direction or heading of the AUV. Only with the information of heading can the accelerations be resolved into a pre-defined local inertial frame. An inertial measurement unit (IMU) usually consists of two types of sensors:

1. **Accelerometers:** Sensors used for measuring acceleration.

2. **Gyroscopes:** Sensors used for measuring rotation.

Together, the above cluster of sensors can measure and maintain an estimate of position, velocity, and attitude rated of an underwater vehicle. IMUs used in today's AUVs incorporate the accelerometers and gyroscopes into one compact, low cost device that is strapped down to the body. Kinsey et al. [12] provide a detailed analysis of different types of sensors used in modern INS systems and their evolution.

The measurements provided by the IMUs are in an inertial reference frame defined for the device itself. Based on the orientation in which it is mounted, it's inertial frame might not coincide with the body frame of the vehicle. In such cases, the IMU's readings are fed to the on-board computer which performs the necessary calculations (usually involving rotation matrices) to resolve it into the vehicle's body frame. This calculation could be a source of error, hence it is necessary to precisely measure the orientation of the IMU with respect to the host vehicle.

Originally, dead reckoning was done by estimating vehicle speed *a priori* by doing controlled experiments to calibrate propeller speed with different water speeds, and vehicle heading was calculated using magnetic compasses. The estimated speed was found to be erroneous due to the inherent nature of calibration, and magnetic compasses were prone to interference from ocean bottom and on-board senors which contained metallic parts [19]. To overcome these challenges, Doppler Velocity Log (DVL) units and gyro-compasses are used in modern INS systems. DVL units provide an estimate of an AUV's velocity relative to the ocean floor. They use downward facing transducers that sense Doppler shifts, which are then used to estimate the vehicle velocity. A gyro-compass is an electrically powered compass capable of finding true north while being impervious to external magnetic fields. This is accomplished by exploiting the rotation of the earth [21, 22]

Apart from the quality of the accelerometers and gyrocompasses used, the accuracy of INS systems is also affected by the type of algorithm used for state estimation [20]. Typically

state estimation is done using an Extended Kalman Filter (EKF). When the assumption of Gaussian noise does not hold, an Unscented Kalman Filter (UKF) can be used, though it is computationally more expensive.

As stated earlier, INS systems primarily have the ability to measure acceleration using accelerometers. The acceleration measured is integrated once to obtain velocity and twice to obtain position. The position is relative to the initial position fed to the AUV prior to diving. For this process of numerical integration, various approximations are used. Hence, any error in the measurement of acceleration is compounded with each integration. Since there is no reinitialization during a dive the position estimate slowly drifts over time and distance. The drift rates for the best (most expensive) INS systems are 0.1% of the amount of distance traveled and for typical ones it can be anywhere between 2-5% of the distance [20]. For marine operations which are conducted in shallower waters with less distance traveled, and where the vehicle frequently resurfaces to reinitialize the INS with GPS position, such systems can be useful. However, for long endurance missions INS systems are coupled with external acoustic systems for better performance. Morgado et al. [23] describe an integration technique for error estimation in low cost INS systems coupled with an USBL type external acoustic system, and Whitcomb et al. [24] give an example of an INS system integrated with an external acoustic LBL system which will be described in the next section.

2.3 External Acoustic Systems

In external acoustic systems, localization is achieved by measuring ranges from the TOF of acoustic signals. These methods started gaining widespread usage to overcome the challenges of dead reckoning or INS systems. They are typically integrated with these systems to improve accuracy for long duration missions at higher depths.

External acoustic positioning systems are used by underwater vehicles to laterate/angulate their position based on range/bearing information, between external acoustic transmitters, and one (or an array) mounted on the vehicle. External acoustic systems use 5 main types of transmitters/receivers (adapted from [2]).

1. **Transducer:** Sends out an interrogation/ping on one frequency and receives a response on another. These frequencies are closely spaced (generally 0.5kHz apart).
2. **Transponder:** Receives an interrogation on one frequency and sends a response on another.
3. **Beacon/pinger:** A free running device which continuously sends out a pulse on a particular frequency.
4. **Hydrophone:** A listening device which can receive responses from a transponder or a beacon.
5. **Responder:** A transmitter which is externally triggered, to transmit an interrogation signal to a transponder or hydrophone.

The above terminology is based on functionality. The same hardware can be configured to perform different acoustic tasks. Three different primary geometries are used for external acoustic navigation systems: Short Baseline (SBL), Ultra Short Baseline (USBL), and Long Baseline (LBL). The distance between the active sensing elements (transmitters/receivers) is called the acoustic baseline. The length of this baseline is used to differentiate external acoustic systems.

External Acoustic System	Baseline Length
USBL	< 10cm
SBL	$20 - 50$m
LBL	100m $- 10$km

Table 2.1: Baseline Lengths

Some challenges associated with deploying, maintaining and using external acoustic systems are listed below [19, 9],

- **Coordinate system compatibility:** Deploying external acoustic systems involves the use of three different platforms. The orientation of all them must be measured and a local inertial frame must be setup.

    1. Acoustic transmitters/receivers: These are the acoustic sensors of any type described above. They can be deployed in the form of buoys on the seabed, in the water column, or on the hull of the host vessel. Their orientation must be measured.
    2. AUV: This is the heading and pose of the AUV which must be precisely measured. Can be done using an IMU described in Section 2.2.
    3. Host vessel: Underwater missions involving AUVs are generally centered around a ship where all the data analysis from the AUVs as well acoustic sensors is done. It's orientation and position must also be measured. GPS can be used to measure its position. This position is used to set up a local inertial frame. The position of the acoustic sensors is calibrated using the ship's position. Hence high accuracy of the ship's position is critical to ensure the external acoustic system provides a reliable estimate of the AUVs position.

- **Outlier rejection:** The underwater environment is very noisy for acoustic operations

(refer [9] for the types of noise). The acoustic TOAs measured by acoustic sensors are affected by this noise. Also, acoustic transmissions are affected by multipath scenarios due to various rock formations as well as regions in-conducive to acoustic transmissions called shadow zones [2]. Hence the AUV receives many inaccurate measurements of TOAs as well as some dropouts. These measurements could be confused with accurate ones and are called outliers. The AUV must be equipped with algorithms which can reject them. Vaganay et al. [25] provide a method for outlier rejection.

2.3.1 Short Baseline Navigation

SBL systems, as shown in Figure 2.1 are characterized by an acoustic baseline length of 20-50m. In these systems a set of acoustic transponders are mounted on the underside of a ship, boat or any sufficiently large floating device. They use TOAs of acoustic signals from each of these to a transducer on the AUV to compute position using trilateration. The accuracy of the computation is largely dependent on the length of the baseline. Generally, the farther apart the transducers, the better the accuracy. The position is calculated relative to the GPS position of the host vessel and requires precise calibration of the transducer array. Depth sensors on-board the AUV are used in conjunction with the acoustic travel times to obtain 3D position information.

SBL systems require the AUV to operate in the close vicinity of the ship and have a lesser range than LBL systems which will be described later. However, they are easier to deploy and operate, and are favored for research operations centered around a host vessel [18].

Nodland et. al [7] describe a SBL system used to track SPURV II. It was a modified SPURV with improved platform stability and higher frequency bandwidths for its sensors, and was used for various underwater research operations until 1979. The tracking and command

system consisted of various CRT displays which helped an operator to maneuver it on-board a nearby ship. The localization system consisted of a triangular array of hydrophones mounted beneath the ship's hull. Frequency shift keying (FSK) modulation technique was used to carry commands to SPURV and vehicle status back to the operator. It will be later discussed that this technique is used in the WHOI (Woods Hole Oceanographic Institute) acoustic modems used for LBL localization developed in this project. References [23], [26] are some other examples of applications of the SBL system.

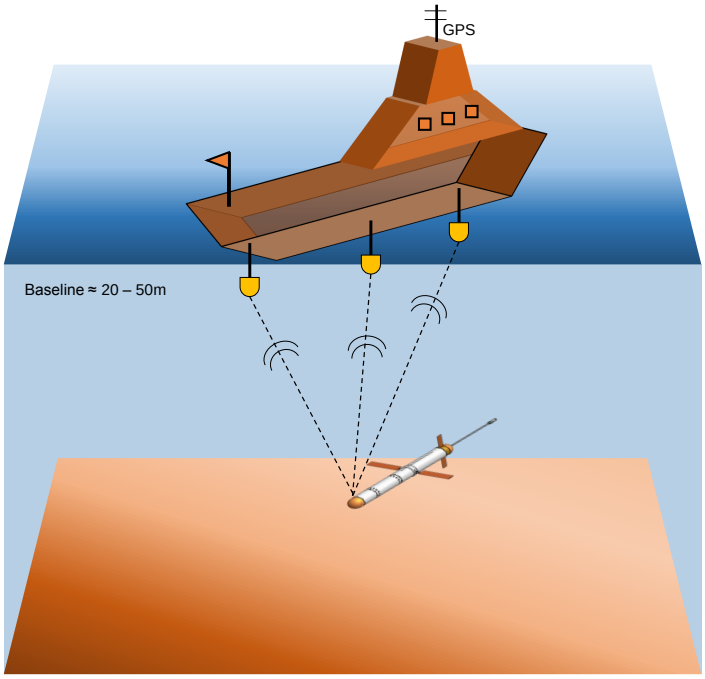


Figure 2.1: Short Baseline Navigation System

2.3.2 Ultra-Short Baseline Navigation

USBL systems as shown in Figure 2.2 consist of a transducer array, with the transducers placed $\sim$ 10cm apart. Thus the name ultra-short baseline. There are generally at least 3 transducers in this array and is called the transceiver. A transducer on the target AUV

acoustically pings the transceiver and receives a response. The travel time is used to calculate the distance to the transceiver. However, just the distance information is not enough to estimate the AUV position in an inertial frame. The angle or direction information is also required. To accomplish this a technique called phase-differencing is used. In this technique the angles of arrival of the acoustic pulse from the AUV to each transducer in the transceiver are used to discern the direction [18]. Hence, the distance and direction are together used to calculate the AUV position. Similar to SBL the host vessel's GPS is used as a reference and the area of operation is limited to it's vicinity.

USBL systems are easier to deploy than SBL systems and are the most widely used. However, since the baseline is small, they are less accurate than SBL or LBL systems. [23] is an example of USBL system.

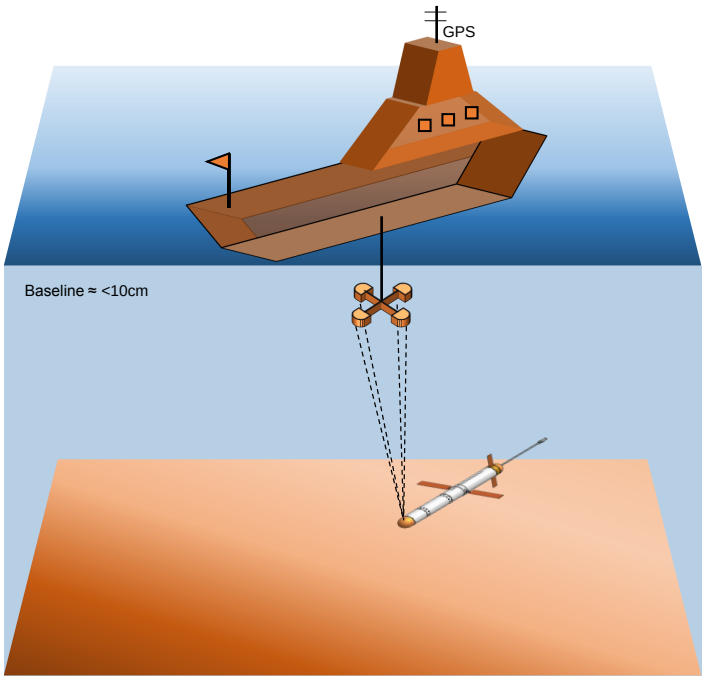


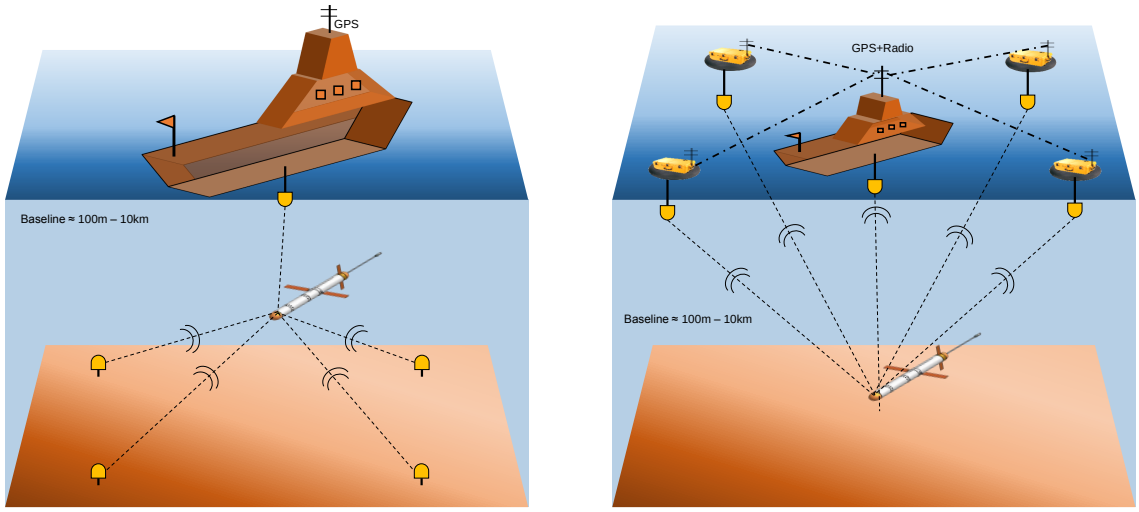
Figure 2.2: Ultra Short Baseline Positioning System

2.3.3 Long Baseline Navigation

Long-baseline (LBL) positioning refers to the determination of position via interrogation of two or more fixed transducers separated by large distances (long baselines). LBL systems can be broadly classified into two types, based on where the fixed/static transducers are deployed. Figure 2.3 illustrates these two types.

1. Classical LBL systems, where the transponders are moored to the seabed and fixed to the bottom.

2. Mobile LBL systems, where buoys with GPS (GIBs) are deployed on the water surface with the transponders submerged in the water at a predetermined depth. The LBL system developed for VTUG is of this type.



(a) Classical LBL (b) Mobile LBL (GIBs)

Figure 2.3: Long Baseline Positioning Systems

Classical LBL systems are similar to SBL systems with the topology inverted. These systems require a precise knowledge of the bathymetry of the area of operation. The transducers

are deployed at the ocean floor and their topology needs to be accurately calibrated. The distances between the transducers, their depth information and GPS position of the host vessel is used to set up a local inertial frame using which the target AUV position is estimated. This process of calibration is an important source of error and if done improperly deeply impacts the accuracy of the system.

Classical LBL systems require lot of time and effort to successfully implement and are not portable. Also, at higher depths (¿200m) divers are unable to mount the transducers on the seafloor. In such situations ROVs are required to set up the acoustic net which further increases the cost and complexity of implementation. To overcome these difficulties mobile systems were developed.

In mobile LBL systems, the beacons/buoys (GIBs) float on the water surface with the transponders submerged in the water. They are generally anchored to the ocean floor to restrict their movements due to currents. The transponders can be submerged at a predetermined depth. GIBs remove the need for the LBL beacons to be installed at the seafloor, which can reduce costs. Since the GIBs are also equipped with GPS, it becomes easier to establish a local inertial frame.

The process of localization is similar to SBL systems. The AUV transducer pings each of the transducers in the LBL system in a round robin fashion. The travel times from the response signals are used to estimate its position. Static or dynamic estimation algorithms are available to accomplish this. Some of them are discussed in Chapters 3 and 4. The accuracy of the travel times depends on a lot of factors. Most important of these are the baseline length, and frequency of interrogation. See Table 2.2 [9], for the different frequency ranges, the corresponding maximum ranges, and static accuracy.

	Frequency Range	**Maximum range**	**Typical relative, static accuracy***
Low frequency (LF)	8kHz to 16kHz	$\backsim$ 10km	2m to 5m
Medium frequency (MF)	18kHz to 36kHz	2km to 3.5km	0.25m to 1m
High frequency (HF)	30kHz to 60kHz	$1,500$m	0.15m to 0.25m
Extra high frequency (EHF)	50kHz to 110kHz	$< 1,000$m	< 0.05m
Very high frequency (VHF)	200kHz to 300kHz	< 100m	< 0.01m

Table 2.2: LBL positioning accuracy versus frequency

*Static accuracy (multiple acoustic observations in the same place) is often quoted as being achievable for dynamic moving objects (position update every 3-4 seconds), which is rarely the case.

The lower the frequency range used for interrogation, the higher the effective range, and lower the accuracy. The trade-off between frequency and accuracy is crucial for the design of an LBL system. The decision on using a particular frequency band is made based on the application at hand. The depth of operation is central to this decision. The deeper the area of operation the lower the frequency band used. The simple reason being higher frequencies are attenuated more than lower ones. Chapter 5 is dedicated to describing in detail the AUV and LBL system developed as part of this project.

2.4 Geophysical Navigation

Geophysical navigation includes a range of techniques for underwater localization and navigation using the underwater terrain and surrounding environmental features as references. The basic principle in these techniques is to use observable physical features to estimate an AUV's position. Physical features that can be used include (but not limited to) tidal inlets, hydrothermal vents, special rock formations, geomagnetic data, bathymetry data etc. A map

of the area of the mission can be created using such physical features. Such a topological map can be constructed prior to an AUV's mission and supplied to it, or generated while a mission is underway using SLAM (simultaneous localization and mapping) techniques. Stutters et al. [16] give an account of various geophysical navigation techniques that have been implemented using SLAM. Similar to the argument of combining external acoustic systems with INS to improve accuracy, geophysical techniques can also be combined with INS/DR systems.

The various geophysical navigation techniques can be divided into three categories [20],

1. Magnetic
2. Optical
3. Sonar

Since the description of geophysical navigation techniques is beyond the scope of this thesis, the author recommends referring to Paull et al. [20] and the references therein for a thorough review. Implementing geophysical navigation techniques requires precise knowledge of the underwater terrain and expensive on-board sensors. SLAM algorithms are computationally extensive, thus requiring high-speed processors to perform real time. Many of the geophysical navigation techniques have yet to be well tested, however, they have been shown to be the way forward in underwater localization. In the near future they could be capable of overcoming the disadvantages of using external acoustic systems and be reliable enough for wide spread use. However, for a low cost AUV such as the VTUG, external acoustic systems (LBL) are the primary means of localization.

Chapter 3

Static Position Estimation

3.1 Background

Static position algorithms do not account for the vehicle dynamics. As such they are simpler to implement, and computationally cheap. Most underwater static positioning algorithms are either spherical or hyperbolic. Urruela et. al [27], and Deffenbaugh et. al [28], give a detailed description of the performance of both these approaches. The algorithms discussed below are for multi-beacon LBL systems (> 2 beacons).

3.1.1 Spherical and Hyperbolic Positioning

In general, the localization can be based on *time of arrival* (TOA) measurements or *time-difference of arrival* (TDOA) measurements. TOA schemes fall under trilateration and TDOA schemes fall under multilateration localization techniques. TOA is the time it takes the signal to travel from the transmitter to the receiver. TDOA is the difference in time it takes signals from two different transmitters to reach the receiver. Spherical positioning

systems use TOAs and Hyperbolic positioning systems use TDOAs.

Spherical positioning systems determine position by measuring acoustic one-way-travel-times (OWTTs) or TOAs from beacons at known locations. With an available estimate of the sound-velocity profile in a particular underwater environment, these times can be converted to ranges. Each range measurement corresponds to the radius of a sphere centered at the location of the beacon. Range measurements from several beacons provide several spheres, the intersection of which is the position of the source. In Figure 3.1(a), spherical positioning is illustrated in a planar restricted scenario. The beacons are shown as colored dots arranged in a triangular geometry. The circles centered at each beacon have a radius equal to the range to source. The source must lie within this circle. The intersection of the three circles gives the location of the source. For the range measurements to be precise it is required that the source clock is synchronized with the clocks on the beacons. Note that, at least three beacons are required to obtain an unique *fix*. However, with two nodes, the fix is ambiguous and external information is used to choose between two possible solutions. Also, the slant range measurements have been projected onto the plane. In practical applications, this is possible with a depth sensor on-board the underwater vehicle. Hence, we have 3D position information. In Sections 3.3 and 3.4 two spherical positioning algorithms are explained in detail. These have been implemented for the LBL system developed for VTUG.

Hyperbolic positioning systems determine position by measuring *differences* in travel times between signals from the beacons. These differences in travel times is converted to differences in ranges using the sound-velocity profile. The differences in ranges defines a hyperboloid on which the receiver must lie. Several of these range differences provide several hyperboloids, the intersection of which is the position of the source. In Figure 3.1(b), the intersection of the hyperbolae gives the position of the source, which is the black diamond. These hyperbolae are the loci of TDOAs from the red and green beacons, and red and blue beacons.

For consistency the same topology is used for spherical and hyperbolic positioning. The advantage of hyperbolic system is that it does not use the time at which the beacons transmit in its calculation, and hence the source does not need to be synchronized with the beacons. The travel time differences can be found as long as the beacons transmit at the same time, or with known delays relative to each other. Hyperbolic positioning is not implemented as part of this project due to the availability of precise TOA measurements, made possible by using WHOI acoustic modems.

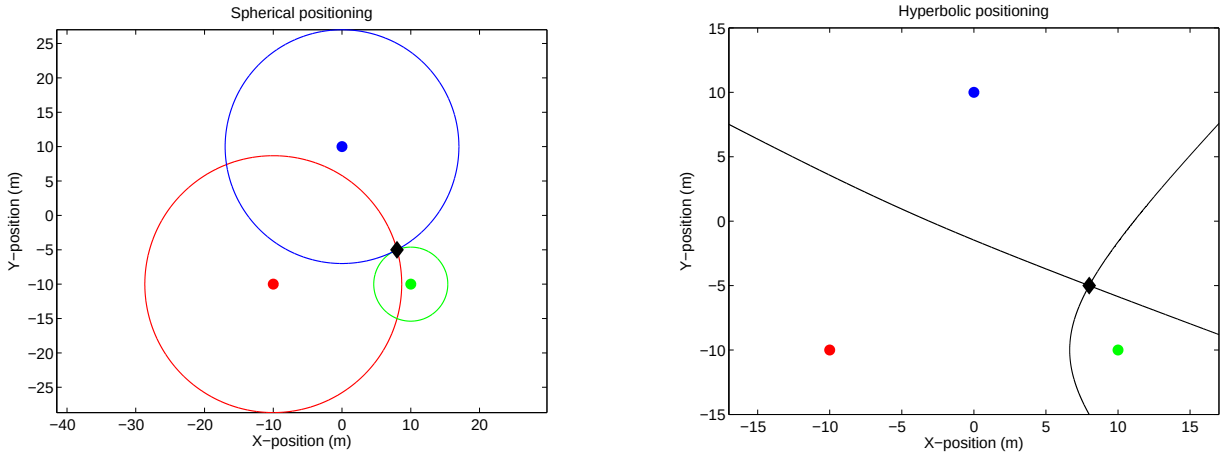


Figure 3.1: Spherical(a) and Hyperbolic(b) positioning

3.1.2 Acoustic signal travel time and range

The measured *round-trip* travel time(t_{RTT}) of an acoustic signal is given by the equation [29],

$$t_{RTT} = t_{TAT} + \frac{2R^{3D}}{c} + \varepsilon, \tag{3.1}$$

Where, t_{TAT} is the turn-around time, which is the amount of time electronics needs to detect

the signal and send a response, $R^{3D}$is the slant range from a beacon to the vehicle, c is the speed of sound in water and ε is the error between true and measured round-trip travel time.

Clearly, the OWTT or TOA is given by half of the round-trip travel time.

$$t_{OWTT} = \frac{t_{RTT}}{2} \tag{3.2}$$

Hence, the slant range is given by,

$$R^{3D}_{slant} = t_{OWTT} \times c \tag{3.3}$$

3.2 Spherical Localization Using Iterated Nonlinear Least Squares

This section is dedicated to the method of least squares for a simple model of spherical localization. A least squares (LS) estimate minimizes the weighted sum of the squared range errors (difference between true and estimate). It can be studied from two points of view: the classical *batch-processing* approach, in which all the measurements are processed together at one time, and the *recursive processing* approach, in which measurements are processed only few (or even one) at a time. The estimation algorithm is adopted from Chapters 2,3, and 5 of [30], and can be referred to for a full treatment on the topic. The LS estimator implemented in this project is a *batch-processing* estimator. Given an initial condition the least squares algorithm is applied until a desired estimate (based on a stopping criterion) is achieved.

3.2.1 Posing the problem

The starting point for the method of least squares is the simple nonlinear model shown in Eq. (3.4) where $r(i)$ is the measurement vector, $f(\theta, i)$ is an explicitly defined function of the unknown parameter to be estimated(θ), $\hat{\theta}$ is the estimate of the unknown θ, and $v(i)$ is the measurement noise vector. Eq. (3.4) is later linearized to take the form of a *generic linear model* [30], used to derive the least squares estimator.

$$r(i) = f(\theta, i) + v(i) \tag{3.4}$$

The nonlinear estimator model is given by,

$$\hat{r}(i) = f(\hat{\theta}, i) \tag{3.5}$$

For the case of spherical localization $f(\theta, i)$, is defined for the measured ranges from N known locations with M degrees-of-freedom [31],

$$\hat{r}(i) = f(\hat{\theta}, i) = \left(\sum_{k=1}^{M} (x_k - b_{k,i})^2 \right)^{1/2} \tag{3.6}$$

where, $\hat{r}(i)$ are the range estimates, where $i = 1, ..., N$, x_k are the Cartesian location coordinates, where $k = 1, ..., M$, and $b_{k,i}$ are the beacon locations in Cartesian coordinates.

The planar spherical localization method as shown in Figure 3.1, attempts to find the best estimate of the vehicle's position based on three slant range measurements ($N = 3$) from the beacons. The glider is equipped with a depth sensor, hence its depth is known at each time instant. Also, the transducers are submerged mid-water and their depth can be measured

from the length of the cables during deployment. Hence, the difference in the depth between the glider and the transducers, $\delta z_U = z_g - z_t$, is known. With δz_U, one can calculate the projections of the slant ranges onto the plane, and perform 2D circular localization. The reference frame used in the Cartesian coordinate system is shown in Figure 3.2.

$$r(i)_{in-plane} = \sqrt{(R_i^{3D})^2 - \delta z_U^2} = \sqrt{(x_E - x_i)^2 + (y_N - y_i)^2} \tag{3.7}$$

where, R_i^{3D} are the slant range measurements to the beacons, where $i \in \{1, 2, 3\}$, $(x_E, y_N)^T$ is position of the glider in the horizontal plane, and $(x_i, y_i)^T$ are the spatial position coordinates of the nodes.

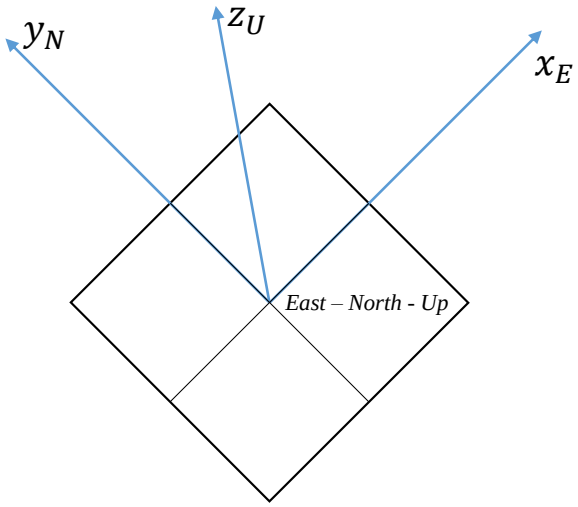


Figure 3.2: Inertial Frame of Reference

3.2.2 Linearization

Eq. (3.7) represents the nonlinear model for the LBL localization problem. The generic linear model and the linear estimator are given by [30],

$$\boldsymbol{r} = \boldsymbol{H}\boldsymbol{\theta} + \boldsymbol{v} \tag{3.8}$$

$$\hat{\boldsymbol{r}} = \boldsymbol{H}\hat{\boldsymbol{\theta}} \tag{3.9}$$

In this model, $\boldsymbol{r}$, which is $N \times 1$, is called the *measurement vector*, and $\boldsymbol{\theta}$ which is $n \times 1$, is called the *parameter vector*. $\boldsymbol{\theta}$ contains the unknown parameters that are to be estimated. $\boldsymbol{H}$ is an $N \times n$ matrix also called the *observation matrix,* $\boldsymbol{v}$, which is $N \times 1$, is called the *measurement noise* vector, and N is the number of measurements; $\hat{\boldsymbol{r}}$, which is $N \times 1$, is the "estimate" of the *measurement vector* and $\hat{\boldsymbol{\theta}}$*,* which is $n \times 1$, is the "estimate" of the *parameter vector,* which is to be calculated.

For the above model, another important quantity, the *residual error* $\boldsymbol{e}$ can be defined as,

$$\boldsymbol{e} \equiv \boldsymbol{r} - \hat{\boldsymbol{r}} \tag{3.10}$$

Then Eq. (3.9) can be rewritten as,

$$\boldsymbol{r} = \boldsymbol{H}\hat{\boldsymbol{\theta}} + \boldsymbol{e} \tag{3.11}$$

The next step is to express the nonlinear model in terms of the generic linear model. Nonlinear functions such as the one defined in Eq. (3.6) make the solution (i.e. $\hat{\boldsymbol{\theta}}$) difficult to find explicitly. For this reason, attention is directed to construction of a successive approximation procedure developed by Gauss, that is designed to converge to accurate least squares estimates, given approximate starting values. This is done through linearization, and vectorization of Eq. (3.7). Since this algorithm is applied *iteratively*, an initial guess

$(x_n, y_n)^T$ for the position of the glider is required. The glider's GPS position just prior to diving can be used. Given this initial guess, one may write,

$$\begin{aligned} x_E &= x_n + \delta x \\ y_N &= y_n + \delta y \end{aligned} \tag{3.12}$$

where $(\delta x, \delta y)^T$are the unknown corrections that need to be estimated recursively. At each time step, $(x_n, y_n)^T$is updated with the previous $(x_E, y_N)^T$, and becomes the nominal position of the glider. The nominal in-plane distance between this nominal position and the i^{th} node can be written using the 2-norm as,

$$r_n(i) = \parallel (x_n, y_n)^T - (x_i, y_i)^T \parallel \tag{3.13}$$

Applying the Taylor series expansion to the above equation at $(x_{E,}y_N)^T$, and neglecting the higher order terms, yields the first order approximation for the in-plane range.

$$r(i)\Big|_{(x_E, y_N)^T} \approx r_n(i) + \frac{(x_n - x_i)}{r_n(i)}\delta x + \frac{(y_n - y_i)}{r_n(i)}\delta y \tag{3.14}$$

The vector form of Eq. (3.12), gives Eq. (3.8) with $\boldsymbol{r}$,$\boldsymbol{\theta}$,and $\boldsymbol{H}$ defined as follows,

$$\boldsymbol{r} = \begin{pmatrix} r(1) - r_n(1) \\ r(2) - r_n(2) \\ r(3) - r_n(3) \end{pmatrix}, \qquad \boldsymbol{\theta} = \begin{pmatrix} \delta x \\ \delta y \end{pmatrix}, \qquad \boldsymbol{H} = \begin{pmatrix} \frac{x_n - x_1}{r_n(1)} & \frac{y_n - y_1}{r_n(1)} \\ \frac{x_n - x_2}{r_n(2)} & \frac{y_n - y_2}{r_n(2)} \\ \frac{x_n - x_3}{r_n(3)} & \frac{y_n - y_3}{r_n(3)} \end{pmatrix} \tag{3.15}$$

3.2.3 Number of Measurements and noise

Assuming $\boldsymbol{H}^{-1}$ exists, Eq. (3.8) can be written in terms of $\boldsymbol{\theta}$ as,

$$\boldsymbol{\theta} = \boldsymbol{H}^{-1}\boldsymbol{r} - \boldsymbol{H}^{-1}\boldsymbol{v} \tag{3.16}$$

Because $\boldsymbol{v}$ cannot be measured, it is usually neglected in the calculation of Eq. (3.14). For small amounts of noise neglecting $\boldsymbol{v}$ is permissible, however if the measurements are noisy, neglecting $\boldsymbol{v}$ can result in bad estimates.

Note that, $dim(\boldsymbol{r}) > dim(\boldsymbol{\theta})$, meaning there are more measurements than the parameters to be estimated. This makes Eq. (3.14) an overdetermined system. The extra measurement(s) can be used to offset the effects of the noise; i.e., they "filter" the data. However, if $size(\boldsymbol{r}) = size(\boldsymbol{\theta})$, the number of measurements and unknowns are equal. There can be a solution in this case, as long as the measurements are linearly independent, so that $\boldsymbol{H}^{-1}$exists. Hence, spherical localization is possible with 2 beacons as well, though noise will have a higher impact on the results.

3.2.4 Deriving the Estimator

Eqs. (3.8) and (3.9) are identical, of course, if $\hat{\boldsymbol{\theta}} = \boldsymbol{\theta}$, and if the assumption of zero model errors is valid. However, if this was true, then there would be no need for an estimator. An indirect method for obtaining $\hat{\boldsymbol{\theta}}$ is based on minimizing the objective function, which is nothing but the sum square of the residual errors defined by Eq. (3.10).

$$\boldsymbol{J}[\hat{\boldsymbol{\theta}}] = \frac{1}{2}\boldsymbol{e}^T\boldsymbol{e} \tag{3.17}$$

Substituting Eq. (3.11) for $\boldsymbol{e}$ into Eq. (3.17) and using the fact that a scalar equals its transpose yields,

$$\boldsymbol{J}[\hat{\boldsymbol{\theta}}] = \frac{1}{2}\left(\boldsymbol{r}^T\boldsymbol{r} - 2\boldsymbol{r}^T\boldsymbol{H}\hat{\boldsymbol{\theta}} + \hat{\boldsymbol{\theta}}^T\boldsymbol{H}^T\boldsymbol{H}\hat{\boldsymbol{\theta}}\right) \tag{3.18}$$

The objective is to find a $\hat{\boldsymbol{\theta}}$, that minimizes $\boldsymbol{J}$. Clearly, Eq. (3.18) is a quadratic function. To find a global minimum of a quadratic function, the following two requirements have to be met:

Necessary condition,

$$\nabla_{\hat{\boldsymbol{\theta}}}\boldsymbol{J} \equiv \begin{bmatrix} \frac{\partial \boldsymbol{J}}{\partial \delta x} \\ \frac{\partial \boldsymbol{J}}{\partial \delta y} \end{bmatrix} = (\boldsymbol{H}^T\boldsymbol{H}\hat{\boldsymbol{\theta}}) - \boldsymbol{H}^T\boldsymbol{r} = 0 \tag{3.19}$$

Sufficient condition,

$$\nabla^2_{\hat{\boldsymbol{\theta}}}\boldsymbol{J} \equiv \frac{\partial^2 \boldsymbol{J}}{\partial \hat{\boldsymbol{\theta}}\, \partial \hat{\boldsymbol{\theta}}^T} = \begin{bmatrix} \frac{\partial \boldsymbol{J}}{\partial \delta x\, \partial \delta x} & \frac{\partial \boldsymbol{J}}{\partial \delta x\, \partial \delta y} \\ \frac{\partial \boldsymbol{J}}{\partial \delta y\, \partial \delta x} & \frac{\partial \boldsymbol{J}}{\partial \delta y\, \partial \delta y} \end{bmatrix} = \boldsymbol{H}^T\boldsymbol{H} \text{ must be positive definite} \tag{3.20}$$

where $\nabla_{\hat{\boldsymbol{\theta}}}\boldsymbol{J}$ is the *Jacobian*, and $\nabla^2_{\hat{\boldsymbol{\theta}}}\boldsymbol{J}$ is the *Hessian*.

Considering the sufficient condition first, any matrix C such that,

$$\boldsymbol{x}^T C \boldsymbol{x} \geq 0$$

$\forall \boldsymbol{x} \neq 0$, is called *positive semi-definite*. By setting $\boldsymbol{h} = \boldsymbol{H}\boldsymbol{x}$ and squaring,

$$\text{LHS}: \ \boldsymbol{h}^T\boldsymbol{h} = h^2(\text{scalar})$$
$$\text{RHS}: \ (\boldsymbol{H}\boldsymbol{x})^T(\boldsymbol{H}\boldsymbol{x}) = \boldsymbol{x}^T(\boldsymbol{H}^T\boldsymbol{H})\boldsymbol{x} \tag{3.21}$$

From, Eq. (3.21) and (3.22), it can be seen that $\boldsymbol{H}^T\boldsymbol{H} \geq 0$, is always *positive semi-definite*. It becomes *positive definite* when $\boldsymbol{H}$ is of maximum rank ($n = 2$).

From the necessary condition of Eq. (3.19),

$$(\boldsymbol{H}^T\boldsymbol{H})\hat{\boldsymbol{\theta}} = \boldsymbol{H}^T\boldsymbol{r} \tag{3.22}$$

If the rank of $\boldsymbol{H}$ is $n = 2$ (i.e., there are at least 2 independent range measurements in this case), then $\boldsymbol{H}^T\boldsymbol{H}$ is *strictly* positive definite and can be inverted to obtain the explicit solution for the optimal estimate:

$$\hat{\boldsymbol{\theta}}_{\boldsymbol{LS}} = (\boldsymbol{H}^T\boldsymbol{H})^{-1}\boldsymbol{H}^T\boldsymbol{r} \tag{3.23}$$

3.2.5 Weighted Least Squares

The least squares criterion of Eq. (3.24), minimized to determine $\hat{\boldsymbol{\theta}}$, implicitly places equal emphasis on each measurement $r(i)$. In the LBL system developed for VTUG, the three measurements are not all recorded at the same time. There is a significant delay ($5s$) between successive range measurements. This is explained in more detail in Section 5.4. Hence, this "equal weight" approach seems logically unsound as the first and last measurements could be $10 - 15s$ apart. In order to incorporate appropriate weighting, the LS criterion is set up

in the form,

$$\boldsymbol{J} = \frac{1}{2}\boldsymbol{e}^T W \boldsymbol{e} \tag{3.24}$$

where W is an $m \times m$ symmetric weighting matrix (it is symmetric because the terms $e_i e_j, i \neq j$, are always equally weighted with the corresponding $e_j e_i$ terms). Next, the requirements are modified as,

Necessary Condition,

$$\nabla_{\hat{\boldsymbol{\theta}}} \boldsymbol{J} = \boldsymbol{H}^T W \boldsymbol{H}\hat{\boldsymbol{\theta}} - \boldsymbol{H}^T W \boldsymbol{r} = \boldsymbol{0} \tag{3.25}$$

Sufficient Condition,

$$\nabla^2_{\hat{\boldsymbol{\theta}}} \boldsymbol{J} = \boldsymbol{H}^T W \boldsymbol{H} \text{ must be positive definite} \tag{3.26}$$

Similar to the LS case, from Eq. (3.26), we obtain the solution for $\hat{\boldsymbol{\theta}}$ as,

$$\hat{\boldsymbol{\theta}}_{\boldsymbol{WLS}} = (\boldsymbol{H}^T W \boldsymbol{H})^{-1} \boldsymbol{H}^T W \boldsymbol{r} \tag{3.27}$$

Defining a vector $\boldsymbol{x}_k$ comprising the estimate of the coordinates of the glider at the k^{th} iteration,

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \hat{\boldsymbol{\theta}}_{\boldsymbol{WLS}} \tag{3.28}$$

Note that, $\boldsymbol{x} = (x_E, y_N)^T$, $\hat{\boldsymbol{\theta}}_{\boldsymbol{WLS}} = (\delta x, \delta y)^T$.

Also, from Eq. (3.26), W must be positive definite. From Chapter 2 of [32], the statistically optimal ("maximum likelihood") choice for the weights is shown to be the reciprocal of the measurement error variance for each beacon, i.e.,

$$W = diag\left(\frac{1}{\sigma^2_{r(1)}}, \frac{1}{\sigma^2_{r(2)}}, \frac{1}{\sigma^2_{r(3)}}\right) \tag{3.29}$$

An initial guess $(x_{in}, y_{in})^T$sufficiently close to the true solution is required to begin the algorithm. A stopping condition with an accuracy dependent tolerance for the minimization of $\boldsymbol{J}$ is given by,

$$\delta \boldsymbol{J} = \frac{\parallel \boldsymbol{J}_k - \boldsymbol{J}_{k-1} \parallel}{\boldsymbol{J}_k} < \frac{\epsilon}{\parallel W \parallel} \tag{3.30}$$

where ϵ is a prescribed small value. If Eq. (3.30) is not satisfied then the update procedure is iterated with the new estimate as the current estimate until the process converges, or unsatisfactory convergence progress is evident.

3.3 Spherical Localization Using Direct Algebraic Solution

In the nonlinear least squares solution described above, the linear least squares is applied iteratively to a linearized form of the function until convergence is achieved. However, it is often possible to linearize a nonlinear function at the outset, and still use linear methods for determining fit parameters without resorting to iterative procedures. The Bancroft algorithm does just that, by using the concept of the pseudoinverse. It will later be shown that the pseudoinverse has the form of the weighted least squares solution derived in Eq. (3.27). It

provides a direct algebraic solution to estimate the position without an initial guess, i.e., allows for a "cold start." It was originally developed for GPS by Stephen Bancroft [6]. The algorithm is modified from the original since there is no need to estimate the GPS clock offset. This is because, unlike the GPS system, the transmitter and receiver Micro-Modems are synced. More of this will be discussed in Section 5.3. The modified form is adopted from [33].

As noted above the direct algebraic solution is but another approach to solve the localization problem. The starting point is the nonlinear range-computing function defined by Eq. (3.6),

$$r_i = f(\hat{\theta}, i) = \sqrt{\left(\sum_{k=1}^{M}(x_k - b_{k,i})^2\right)} \tag{3.31}$$

where, $r(i)$: range estimates. $i = 1, ..., N$; x_k: the glider Cartesian location coordinates, $k = 1, ..., M$; and $b_{k,i}$: beacon locations in Cartesian coordinates. For the particular case of spherical localization, there are 3 coordinates, $M = 3$.

$$r_i = \sqrt{(x_E - x_i)^2 + (y_N - y_i)^2 + (z_U - z_i)^2} \tag{3.32}$$

This equation can be turned into a linear algebra problem using some algebraic manipulations. These will reduce the equations to a least squares problem described in Section 3.2. Rewriting Eq. (3.31), with $\boldsymbol{x} = (x_E, y_N, z_D)^T$ as the estimate of the glider's position, and $\boldsymbol{b}_i = (x_i, y_i, z_i)^T$ as the position of the i^{th} beacon in the inertial frame shown in Figure 3.2, with $i = 1, ..., n$, where n is the number of ranging units:

$$r_i^2 = (\boldsymbol{x} - \boldsymbol{b}_i)^T(\boldsymbol{x} - \boldsymbol{b}_i) = \boldsymbol{x}^T\boldsymbol{x} - 2\boldsymbol{x}^T\boldsymbol{b}_i + \boldsymbol{b}_i^T\boldsymbol{b}_i \tag{3.33}$$

$$\boldsymbol{b}_i^T \boldsymbol{x} = \frac{1}{2}(\boldsymbol{x}^T \boldsymbol{x} + \boldsymbol{b}_i^T \boldsymbol{b}_i - r_i^2) \tag{3.34}$$

Define $\alpha = \frac{1}{2}\boldsymbol{x}^T \boldsymbol{x}$ and $\beta_i = \frac{1}{2}(\boldsymbol{b}_i^T \boldsymbol{b}_i - r_i^2)$. Now, Eq. (3.32) can be written as,

$$\langle \boldsymbol{b}_i, \boldsymbol{x} \rangle = \alpha + \beta_i \tag{3.35}$$

where $\langle \boldsymbol{b}_i, \boldsymbol{x} \rangle$is the *euclidean inner product* or the dot product. Also, define $\boldsymbol{A} = [\boldsymbol{b}_1^T, \boldsymbol{b}_2^T, ..., \boldsymbol{b}_n^T]^T$, $\boldsymbol{1} = [1, ...1]^T$, $\boldsymbol{\beta} = [\beta_1, \beta_2, ..., \beta_n]^T$. The system of equations for all the ranging units can be written in a compact form as,

$$\boldsymbol{A}\boldsymbol{x} = \alpha \boldsymbol{1} + \boldsymbol{\beta} \tag{3.36}$$

Next, compute the weighted pseudoinverse,

$$\boldsymbol{B} = (\boldsymbol{A}^T W \boldsymbol{A})^{-1} \boldsymbol{A}^T W \tag{3.37}$$

where W is a symmetric positive definite weighting matrix (Eq. (3.29)). Compute the column vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ from,

$$\boldsymbol{u} = \boldsymbol{B}\boldsymbol{1} \tag{3.38}$$

$$\boldsymbol{v} = \boldsymbol{B}\boldsymbol{\beta} \tag{3.39}$$

Then from Eq. (3.34),

$$\boldsymbol{x} = \boldsymbol{u}\alpha + \boldsymbol{v} \tag{3.40}$$

Squaring the above equation,

$$\boldsymbol{x}^T\boldsymbol{x} = \alpha^2 \langle \boldsymbol{u}, \boldsymbol{u} \rangle + 2\alpha \langle \boldsymbol{u}, \boldsymbol{v} \rangle + \langle \boldsymbol{v}, \boldsymbol{v} \rangle \tag{3.41}$$

Using the definition of α, Eq. (3.39) can be written as,

$$\alpha^2 \langle \boldsymbol{u}, \boldsymbol{u} \rangle + \alpha 2(\langle \boldsymbol{u}, \boldsymbol{v} \rangle) + \langle \boldsymbol{v}, \boldsymbol{v} \rangle = 0 \tag{3.42}$$

The above quadratic equation can be solved for α. The solution is then found by substituting α in Eq. (3.40). Since there will be two solutions, one of them is rejected using depth sensor data. Generally, one of the solutions will be above the water surface and is eliminated.

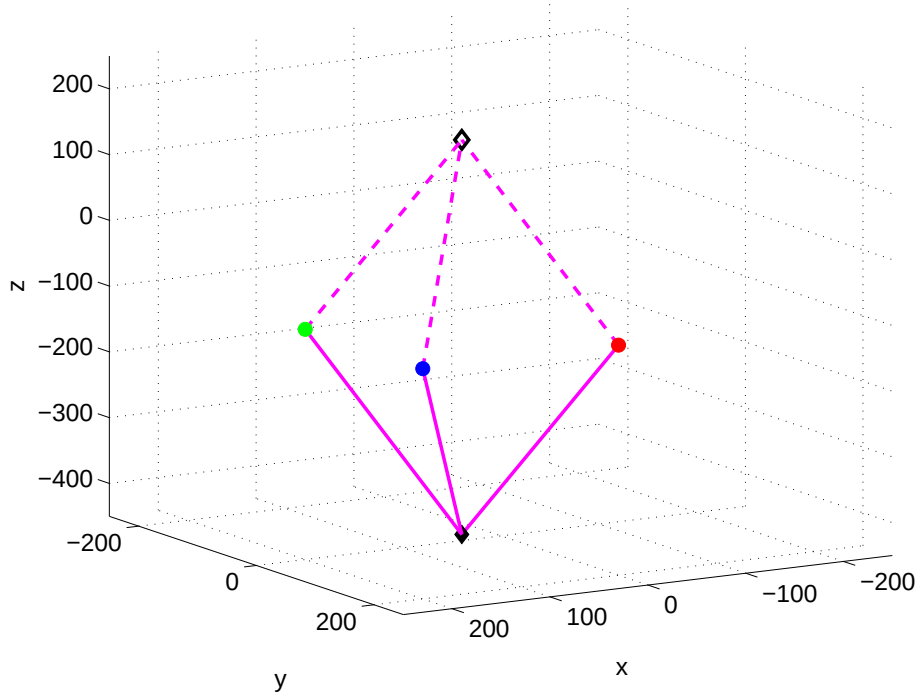


Figure 3.3: 3 beacons provide 2 solutions

3.4 Simulation Results

A simple numerical example is shown below to demonstrate the static estimation algorithms discussed in Sections 3.2 and 3.3. It is assumed that the glider started a turning descent at time $t = 0$. The table below shows all the parameters and their descriptions used in the simulations. All the positions are with respect to an Earth-fixed inertial frame. The ranging units were submerged underwater at a depth of 5m.

Parameter	Description	Value
$\boldsymbol{x}_0$	Initial glider position	$[200, 200, -100]^T$
$\boldsymbol{x}_{IC}$	Initial condition for least squares algorithm	$[210, 210, -100]^T$
$\boldsymbol{s}_1$ $\boldsymbol{s}_2$ $\boldsymbol{s}_3$	Positions of the three ranging units	$[-500, 500, -5]^T$ $[500, 500, -5]^T$ $[0, -1000, -5]^T$
$\boldsymbol{\omega}$	Turn rate	0.0003rad/s
$\boldsymbol{V}_a$	Forward speed	1.5m/s
$\boldsymbol{V}_x$	Velocity of current along x direction	0.4m/s
$\boldsymbol{V}_y$	Velocity of current along y direction	-0.4m/s
$\dot{\boldsymbol{z}}$	Vertical descent rate	-0.2m/s
n	Number of measurements	600

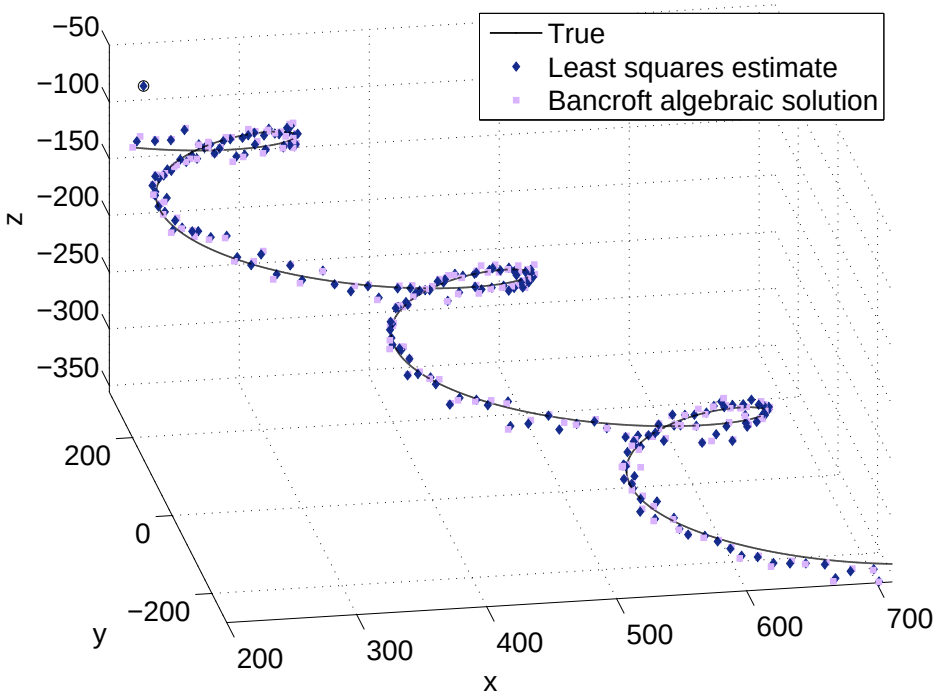
Table 3.1: Parameters of simulation for static estimation

The ranging measurements were assumed to be noisy. The noise was modeled as uniformly distributed random variable with $\sigma^2 = 3\text{m}^2$. Clearly, higher variance will lead to noisier and distorted estimations. The results are illustrated in Figure 3.4. The first figure shows a 3D plot, and the second shows a part of it projected on to the xy plane. As discussed in Section 3.2, since the vehicle is equipped with a depth sensor, 3D slant ranges can be converted to 2D projections, and all the analysis can be done in plane.

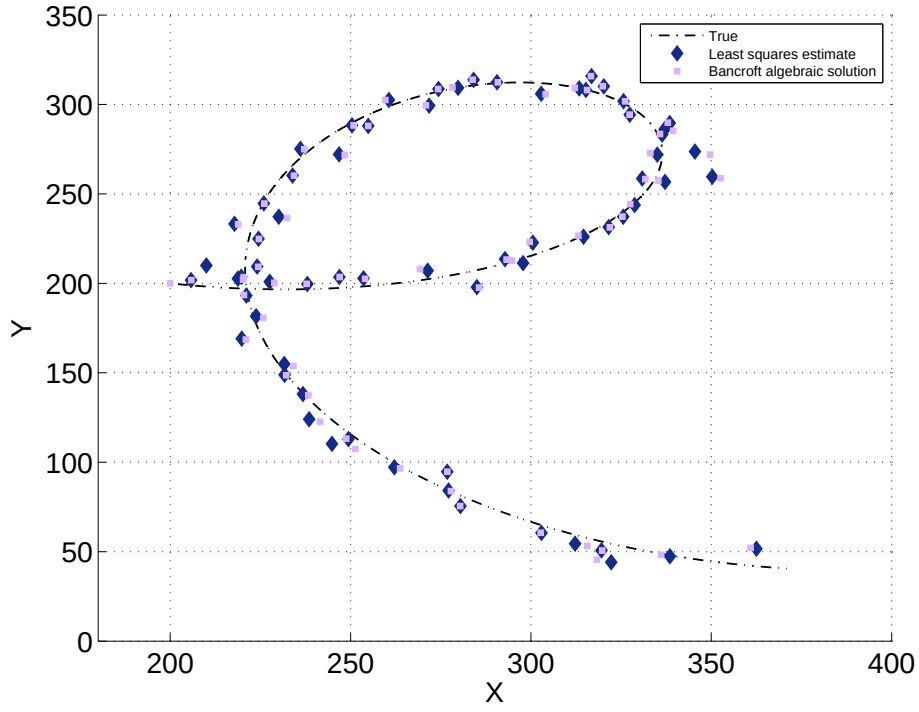
From the results it can observed that, both the methods yield comparable performance in simulations. Some advantages and disadvantages are discussed in Table 3.2.

Direct Solution	**Advantages**	**Disadvantages**
	• Non-iterative, Computationally efficient, and numerically stable. • Admits extended batch processing. • Allows for a "cold start" - no initial guess required. • Improves accuracy in bad geometric dilution of precision (GDOP) environments.	• Due to the nonlinear transformation (pseudoinverse), it's difficult to comment on the optimality • Properties of estimators such as finite sample and asymptotic properties are not known, including biasedness, efficiency
Iterative Least squares	**Advantages**	**Disadvantages**
	• Converges even with bad initial guesses. • With zero-mean Gaussian noise it is the minimum variance and maximum likelihood estimator. • With zero-mean Gaussian noise it is asymptotically efficient, and asymptotically unbiased.	• Requires an initial guess

Table 3.2: Comparing static estimation algorithms



(a) 3D plot



(b) 2D projection of (a)

Figure 3.4: Results of static estimation algorithms

Chapter 4

Dynamic Position Estimation

The previous section discussed localization algorithms which were simple, robust and easy to implement. They were static estimation algorithms, and did not account for any dynamic behavior of the vehicle and it's environment. Such methods are useful in ocean-mapping applications, where the accuracy required in position resolution is of the order of several meters ($5 - 10$m). However, to implement path planning and path following algorithms, such as those in [34] and [35], may require the accuracy in position estimates to be of the order of tens of centimeters or even less. There are many reasons as to why static localization algorithms provide such inaccurate measurements,

- Range measurements are inherently noisy due to many environmental factors
- All the range (from 3 beacons) measurements do not arrive simultaneously. In the design of the LBL system for the VTUG, there is a 5s delay between successive range measurements. Hence an estimate of the position is made available every $15s$. Also, during the 15s window mentioned above, the vehicle is assumed to be stationary, which is not a very good assumption, especially for a highly maneuverable glider such as the

VTUG.

- Static estimation algorithms require two/three (2D/3D) distinct successive range measurements, however, it has been observed in practice that some range measurements do not arrive within the time window. This basically means many measurements are missed/skipped and the range vector (Eq. (3.15)) is incomplete. In such instances, the algorithm fails to provide an estimate, and waits for all three measurements to arrive successively. The incomplete range vector is rendered obsolete, though it contains useful range information from some beacons, but not all.

Taking into account the above factors an intuitive solution to improve position accuracy, is to incorporate the dynamics of the vehicle, by using a vehicle model. Such a model takes into account the history of how the vehicle arrived at a particular configuration. This is the underlying idea behind the Kalman filter (KF). The KF has the ability to incorporate the dynamic motion of the glider into the estimation algorithm, thus providing better estimates.

The KF is the workhorse of estimation and has been used for a vast variety of applications. It provides a minimum variance state estimate for an observable linear system subject to zero-mean, Gaussian white noise. If applicable it is highly efficient and the best possible estimator (linear/non-linear). Implementation of the Kalman filter extends the fundamental recursive method, which is:

1. *Predict* the sate, covariance, and observations
2. *Observe* the measurable quantities related to the state
3. *Update* the predictions using the calculated gain

The Extended Kalman Filter (EKF) builds on the theoretical elegance of the Kalman filter. It is a generalization of the KF in case of a nonlinear state and/or measurement model. It

simply linearizes the model about the state estimate at each time-step. The approach used in the EKF assumes that the true state is "close" to the estimated state. This restriction can prove to be especially damaging for highly nonlinear applications with large initial condition errors. It is also not precisely "optimum" like the KF, however, it has been successfully applied to many nonlinear systems in the past. The EKF has the ability to process the range measurements independently as they become available. This way it overcomes the 2nd, and 3rd challenges of static estimation algorithms listed above.

In literature there are numerous examples of it being used in AUV localization and navigation. In [36], Baccou *et al.* use the EKF as well the iterated nonlinear least squares for position estimation using only one beacon. They also compare it to conventional LBL systems. A similar work is also presented in [37] by Gadre *et al.* from Virginia Tech. Bingham [31] gives a thorough treatment of EKF and how it can also be used to study errors in a combined DVL-LBL system. Other examples include [5] and [38].

The implementation presented in Section 4.2, is hybrid or continuous-discrete EKF adopted from chapter 5 of [32], and [29]. The LBL system developed for the seaglider by Techy *et al.* [29], served as the principal guide in the design and implementation of one for the VTUG.

4.1 Vehicle Model

A simple unicycle model is used to describe the dynamics of the glider. This model, in general, describes a robot moving in a 2D world, having some forward speed, but zero instantaneous lateral motion [39]. The below equations describe the basic unicycle model.

$$\dot{x} = V \cos\theta \tag{4.1}$$

$$\dot{y} = V \sin\theta \tag{4.2}$$

$$\dot{\psi} = u \tag{4.3}$$

The above model is extended with underwater current velocities in order to estimate them.

$$\dot{x}_E = V_a \cos\psi + V_x \tag{4.4}$$

$$\dot{y}_N = V_a \sin\psi + V_y \tag{4.5}$$

$$\dot{\psi} = u \tag{4.6}$$

$$\dot{V}_a = 0 \tag{4.7}$$

$$\dot{V}_x = 0 \tag{4.8}$$

$$\dot{V}_y = 0 \tag{4.9}$$

V_a is the Flow-relative speed of the glider, V_x denotes the east component of the current velocity, V_y denotes the north component of the current velocity, ψ is the heading angle measured from the east, and u is the turn rate of the vehicle, which is the only control input in this model. Figure 4.1 illustrates this model. Clearly, the above model is nonlinear in the glider position $(x_E, y_N)^T$, and hence the need to use an EKF.

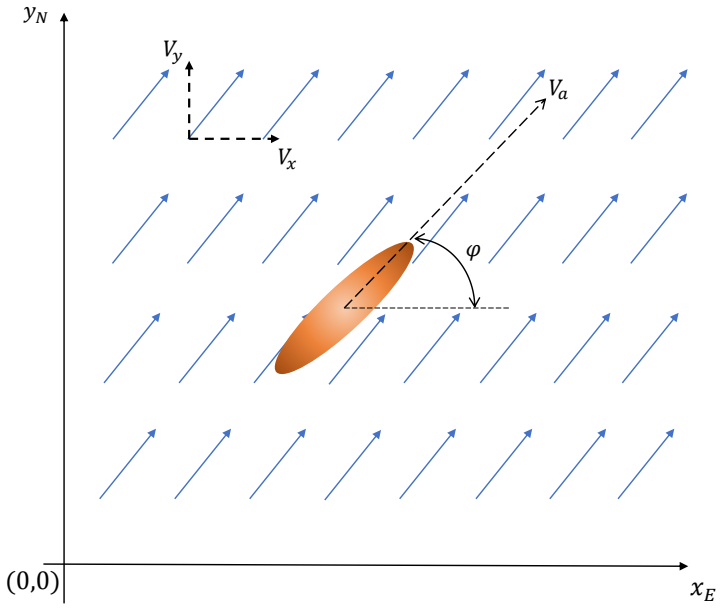


Figure 4.1: Unicycle model for the glider

The turn rate is determined by the vehicle roll angle. During a run, it can be obtained from logging the serial data from inertial measurement unit (IMU). For the above state-space model, the state vector would be,

$$\boldsymbol{x} = (x_E, y_N, \psi, V_a, V_x, V_y) \tag{4.10}$$

Notice that the velocities V_x, V_y and V_a are treated as states. In actuality, they are constant or slowly varying parameters. The EKF, along with providing estimates of the position, also provides an estimate of these velocities. Another important observation here is that the EKF provides 2D estimates of position. However, the glider is equipped with a depth sensor whose data is logged independently. It is a trivial task to incorporate 3D positioning into this model.

4.2 Hybrid Extended Kalman Filter

Modern day robots are dynamical systems implemented with continuous time models and discrete time measurements, obtained from a digital signal processor. The EKF is hybrid, because the states are propagated in continuous time, while the measurements are obtained at discrete time steps, whenever a new range measurement is available. Figure 4.2 illustrates the mechanism of a hybrid Kalman filter [32],

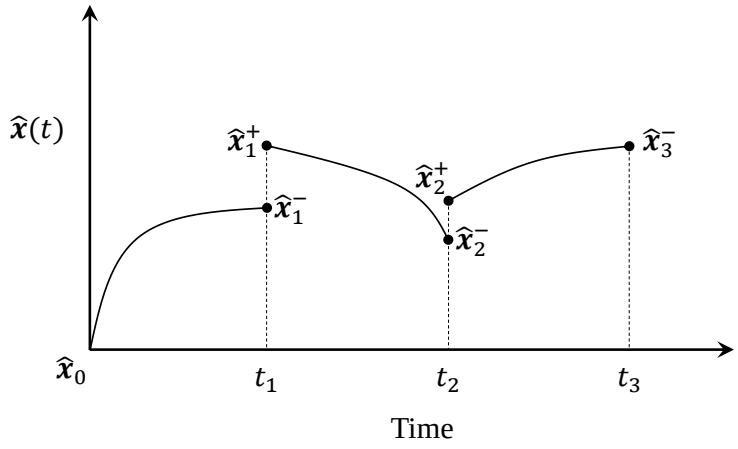


Figure 4.2: Mechanism for hybrid Kalman filter

It can be observed from Figure 4.2 that, the state estimate is propagated forward in time until a measurement occurs, given at time t_1. Then a discrete-time state update occurs, which updates the final value of the propagated state $\hat{\boldsymbol{x}}_1^-$ to the new state $\hat{\boldsymbol{x}}_1^+$. Finally this state is then used as the initial condition to propagate the state estimate model to time t_2.

As mentioned, the EKF is an extension of the Kalman filter to work with nonlinear process models. Eqs. (4.4 - 4.9), can be combined into a nonlinear state model as follows,

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}\Big(\boldsymbol{x}(t), u(t)\Big) + \boldsymbol{w}(t) \tag{4.11}$$

where $\boldsymbol{w}(t) \backsim \mathcal{N}(\mathbf{0}, \boldsymbol{W}(t))$ is the process noise, assumed to be zero-mean, Gaussian, and white; $\boldsymbol{f}\Big(\boldsymbol{x}(t), u(t)\Big)$ is assumed to be continuously differentiable. At discrete time intervals $t = kT$ measurements are available, defined by the output state equations as,

$$\boldsymbol{y}(kT) = \boldsymbol{h}(\boldsymbol{x}(kT)) + \boldsymbol{v}(kT) \tag{4.12}$$

where T is the sampling period, $k \in \mathbb{Z}$ and $\boldsymbol{v}(kT) \backsim \mathcal{N}(\mathbf{0}, \boldsymbol{V}_k)$ is the measurement noise, which is also assumed to be zero-mean, Gaussian and white. $\boldsymbol{W}(t)$, $\boldsymbol{V}_k$ and $\boldsymbol{x}(t_0)$ are uncorrelated for all t and k. The measurement vector for the LBL localization problem is,

$$\boldsymbol{h}(\boldsymbol{x}(kT)) = \begin{pmatrix} R(kT) \\ \psi(kT) \end{pmatrix} \tag{4.13}$$

where, $R(kT)$ is the range measurement from a corresponding node. It is obtained by converting the slant range (3D) to in-plane range (2D) using the depth sensor readings on board the glider. $\psi(kT)$ is the vehicle heading angle obtained from the IMU. As shown in Figure 4.2, between the measurements the state is propagated in continuous time using the nonlinear model in Eq. (4.11), but with zero process noise.

$$\dot{\hat{\boldsymbol{x}}}(t) = \boldsymbol{f}\Big(\hat{\boldsymbol{x}}(t), u(t)\Big) \tag{4.14}$$

To satisfy Eq. (4.14) with some nominal turn rate input $u(t)$, the estimated $\hat{\boldsymbol{x}}(t)$, must be a nominal trajectory that exists. Lesson 24 of [30] gives the proof for this necessary condition.

The solution to Eq. (4.14), is obtained through numerical integration, which is evaluated between measurement updates.

The state covariance matrix is also propagated in continuous time using the linearized state matrix, obtained by taking the Taylor series expansion of Eq. (4.14) with respect to the estimate at the time instant a measurement arrives. It is given by,

$$\dot{\boldsymbol{P}}(t) = \boldsymbol{A}(\hat{\boldsymbol{x}}(t))\boldsymbol{P}(t) + \boldsymbol{P}(t)\boldsymbol{A}^T(\hat{\boldsymbol{x}}(t)) + \boldsymbol{W}(t) \tag{4.15}$$

$$\boldsymbol{A}(\hat{\boldsymbol{x}}(t)) \equiv \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\right|_{\hat{\boldsymbol{x}}(t)} = \left.\begin{pmatrix} 0 & 0 & -V_a \sin\psi & \cos\psi & 1 & 0 \\ 0 & 0 & V_a \cos\psi & \sin\psi & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}\right|_{\hat{\boldsymbol{x}}(t)} \tag{4.16}$$

It is also important to note that, Eq. (4.14) and Eq. (4.15) need to be initialized to start the estimation algorithm. These initial conditions need to be "close" to the true values for the estimator to converge. They are initiated at time t_0 as follows,

$$\hat{\boldsymbol{x}}(t_0) = \hat{\boldsymbol{x}}_0 \tag{4.17}$$

$$\boldsymbol{P}_0 = E\{\tilde{\boldsymbol{x}}(t_0)\tilde{\boldsymbol{x}}^T(t_0)\} \tag{4.18}$$

where, operator E is the expectation operator, and $\tilde{\boldsymbol{x}}(t_0) = \hat{\boldsymbol{x}}(t) - \boldsymbol{x}(t)$ is the state error vector.

When a new measurement arrives at time step k, the Kalman gain matrix $\boldsymbol{K}_k$ is computed and corrections are applied to the mean and covariance estimate of the state vector. The gain matrix is given by,

$$\boldsymbol{K}_k = \boldsymbol{P}_k^- \boldsymbol{H}_k^T(\hat{\boldsymbol{x}}_k^-)\left(\boldsymbol{H}_k(\hat{\boldsymbol{x}}_k^-)\boldsymbol{P}_k^- \boldsymbol{H}_k^T(\hat{\boldsymbol{x}}_k^-) + \boldsymbol{V}_k\right)^{-1} \tag{4.19}$$

In the above equation, similar to Eq. (4.16), $\boldsymbol{H}_k$ is obtained by taking the first order Taylor series expansion of Eq. (4.13), evaluated at the state estimate just before a new measurement arrives, i.e., $\hat{\boldsymbol{x}}_k^-$ (*a priori* estimate of the state) , as shown below,

$$\boldsymbol{H}_k(\hat{\boldsymbol{x}}_k^-) \equiv \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}}\bigg|_{\hat{\boldsymbol{x}}_k^-} = \begin{pmatrix} \frac{\hat{x}_{E_k}^- - x_i}{\hat{R}_k} & \frac{\hat{y}_{N_k}^- - y_i}{\hat{R}_k} & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}\Bigg|_{\hat{\boldsymbol{x}}_k^-}$$

where $(x_i, y_i)^T$ are the corresponding node locations that provided the range information, $i \in 1, 2, 3$, and

$$\hat{R}_k = \sqrt{(\hat{x}_{E_k}^- - x_i)^2 + (\hat{y}_{N_k}^- - y_i)^2}$$

The update is then applied using the equations,

$$\hat{\boldsymbol{x}}_k^+ = \hat{\boldsymbol{x}}_k + \boldsymbol{K}_k(\boldsymbol{y}_k - \boldsymbol{h}(\hat{\boldsymbol{x}}_k^-)) \tag{4.20}$$

$$\boldsymbol{P}_k^+ = (\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k(\hat{\boldsymbol{x}}_k^-))\boldsymbol{P}_k^- \tag{4.21}$$

where $\boldsymbol{x}_k^-$ is *the a priori* estimate of the state i.e., estimate before the measurement arrived

and $\boldsymbol{x}_k^+$ is the *a posteriori* estimate of the state i.e., estimate after the measurement arrived.

To summarize the implementation of the EKF as a recursive estimation algorithm, refer to the table below [32],

Model	$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}\Big(\boldsymbol{x}(t), u(t)\Big) + \boldsymbol{w}(t)$ $\boldsymbol{y}(kT) = \boldsymbol{h}(\boldsymbol{x}(kT)) + \boldsymbol{v}(kT)$
Initialize	$\hat{\boldsymbol{x}}(t_0) = \hat{\boldsymbol{x}}_0$ $\boldsymbol{P}_0 = E\{\tilde{\boldsymbol{x}}(t_0)\tilde{\boldsymbol{x}}^T(t_0)\}$
Gain	$\boldsymbol{K}_k = \boldsymbol{P}_k^- \boldsymbol{H}_k^T(\hat{\boldsymbol{x}}_k^-)\Big(\boldsymbol{H}_k(\hat{\boldsymbol{x}}_k^-)\boldsymbol{P}_k^-\boldsymbol{H}_k^T(\hat{\boldsymbol{x}}_k^-) + \boldsymbol{V}_k\Big)^{-1}$
Update	$\hat{\boldsymbol{x}}_k^+ = \hat{\boldsymbol{x}}_k^- + \boldsymbol{K}_k(\boldsymbol{y}_k - \boldsymbol{h}(\hat{\boldsymbol{x}}_k^-))$ $\boldsymbol{P}_k^+ = (\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k(\hat{\boldsymbol{x}}_k^-))\boldsymbol{P}_k^-$
Propagate	$\dot{\hat{\boldsymbol{x}}}(t) = \boldsymbol{f}\Big(\hat{\boldsymbol{x}}(t), u(t)\Big)$ $\dot{\boldsymbol{P}}(t) = \boldsymbol{A}(\hat{\boldsymbol{x}}(t))\boldsymbol{P}(t) + \boldsymbol{P}(t)\boldsymbol{A}^T(\hat{\boldsymbol{x}}(t)) + \boldsymbol{W}(t)$

Table 4.1: Hybrid EKF

It should be noted that the sample times (kT), of the measurements need not occur in regular intervals. In fact, different measurement sets can be spread out over various time intervals.

Whenever a measurement occurs, an update is invoked. The measurement set at the time may involve only one measurement or multiple measurements. This is the main advantage of the EKF over static algorithms. The real beauty of the continuous-discrete Kalman filter is that it can handle different scattered measurement sets quite easily.

4.3 RTS Smoothing

In Chapter 3, estimation concepts were formulated and applied to systems whose measured variables were related to the estimated parameters by *algebraic* equations. In the previous sections of this chapter, the static algorithms were extended to include dynamical model of the glider, where the model included, both *algebraic* and *differential* equations. Specifically, a measurement taken at the current time, and an estimate of the state at the previous time with knowledge of its error properties, were combined using the EKF to produce a state estimate of the dynamic system at the current time. In both these cases (static and dynamic) the measurements were processed sequentially: one at a time. In the case of the EKF, all the measurements, up until the latest sample were used recursively to calculate the best estimate.

In this section, the results of the EKF are extended to *batch state estimation*, where, the set of measurements during the entire time interval T, are used to estimate the state at any given time t. A clear disadvantage of batch state estimators is that they cannot be used in real time. However, if all the measurement data is available for post-processing, then at any given time t, the future and past data can be used in the estimation. Such estimators give intuitively better estimates, and have the advantage of providing state estimates with lower error covariance than sequential methods. This can be extremely helpful in situations where there is a high accuracy requirement, but real-time application is not required. Batch estimators are also

called *smoothers*, since they typically "smooth" out the effects of measurement noise.

The *Rauch-Tung-Striebel* (RTS) smoother is a fixed-interval smoothing batch estimator. Fixed-interval smoothers utilize the entire batch of measurements over a fixed interval $(0-T)$ to estimate all the states in the interval, at a variable point t. It has been adopted from Chapter 6 of [32].

Any fixed interval smoother contains the following three components,

1. Forward-time filter
2. Backward-time filter
3. Smoother

Among all the fixed interval smoothers, the RTS smoother is most widely used because it combines the backward-time filter, and smoother into a single backward-time smoother. Hence, the final smoothed estimates are calculated in the backward pass. This is accomplished at any time t, in the following way,

1. First, an EKF is implemented as shown in Table 4.1, up to time t. A forward time state estimate is obtained: $\hat{\boldsymbol{x}}_f(t)$. The forward filter state estimate and covariance are stored.
2. The backward smoother is implemented as a linearized Kalman filter (5.6 of [32]). The linearized Kalman filter is a simpler form of the EKF, where instead of linearizing about the current estimate, a nominal state vector $\bar{\boldsymbol{x}}(t)$ is used. These nominal state vectors are determined *a priori*, hence the gains can be calculated beforehand. This saves a lot of time in real-time applications. The disadvantage being that nominal state estimates are generally worse than the current estimates, i.e., $\left(\bar{\boldsymbol{x}}(t) - \boldsymbol{x}(t)\right) > \left(\hat{\boldsymbol{x}}(t) - \boldsymbol{x}(t)\right)$.

This makes the linearized Kalman filter less accurate than the conventional EKF. In the case of the RTS smoother this weakness is overcome by using the stored state estimate $\hat{\boldsymbol{x}}_f(t)$ (step 1), and covariance, obtained in the forward pass, instead of the nominal state vector to linearize the backward time smoother.

3. After linearization, the dynamic model is integrated in reverse time to obtain the smoothed estimate. The table below summaries the RTS smoother [32].

Note that $\tau = T - t$, and $d\boldsymbol{x}/dt = -d\boldsymbol{x}/d\tau$.

Model	$\frac{d}{d\tau}\boldsymbol{x}(t) = \boldsymbol{f}\Big(\boldsymbol{x}(t), u(t)\Big) + \boldsymbol{w}(t), \;\; \boldsymbol{w}(t) \backsim \mathcal{N}(\boldsymbol{0}, \boldsymbol{W}(t))$ $\boldsymbol{y}(kT) = \boldsymbol{h}(\boldsymbol{x}(kT)) + \boldsymbol{v}(kT), \;\; \boldsymbol{v}(kT) \backsim \mathcal{N}(\boldsymbol{0}, \boldsymbol{V}_k)$
Forward Initialize	$\hat{\boldsymbol{x}_f}(t_0) = \hat{\boldsymbol{x}}_{f0}$ $\boldsymbol{P}_{f0} = E\{\tilde{\boldsymbol{x}}_f(t_0)\tilde{\boldsymbol{x}}_f^T(t_0)\}$
Forward Gain	$\boldsymbol{K}_{fk} = \boldsymbol{P}_{fk}^- \boldsymbol{H}_k^T(\hat{\boldsymbol{x}}_{fk}^-)\Big(\boldsymbol{H}_k(\hat{\boldsymbol{x}}_{fk}^-)\boldsymbol{P}_{fk}^- \boldsymbol{H}_k^T(\hat{\boldsymbol{x}}_{fk}^-) + \boldsymbol{V}_k\Big)^{-1}$ $\boldsymbol{H}_k(\hat{\boldsymbol{x}}_{fk}^-) \equiv \left.\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}}\right\vert_{\hat{\boldsymbol{x}}_{fk}^-}$
Forward Update	$\hat{\boldsymbol{x}}_{fk}^+ = \hat{\boldsymbol{x}}_{fk}^- + \boldsymbol{K}_{fk}(\boldsymbol{y}_k - \boldsymbol{h}(\hat{\boldsymbol{x}}_{fk}^-))$ $\boldsymbol{P}_{fk}^+ = (\boldsymbol{I} - \boldsymbol{K}_{fk}\boldsymbol{H}_k(\hat{\boldsymbol{x}}_{fk}^-))\boldsymbol{P}_{fk}^-$
Forward Propagation	$\frac{d}{dt}\hat{\boldsymbol{x}}_f(t) = \boldsymbol{f}\Big(\hat{\boldsymbol{x}}_f(t), u(t)\Big)$ $\frac{d}{dt}\boldsymbol{P}_f(t) = \boldsymbol{A}(\hat{\boldsymbol{x}}_f(t))\boldsymbol{P}_f(t) + \boldsymbol{P}_f(t)\boldsymbol{A}^T(\hat{\boldsymbol{x}}_f(t)) + \boldsymbol{W}(t)$ $\boldsymbol{A}(\hat{\boldsymbol{x}}_f(t)) \equiv \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\right\vert_{\hat{\boldsymbol{x}}_f(t)}$
Smoother Gain	$\boldsymbol{K}(t) \equiv \boldsymbol{W}(t)\boldsymbol{P}_f^{-1}(t)$
Smoother Covariance	$\frac{d}{d\tau}\boldsymbol{P}(t) = -[\boldsymbol{A}(\hat{\boldsymbol{x}}_f(t) + \boldsymbol{K}(t)]\boldsymbol{P}(t) - \boldsymbol{P}(t)[\boldsymbol{A}(\hat{\boldsymbol{x}}_f(t)) + \boldsymbol{K}(t)]^T + \boldsymbol{W}(t)$
Smoother Estimate	$\frac{d}{d\tau}\hat{\boldsymbol{x}}(t) = -[\boldsymbol{A}(\hat{\boldsymbol{x}}_f(t)) + \boldsymbol{K}(t)][\hat{\boldsymbol{x}}(t) - \hat{\boldsymbol{x}}_f(t)] - \boldsymbol{f}(\hat{\boldsymbol{x}}_f(t), u(t))$

Table 4.2: Hybrid RTS smoother

4.4 Simulation Results

The EKF and RTS smoothing algorithms were tested in simulations. The flight of the underwater vehicle described by Eqs. (4.4 - 4.9), was simulated in Matlab. Table 4.3 shows all the parameter descriptions, and their values used in the simulations. The process and measurement covariance matrices were selected as,

$$\boldsymbol{W} = \begin{pmatrix} \sigma_x^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_y^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_\psi^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{V_a}^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{V_x}^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_{V_y}^2 \end{pmatrix} = 10^{-5} \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 115 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.1 \end{pmatrix} \tag{4.22}$$

$$\boldsymbol{V} = \begin{pmatrix} \sigma_R^2 & 0 \\ 0 & \sigma_\psi^2 \end{pmatrix} = \begin{pmatrix} 10 & 0 \\ 0 & 115 \cdot 10^{-5} \end{pmatrix} \tag{4.23}$$

The measurement noise that was selected for the heading angle measurement was the same as the one estimated during calibration. The variance in the heading angle was about 2^o. The variance of the heading (yaw) was higher than the roll and pitch ($< 1^o$), because the position at which the IMU was mounted inside the glider had high magnetic interference. The IMU uses magnetometers to calculate heading, hence the higher variance.

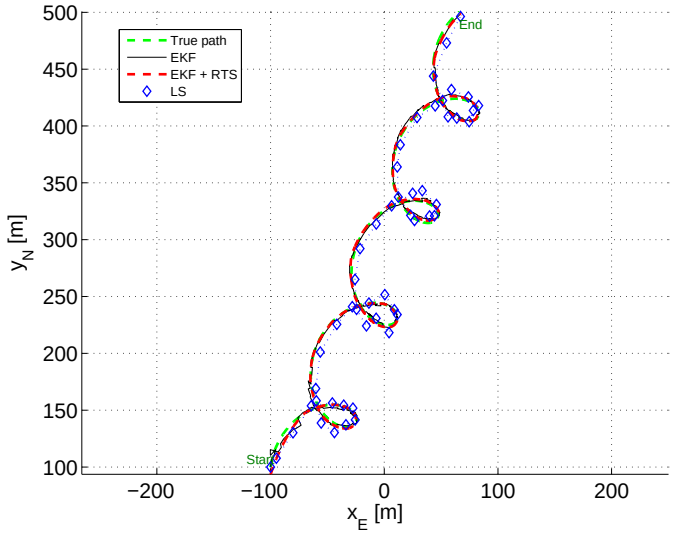
Parameter	Description	Value
$\boldsymbol{s}_1$ $\boldsymbol{s}_2$ $\boldsymbol{s}_3$	Positions of the three ranging units	$[-500, 500]^T$ $[500, 500]^T$ $[0, -1000]^T$
(x_0, y_0)	Starting position of the glider	$[100, -100]^T$
V_a V_x V_y	Constant glider speed Current speed in x direction Current speed in y direction	1m/s 0.5m/s 0.2m/s
$\hat{V}_{a_0}$ $\hat{V}_{x_0}$ $\hat{V}_{y_0}$	Initial glider speed for filter Initial current speed in x direction Initial current speed in y direction	0.8m/s 0m/s 0m/s
u u_{model}	Turn rate of vehicle Turn rate for estimator equations	2deg/s 0deg/s
$\triangle t$	Delay between range measurements	5s
n	Number of measurements	150
T	Total simulation time	12.5min
δt	Sampling period	100ms

Table 4.3: Simulation parameters for dynamic estimation

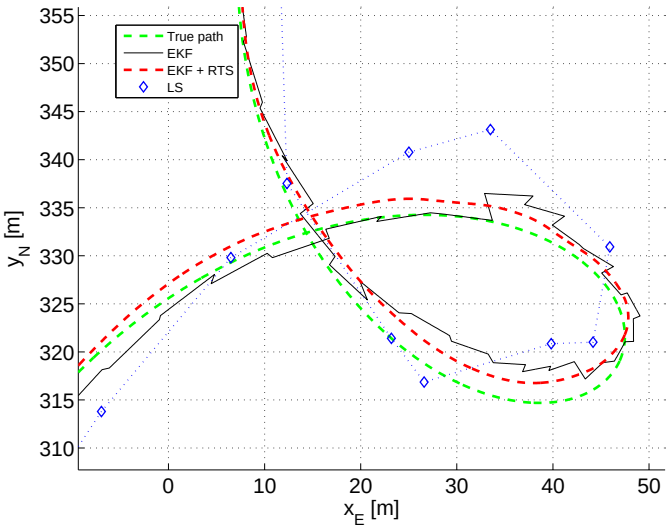
The measurement noise selected for the range measurements, was a calculated guess. The calibration of acoustic modems used to provide range measurements was not done due to time constraints. At distances within 1km, the WHOI acoustic modems are certified to provide

range measurements accurately with a residual error of only 1.11m [40]. So a variance of 10m is an order of magnitude higher than the actual. The determination of process noise was less trivial than the measurement noise. Although the process noise was assumed to zero-mean, Gaussian and white, in practice this is rarely the case. However, the Kalman filter exhibits some amount of robustness in the erroneous selection of the process covariance matrix.

Figures 4.3, 4.4 and 4.5 show the results of the simulations,

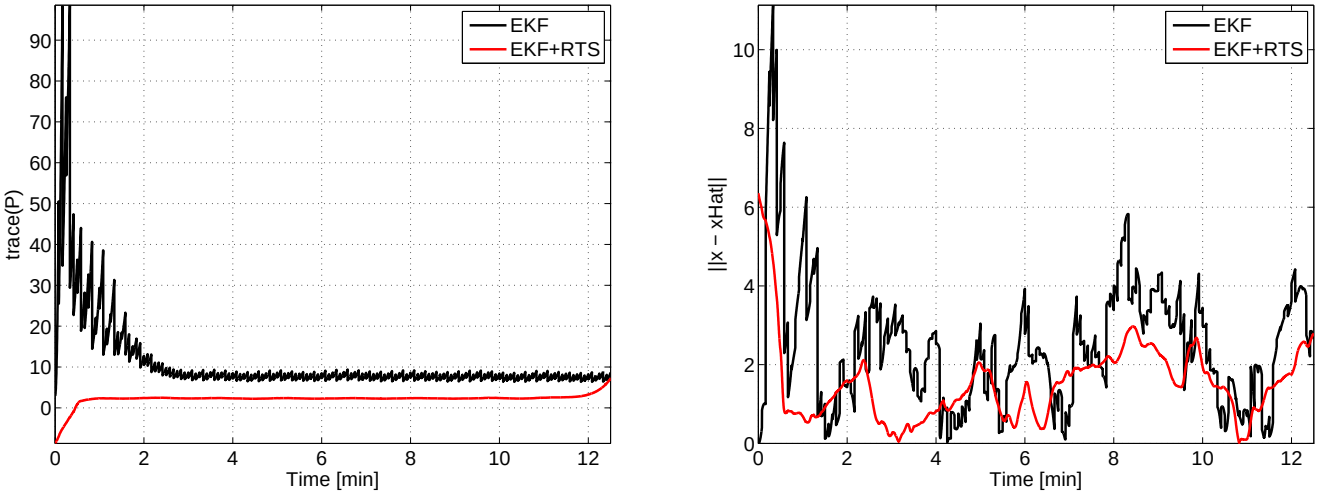


(a) For the whole run



(b) Improvement of EKF, and EKF+RTS over LS

Figure 4.3: Planar estimation using EKF, and EKF+RTS



(a) (b)

Figure 4.4: Trace of the covariance matrix (a), and magnitude of the planar position error (right)

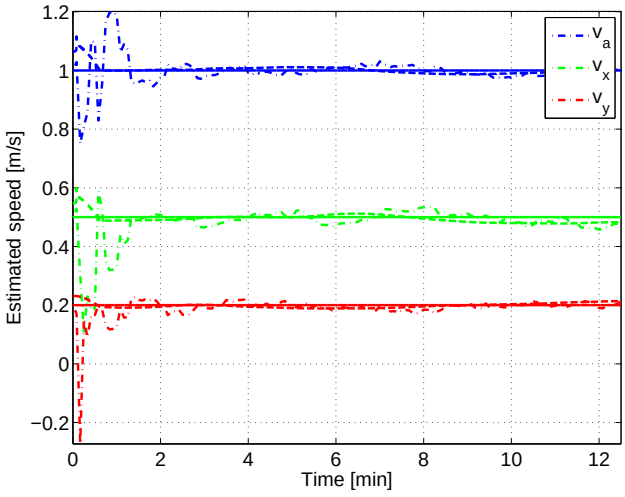


Figure 4.5: Current speed components and glider speed in simulations. Solid line is *true*, dashed-dot is *EKF*, dashed is *EKF+RTS*

Figure 4.3 illustrates the performance of the dynamic EKF in comparison with the static least squares algorithm. The EKF with RTS smoothing further improves the estimate.

Some observations from the simulation results:

- In simulations, from Figure 4.5, it can be observed that the current velocity components, and the glider's flow relative speed converged to the true values. However, if the initial values of these velocities fed into the estimator were not close to the original values, then convergence took much longer, and sometimes did not converge to feasible values.

- If the true vehicle turn rate was set to zero, the estimates of the speed components $(\hat{V}_x, \hat{V}_y)$ were very poor. This is due to the fact that the filter dynamics did not distinguish between glider speed, and glider speed relative to the current speed in this case. For the estimator to distinguish between them, the glider needs to turn.

- The EKF converged to true values of the velocities when the variation in the current speeds were negligible compared to the rate at which the glider's motion is excited in different inertial directions.

- Due to the large variance in the heading angle, (refer [29] for a comparison), the magnitude of the error in the estimates was high, especially during turns, illustrated in Figure 4.3(b). From Figure 4.4(b) it can be seen that even the RTS estimates were not smooth. With a better placement of the IMU, or using IMUs that do not use magnetometers, this can be improved.

- The values of the covariance matrix were selected based on a trade-off between convergence time and stability of the estimate. Smaller values yield smooth and stable estimates, though at a lower rate, and vice versa.

Chapter 5

Experimental Setup

This section describes the experimental setup of the LBL positioning system. It provides a brief overview of the underwater glider, followed by the design of the LBL nodes, the acoustic communication system, and software design overview. The experimental site for all the tests conducted was Claytor Lake, VA, about 30 miles from the Virginia Tech campus.

5.1 Underwater Vehicle Platform: VTUG

Over the past few decades, marine scientists and biologists have envisioned the use of AUV's for long term, large-scale oceanographic monitoring. However, the propulsion systems and power storage limitations of conventional AUVs do not allow for long-term deployments. Conventional, battery-powered, propeller driven AUVs can only operate on the order of a few hours before their power is depleted. Even if there is a significant improvement in their power systems, allowing for longer deployment, their external moving parts are likely to fail due to biological growth, corrosion or collisions.

Gliders typically perform sawtooth type motion patterns underwater by changing their buoyancy at the surface and at depth, the wings then convert the vertical to forward motion. They are basically winged underwater vehicles which locomote by modifying their internal shape. The locomotion is typically done using two modules,

1. **Buoyancy control system** (BCS) which modulates its net weight to help it sink of float. There are many different designs, like, piston-cylinder, oil-filled bladder, or compressed air buoyancy system.

2. **Attitude control system** (ACS) which consists of an internal moving mass with one or more actuators for pitch and/or roll control.

Gliders have high endurance and are highly efficient because they spend most of their time in a steady, stable, equilibrium glide, and they expend energy infrequently to change their buoyancy. Virginia Tech has developed a shallow water (100 meter depth), coastal underwater glider designed to serve as a platform for testing new perception, planning, and control algorithms [34, 35, 41]. Shallow water gliders can be used for measuring concentrations of phycotoxins, perform surveillance, and acoustic mapping. In these applications, robustness is as important as efficiency. The mechanical and electrical design as well as the fabrication of the glider was done by Artur Wolek, under the direction of Dr. Craig Woolsey. Wolek et al. [1] give an overview of the design and development of the VTUG. This section reproduces some aspects of the design illustrated in [1], and discusses recent design improvements.

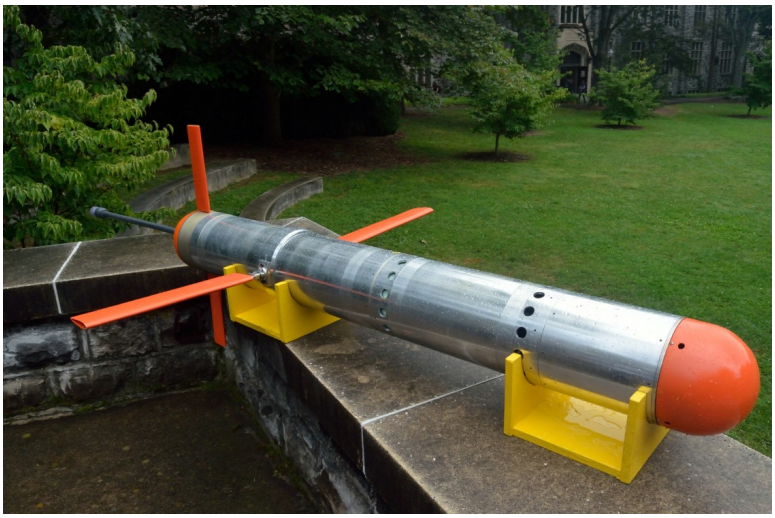
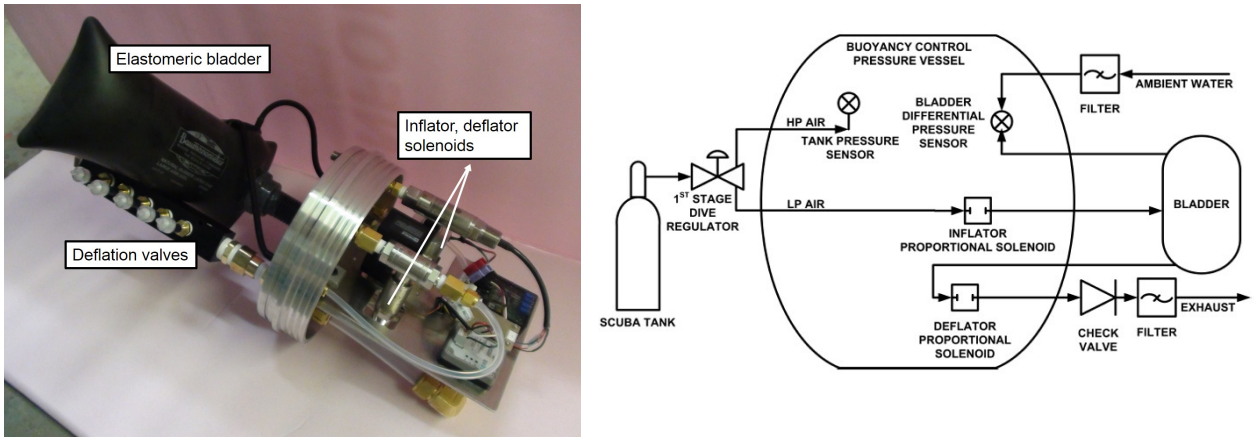


Figure 5.1: Virginia Tech Underwater Glider

The VTUG (Figure 5.1) uses a compressed air buoyancy system. In this configuration the BCS consists of a elastomeric bladder which is inflated and deflated using solenoids (see Figure 5.2(a)), with a scuba tank providing the power (compressed air). This design allows for rapid changes in buoyancy, and easy refueling after missions, however the endurance of the glider is limited by the volume of the scuba tank. Since the VTUG was designed for short duration missions, and emphasis was laid on high maneuverability and speed, this configuration of the BCS was chosen. Figure 5.2(b) shows the schematic of the BCS. Components of the BCS are given in Table 5.1.



(a) BCS assembly (b) BCS schematic

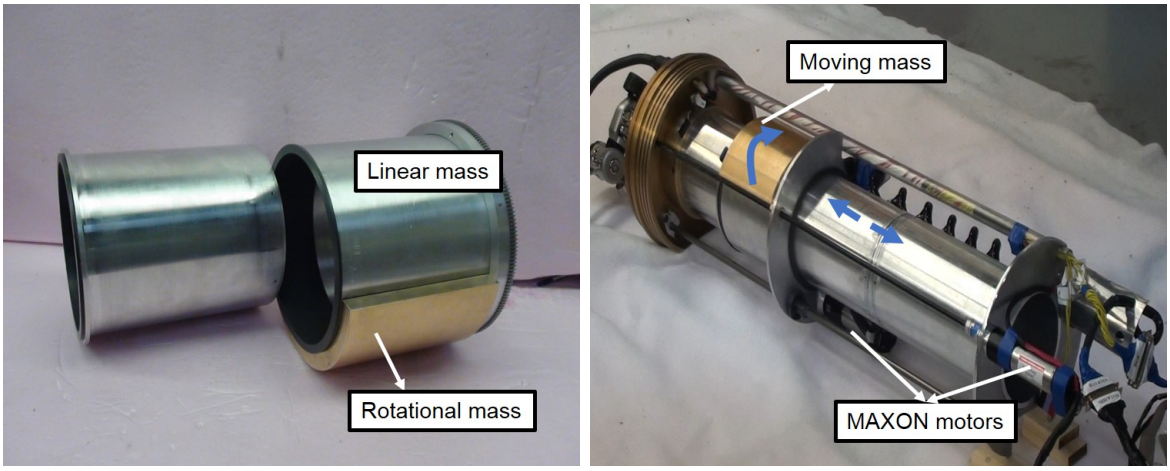
Figure 5.2: Buoyancy Control System ([1])
A. Wolek, J. Burns, C. Woolsey, J. Quenzer, L. Techy, and K. Morgansen, "A maneuverable, pneumatic underwater glider," in *Oceans, 2012.* IEEE, 2012, pp. 1-7. Used with permission of Artur Wolek, 2015.

Component	**Description**	**Model**
Scuba Tank	23 cu.ft.,steel	Faber FX23DV
Dive Regulator	Balanced diaphragm	Zeagle DS-V
Bladder pressure sensor	0 - 5 psid	GE UNIK 5000
Tank pressure sensor	0 - 5000 psia	GE UNIK 5000
Inflator Solenoid	$C_v = 1.0$	Enfield LSV-15
Deflator Solenoid	0.625 in. orifice	iQValves PFCV
Depth Sensor	0 - 200 psia	GE UNIK 5000

Table 5.1: Components of the BCS ([1])
A. Wolek, J. Burns, C. Woolsey, J. Quenzer, L. Techy, and K. Morgansen, "A maneuverable, pneumatic underwater glider," in *Oceans, 2012.* IEEE, 2012, pp. 1-7. Used with permission of Artur Wolek, 2015.

The ACS, configured as an internal moving mass (see Figure 5.3) uses two electric MAXON ECmax 30 brushless DC motors. One of the motors acts as a linear actuator of a horizontal moving mass for pitch control, while the other acts as a rotational actuator to rotate a mass for roll control. A significant difference in the design of VTUG in comparison with legacy

gliders [8], is the fact that it has unlimited roll authority. This was incorporated into its mechanical design as a novel feature in order to exploit various asymmetric wing profiles [1].



(a) Moving mass (b) Moving mass illustrated on an open glider

Figure 5.3: Attitude Control System

Figure 5.4 is a CAD model of the glider which illustrates its major internal components and modules.

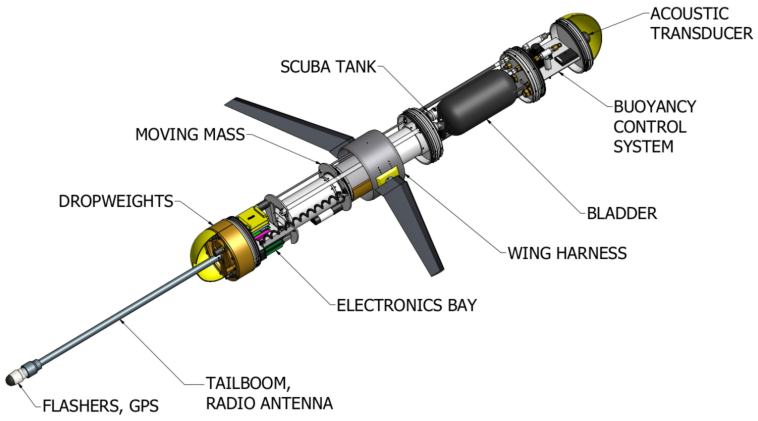
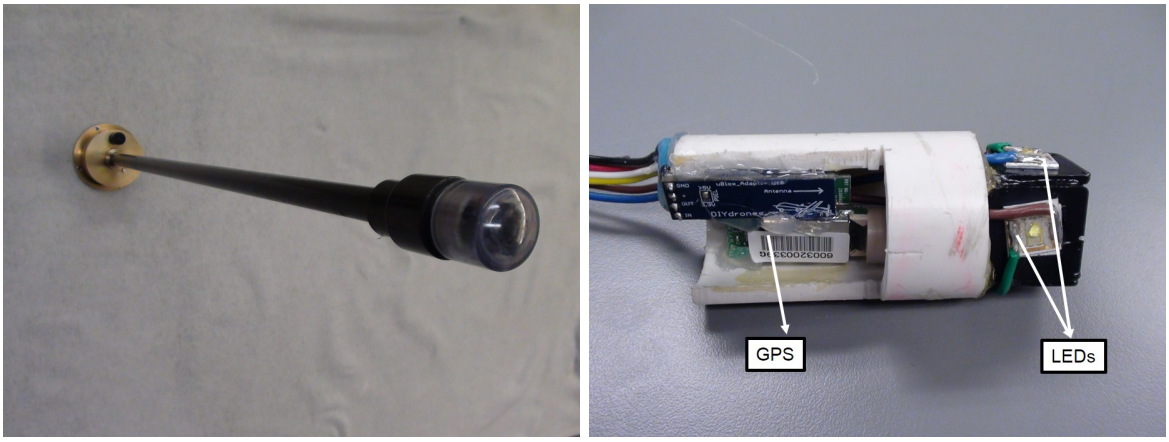


Figure 5.4: CAD model of VTUG ([1])
A. Wolek, J. Burns, C. Woolsey, J. Quenzer, L. Techy, and K. Morgansen, "A maneuverable, pneumatic underwater glider," in *Oceans, 2012.* IEEE, 2012, pp. 1-7. Used with permission of Artur Wolek, 2015.

Some significant modules are summarized here; for a thorough review refer to [1],

- **Tailboom:** Hollow tube made of PVC material, the tailboom measuring about 3ft houses the radio and GPS antennas, along with LED lights which can be configured to send different types of visual signals for retrieving or other monitoring purposes. Components of the tail module are given in Table 5.2.



(a) Tailboom with LED flashers (b) GPS and LEDs

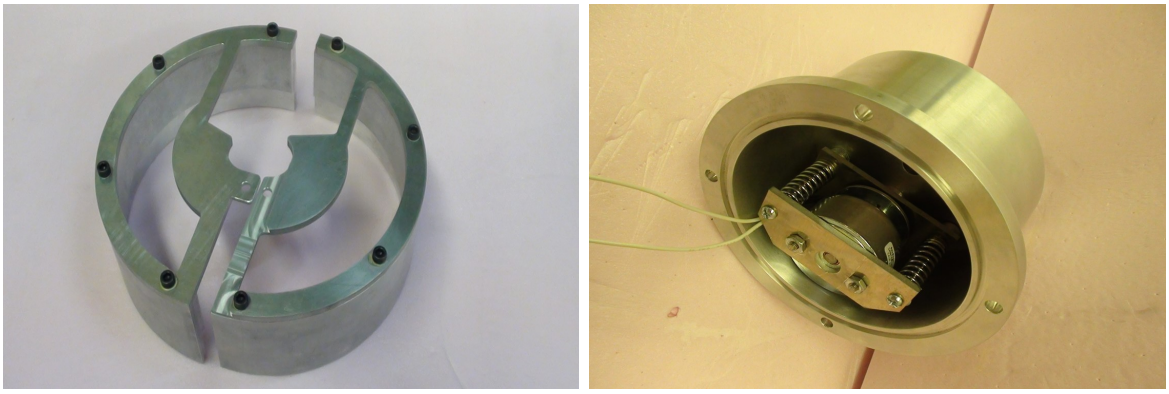
Figure 5.5: Tailboom

Component	Description	Model
GPS	FTDI interface, 2Hz refresh rate (NMEA). Sarantel omni-directional Geo-helix S-type active antenna. Accuracy: $\sim$ 10m	GS407 U-Blox5H
Radio	900 MHz. 60 mile range. 115.2/153.6 kbps speed.	Freewave FGR2
LEDs	Cool White (6500K). 180 lm @ 700mA	Luxeon Star LEDs

Table 5.2: Components of the Tail Module

- **Dropweights:** These are weights calibrated to make the glider buoyant, and float up to the surface when released. They act as recovery aids in the event of an emergency,

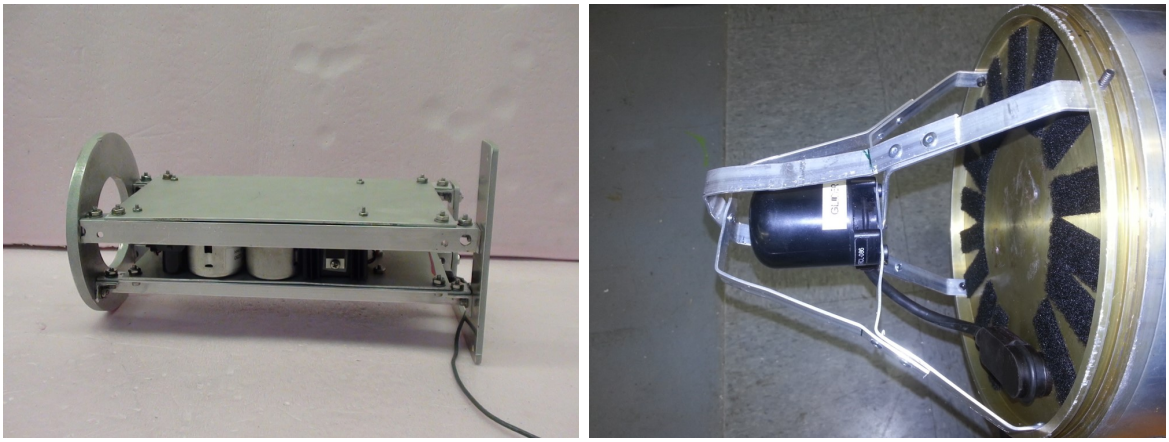
such as a leak. A software signal is automatically sent to actuate a solenoid and release the weights.



(a) Dropweights (b) Dropweight solenoid

Figure 5.6: Dropweight system

- **ACOMMS** (acoustic communication system): Consists of an acoustic modem, and an acoustic transducer. The transducer successively pings the 3 nodes of the LBL system, and the modem decodes the received pings, and provides the one way travel time. The transducer is housed in the nose cone. A protective housing made of thin aluminum plates is used to protect it. The ACOMMS functions and features will be discussed in detail in Section 5.3.



(a) WHOI modem housing (b) Transducer mount on nosecone

Figure 5.7: ACOMMS

- **Wing harness:** The wing attachment mechanism was specially designed to allow for asymmetric wing profiles. The position of the wings on the body, as well as the dihedral angle can be adjusted as shown in Figure 5.8.

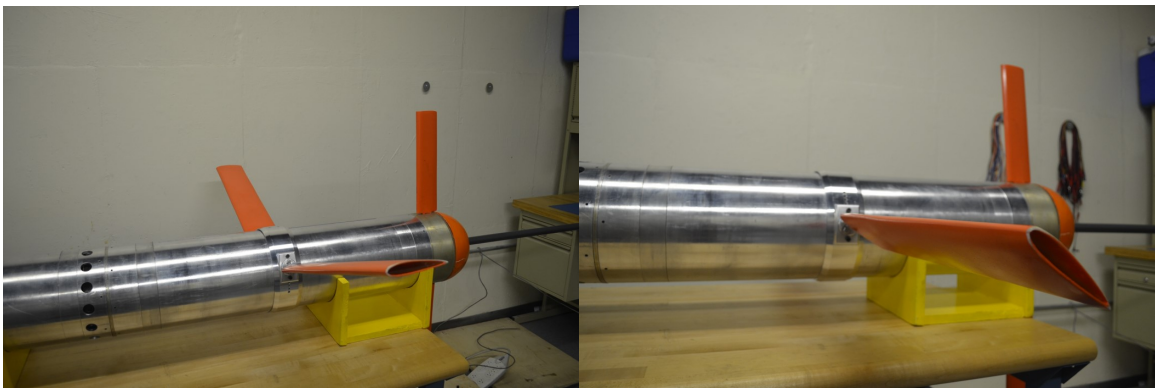


Figure 5.8: Wing harness customizations

- **AHRS:** The attitude and heading reference system is fairly simple. It consists of an IMU (Microstrain 3DM-GX3-25) mounted on the hull of the glider as shown in Figure 5.9. It provides roll-pitch-yaw angles, and the respective angular velocities, and

accelerations. The heading (yaw) angle provided by the AHRS is used in the dynamic estimation (EKF) algorithm of Section 4.2. The IMU is mounted slightly off the main body axes, and hence its readings need to be calibrated to align with the body axes for estimation.

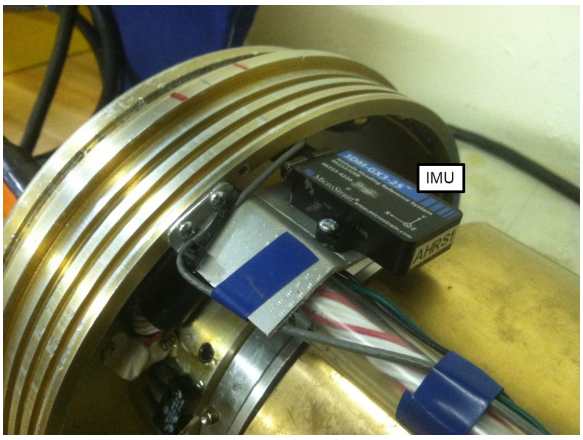


Figure 5.9: Microstrain IMU mount

- **Electronics Main Bay** (EMB): The EMB houses the on-board computer and power system. Table 5.3 gives a overview of sensors and components of the EMB.

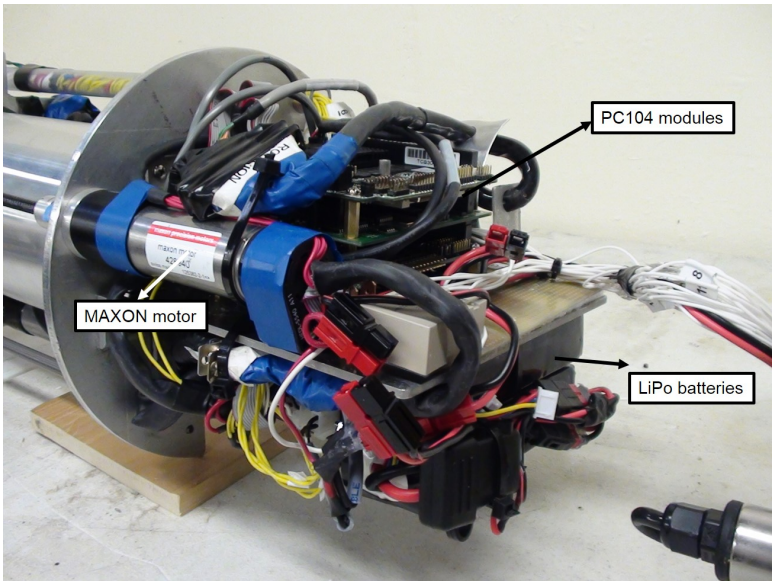


Figure 5.10: Electronics Main Bay

Component	Model
Computer	Diamond Systems PC-104 (stack of three components) - **Pegasus single-board computer:** 500 MHz, low-power, 256 MB RAM, compact flash. - **JUPITER-MM-SIO:** 50 Watt DC/DC Power supply with 2 serial ports. - **DMM-16-AT data acquisition board:** 16-channel, 16-bit Analog I/O with Auto-calibration.
Power	2×5400mAh 11.1V (nominal) LiPo batteries
ACOMMS	WHOI Micro-Modem (stack of three components) - **Micro-modem:** 201002c, version 1.3 - **Power Amplifier:** RFF 202004 - **Co-Processor:** 203004
Motors	EC-max 30, Brushless DC motor (60 Watt)
Depth Sensor	GE UNIK 5000

Table 5.3: Components of the EMB

The software used on-board the PC-104 is called the Robot Operating System (ROS), which is mounted on Ubuntu Linux 12.04. ROS is a community driven, open-source software with many software packages and drivers designed to integrate various hardware components used in robots. It is relatively easy to learn for a new user and has a simple architecture. The same software revision runs on the glider as well as all the 3 LBL nodes. The software design using ROS was done by Laszlo Techy and Jake Quenzer from University of Washington. The software design of the glider is beyond the scope of this project, however its basic components used for LBL positioning will be discussed in Section 5.4.

While on a mission, communication can be established with the glider in three ways,

1. **Ethernet:** This is done by attaching a wet-mateable cable to the tail of the glider. The on-board glider computer has an Ethernet port with 10/100 Mbps speed. This mode of communication is also used to update the software on the glider. As expected

while connected via Ethernet higher data transfer speeds are achieved. ROS allows for build files to be directly copied onto the glider PC104, hence saving precious time in building code on the slower 500Mhz processor.

2. **Radio:** The radio antenna is housed in the tailboom. A serial network is setup for one-to-one radio communication with a ground station. The socket has to be pre-configured, and a necessary ROS node must be running to successfully establish radio connection. Radio contact is not possible when the glider is underwater, as RF frequencies above 1MHz are severely attenuated within a few feet of hitting the water surface. Hence, to establish radio contact the glider must be on the surface and should "stick" its tail out as shown in Figure 5.11. This particular glider orientation is called a "comms stance." A software command can be sent to maneuver the glider into a comms stance. Typically this requires the vehicle to be positively buoyant, and the linear moving mass to be moved all the way to the front of the glider hull.

3. **Acoustic:** While underwater, the glider acoustically communicates with the Ground station. Since acoustic communication is very slow, and prone to interference, it occurs infrequently and only provides critical vehicle health information. The acoustic communications (ACOMMS) module consists of the WHOI Micro-Modem and the acoustic transducer. The glider periodically (every 1 min.) sends a 32byte coded message to the ground station. These messages are called heartbeat messages. Each heartbeat message contains:

   (a) Depth

   (b) Last recorded Lat/Lon from GPS

   (c) Heading

   (d) Tank pressure

(e) Battery Voltage

(f) Leak sensor info

Figure 5.11: Glider Comms Stance

Table 5.4 gives the key glider specifications.

Body Length	6.3 ft
Full length (w/tailboom)	9.3 ft
Diameter	9.0 in
Mass	56 kg
Buoyancy displacement capacity	0 - 5L
Endurance	6 hours
Roll control authority	unlimited
Pitch control authority	±70 deg

Table 5.4: Key glider specifications (Permission granted by Artur Wolek [1])

This section serves as a summary of the glider design, and is meant to give an understanding of the various components used, and how they are integrated. Section (5.2) will be dedicated

to the electrical and mechanical design of the three LBL nodes. It will be duly noted that the LBL nodes use the same components for communication as the glider. The glider itself acts as a (master) node, with the beacons configured as slave nodes in the acoustic sensor network for LBL ranging.

5.2 LBL Node Design

The LBL positioning system consists of three nodes/beacons built to serve as part of the acoustic network. The electrical hardware of the beacons is housed in a rugged Pelican case. Each pelican case is deployed onto inflatable rafts, and the rafts are anchored to the floor of the Claytor Lake, VA. Figure 5.12 shows the image of a beacon deployed on the lake.

Figure 5.12: Beacon deployed on a raft

One of the beacons is used as a ground station node, and is always with the vehicle operators aboard a moving vessel. Two of the nodes are deployed as shown in Figure 5.12. Henceforth the ground station node will be referred to as GS, and the remaining beacons as B1 and B2.

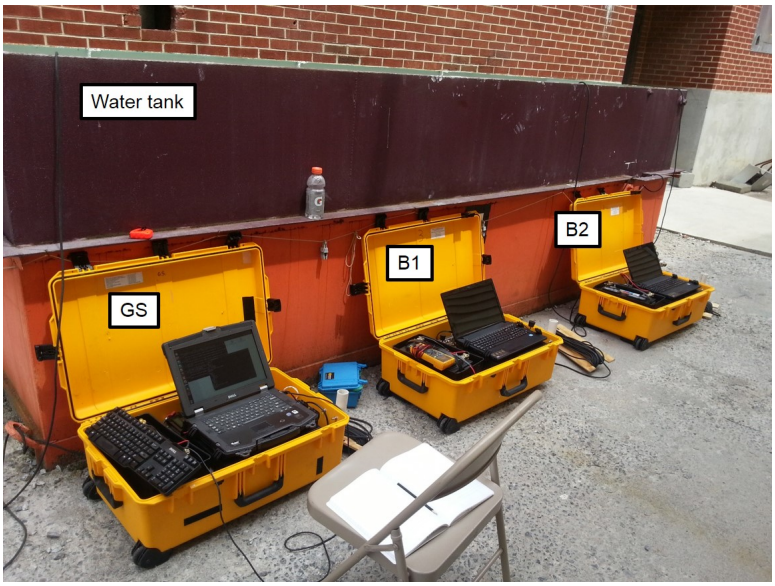


Figure 5.13: All 3 beacons during an ACOMMS tank test

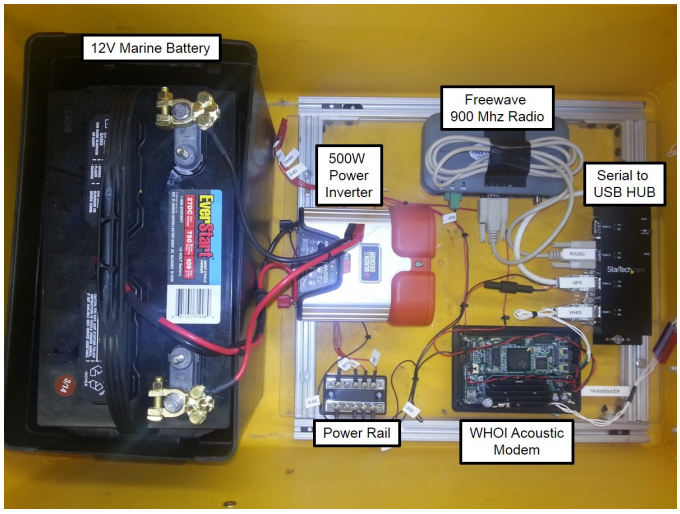
Figure 5.13 shows the three nodes during a calibration test of the acoustic transducers in a water tank at Virginia Tech. The GS is designed a little differently than the other two nodes. Along with all the hardware to perform acoustic positioning it also contains glider specific devices such as a junction box to establish Ethernet connection, and a powerful fiberglass antenna to establish radio connection with the glider. It is also used to start up the glider while on site.

The electrical components are housed in an aluminum 80×20 frame. The frame has two levels. Figure 5.14 illustrates the two levels as well as the antenna and the acoustic wet-mateable cable connected to the acoustic transducer.

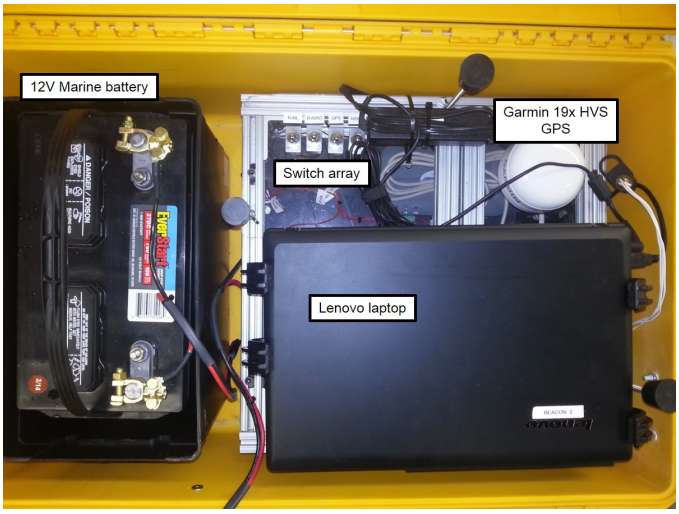
The nodes are powered with the help of a 12V marine deep-cycle battery. The AC power to the laptop is provided using a 500W Black & Decker inverter. A power rail is used to centrally provide power to the radio, GPS and acoustic modem (and transducer). All three of these components run on 12V power supply and output serial data. A 4×1 serial (RS-232) to USB hub from StarTech converts them to a single USB connection to the laptop. Hence,

all the data is accessible in a single format. There is also a switch array to control power to each of the devices separately. This way each component can be tested separately, and only components needed can be turned on, thus saving power. The laptops are loaded with the latest revision of the ROS software, and parameter files pre-configured (detailed in Section 5.4). A 25m Teledyne wet-mateable cable shown in Figure 5.14(c), is used to submerge the transducer at a predetermined depth. To prevent the cable from swaying due to currents, it was weighed down by running it through a 1" PVC pipe. The radio on-board the nodes can be used for inter-beacon communication. For this project this was not implemented as the data analysis was only done in post-processing. However, the software and hardware interfaces exist to establish inter-beacon RF communication.

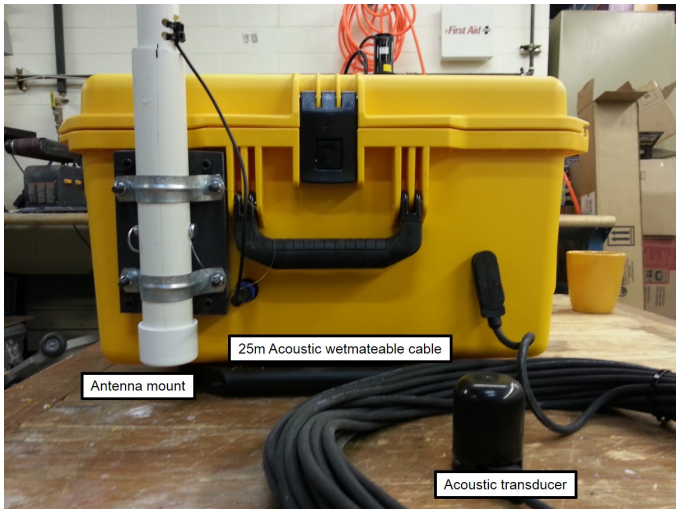
All the components within the Pelican case are friction fit using foam to prevent them from moving around. Table 5.5 gives a summary of the technical specifications of the various components in each of the three nodes.



(a) Level 0 of the beacons



(b) Level 1 of the beacons



(c) Antenna and acoustic transducer

Figure 5.14: Node hardware

Component	Description	Model
Battery	- 12V, 575 CCA (cold cranking amps). - 3hr reserve capacity	DieHard RV 27M
Inverter	500W, 2×110V AC outlets	Black&Decker 500
Laptop	- 1.5GHz, 4GB RAM, Intel i3-3120M. - Ubuntu 12.04 (Precise pangolin) - ROS Fuerte Turtle	Lenovo G580
Radio	- 900 MHz. 60 mile range. - 115.2/153.6 kbps speed.	Freewave FGR2
Antenna	- 12", 150W. Frequency: 896-940 MHz. - Nominal impedance: 50Ω	DATA-LINC (3db) Omni-Directional antenna: EAN0900WC
GPS	- FTDI interface, 2Hz refresh rate (NMEA). - Sarantel omni-directional geo-helix S-type active antenna. - Accuracy: $\sim$ 10m	GS407 U-Blox5H
Acoustic Modem	- 12 V, 4Khz bandwidth (25-29 KHz) - QPSK modulation - Texas Instruments 1000Mhz FPP DSP	WHOI Micro-Modem - **Micro-modem:** 201002c, version 1.3 - **Power Amplifier:** RFF 202004 - **Co-Processor:** 203004
Acoustic Transducer	- Resonance frequency: 28Khz - Max depth: 700m	BTech BT-1RCL
Wet Mateable Cable	25m shielded 20AWG 3 pin cable	Teledyne Belden 8412
Serial to USB hub	4 Port RS-232 to USB serial adapter	StarTech ICUSB2324I

Table 5.5: Node Hardware

5.3 Acoustic Communication System

The acoustic communication system consists of the WHOI Micro-Modem and the acoustic transducer installed on the glider and all the three LBL nodes. This section reviews the principle and working of these devices. It also gives the configuration settings for the WHOI Micro-Modem used to perform long baseline ranging.

5.3.1 WHOI Micro-modem

The WHOI Micro-Modem is a compact, low-power acoustic transceiver that has a modem-to-modem ranging capability, and can be configured to provide synchronous one way travel times, through an integrated precision clock. Its form factor and versatility make it ideal for integration in AUVs. It supports use of both broadband and narrow-band transponders for long baseline ranging systems. Along with providing one way travel time information, it is also capable of providing high-rate underwater acoustic communications [40]. For data transmission, it has the capability to perform low-rate frequency-hopping frequency-shift keying (FH-FSK) [42] , and variable rate phase-coherent keying (PSK) [43]. To interact with the modem hardware, a standard command set of NMEA 0183 sentences is used [44].

The WHOI Micro-Modem model 250012-0119, is a three stack 25kHz modem system. The three stacks are shown in Figure 5.15, and their brief description is given below,

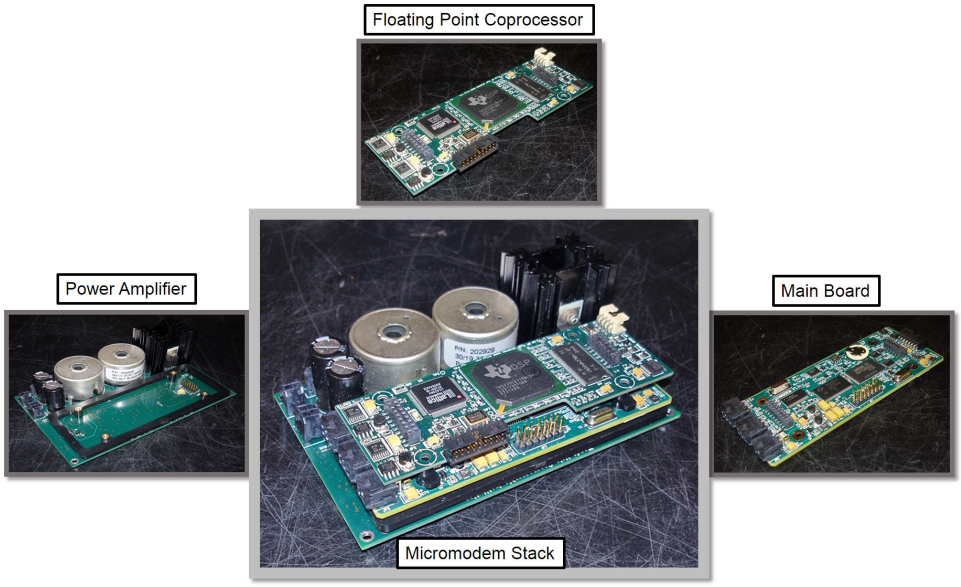


Figure 5.15: WHOI Micro-Modem

1. **Power Amplifier:** The power amplifier provides efficient and easily modifiable power for use with various underwater vehicle power systems. It is designed as a single channel class D amplifier. A class D amplifier has superior efficiency, and ease of impedance matching with several transducers. It provides initial power conditioning circuitry with many safety features built in to avoid failure. For a typical application about 50W of power is required to transmit 190dB at 1uPa of sound pressure with an omnidirectional transducer [43]. The power amplifier is used only when transmitting, other times the power amplifier is nominally shutdown. It's design can support use with multiple transducers. In the case of this project, only one transducer is used per modem, and while transmitting a power of ˜15W is used.

2. **Main Board:** The main board is compact and designed to use low power. The real time operating system of the Micro-Modem is implemented on a Texas Instruments TMS320C5416 DSP [40]. A 12 bit A/D converter samples the received signals, typically at 80kHz [43]. The analog output is obtained via a 12 bit D/A converter and a

low pass filter. An on-board real time clock provides timing and synchronization. For precision timing a PPS (pulse-per-second) signal can be provided by the user. The user can adjust various parameters such as analog gain, frequency band, type of data modulation (FSK,PSK), packet transmission rate etc., using standard NMEA sentences, sent through an RS-232 serial port.

3. **Floating Point Coprocessor:** The coprocessor is an add-on board, specifically to support computationally intensive PSK equalization algorithms. Even without the coprocessor, the Micro-Modem can perform ranging and provide one-way-travel-times. However, only FSK can be used, and high transmission rates cannot be obtained. A Texas Instruments TMS320C6713 high speed floating point DSP with fast SDRAM enables high speed processing. Using the coprocessor consumes as much as 2W when active [43].

The Micro-Modem wiring diagram for the purposes of LBL ranging is shown in Figure 5.16. It is fairly simple. The coprocessor wiring is not shown as it was used in its default basic configuration, with which it arrived.

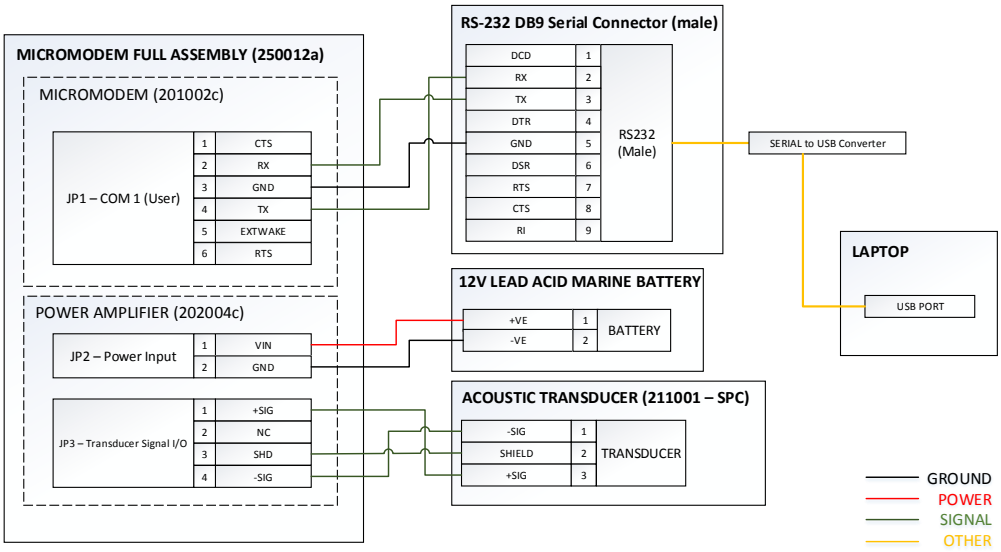


Figure 5.16: Micro-Modem wiring

The data transmitted by the Micro-Modem is broken up into packets. Each packet contains a frame of data. Depending on the data rate, there may be more than one frame within a packet. High rates contain multiple frames. The length of a frame is variable, and is measured in terms of bytes. The FSK packet contains one 32 byte frame. At the lowest-rate PSK data rate there are three 32-byte frames per packet. The highest current PSK rate has eight 256-byte frames per packet. The data rates and packet formats are listed below in Table 5.6 [44].

Packet Type	**Num. Frames (max)**	**Frame Size (bytes per frame)**	**Total Payload**
1: FSK	1	32	32 bytes
2: PSK	3	32	96 bytes
3: PSK	**3**	**64**	**192 bytes**
4: PSK	2	256	512 bytes
5: PSK	8	256	2048 bytes

Table 5.6: WHOI Micro-Modem packet types and rates

The PSK data rate marked in red, is used for data communication in this project. Apart from these (long) packet types, the Micro-Modem also provides for short packet types which are 21 bytes in size, called *mini-packets*. These mini-packets are of different types based on their function. The types specifically used during an LBL ranging cycle are,

1. **Ping:** A ping mini-packet is sent by the glider to the nodes. The ping packet contains information about the source and the destination. The node specified as the destination responds with a fixed turn-around time. The glider uses this information to calculate the one-way-travel time to that specific node.

2. **Data Acknowledgment:** This mini-packet is sent automatically from a node when it successfully receives data from the glider. The glider data frames must have the acknowledgment flag enabled to receive these.

5.3.1.1 Synchronous Transmission

To provide accurate positioning of the nodes, the modems provide for synchronous transmission with the help of a reference clock. By using clock-referenced signals, the time of arrival (TOA) can be calculated, and hence localization can be performed in an acoustic network. To implement synchronous transmission, and provide accurate and precise timing, the Micro-Modem uses an external pulse-per-second (PPS) signal, along with it's A/D converter sample clock. Between successive PPS signals, the A/D clock provides timing with a resolution of 12.5 microseconds. The data is processed after demodulation to 8 kHz sampled base-band, which results in a measurement resolution of 125 microseconds, or 18.75 cm using a sound speed of 1500 m/sec [45].

The timing diagram obtained from [45] is shown in Figure 5.17. The message signal is sent on a rising edge of a PPS signal. A message need not be sent after every PPS rising edge.

The PPS signal is initialized every time the modem boots. In case of failure of the PPS signal the modem reports different error messages based of cause [45]. The time limits are,

$$t_o : t_0 \geq 1\,\text{milliseconds}$$

$$t_1 : t_1 \geq 50\,\text{milliseconds}$$

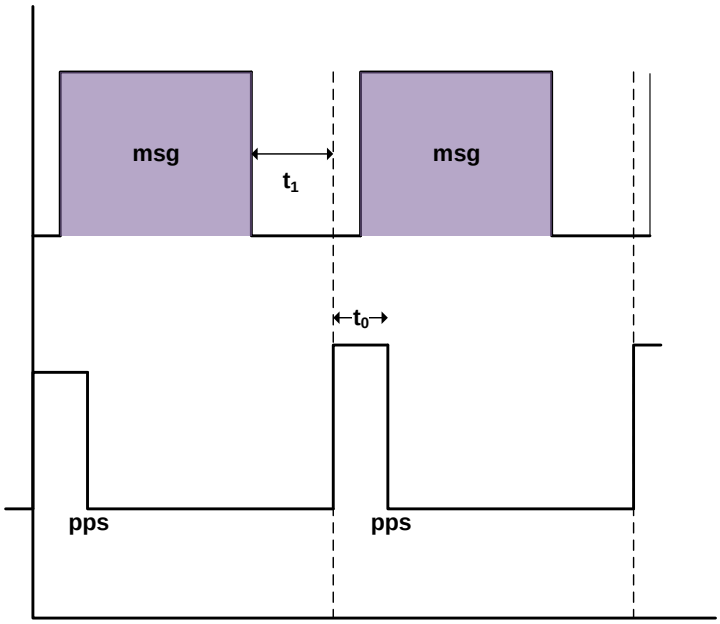


Figure 5.17: Timing diagram for synchronization

The NMEA command sentences used as part of the LBL ranging system are described as part of the software design in the next section. For all the commands available refer to the software interface guide [44]. The modem has numerous parameters that can be set. These parameters define the gains, type of packet, data rate, bandwidth, frequency range etc. The important parameter settings are also described as part of the software design.

5.3.2 Acoustic Transducer

The acoustic transducer is the sensor used to send and receive acoustic data. It is submerged in the water column using a wet-mateable cable which is run along a PVC pipe to weigh it down, and prevent swaying due to underwater currents (Figure 5.18(b)). In this project, a BTech-1RCL model transducer was used (Figure 5.18(a)). The resonance frequency of the transducer is 28kHz, and the nominal range of operation is 20 - 40kHz. The beam pattern is omnidirectional.

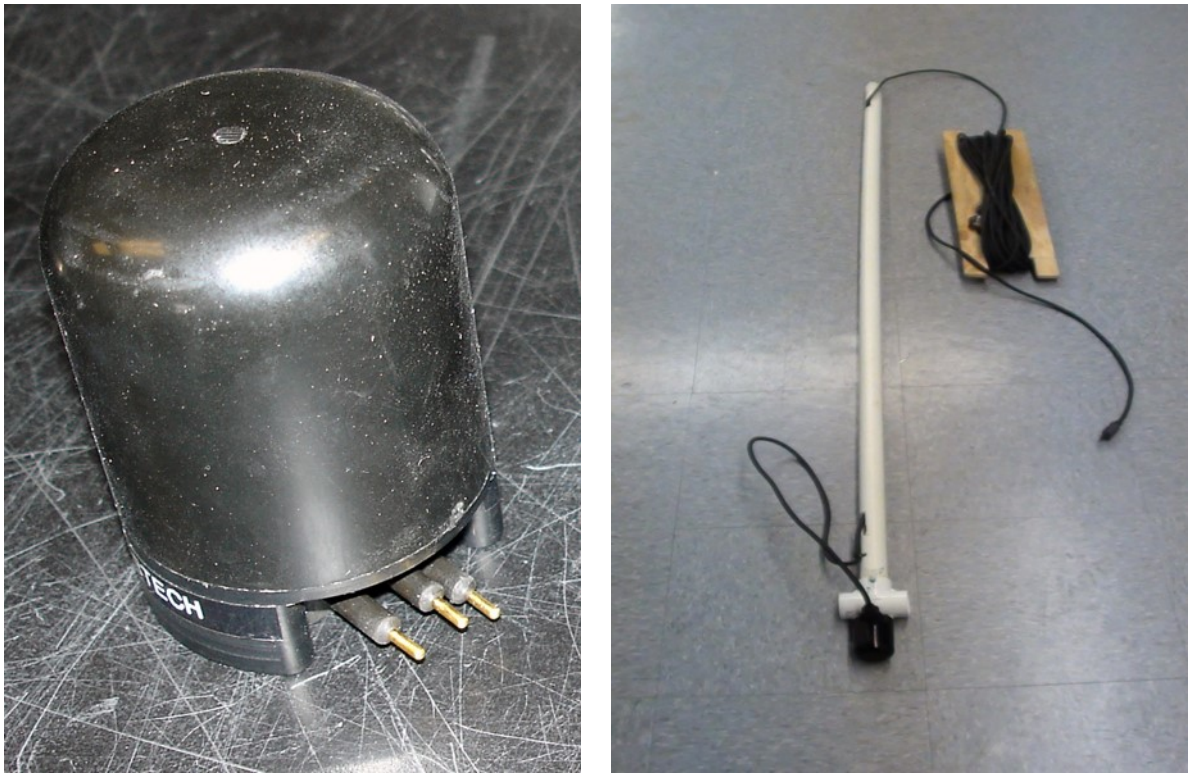


Figure 5.18: BTech Transducer and cable

The WHOI Micro-Modems can be operated in 4 different frequency bands. Three of these are predefined for use with FSK (A,B, and C). The 0^{th} band, which is a user-defined band is used for PSK. Table 5.7 lists the frequency bands available, each with a bandwidth of ˜4kHz.

To define the PSK band the parameters BND, FCO, and BWO are used to set the frequency bank, starting frequency and bandwidth respectively. The bands are equally divided into bins, each separated by 250Hz [44].

Frequency Bank (BND)	**Band**	**Frequency Range**
1	A	7.6 – 11.8kHz
2	B	12.8 – 16.9kHz
3	C	23 – 27.2kHz
0	**BND = 0** **FCO = 25000** **BWO = 4000**	25 – 29**kHz**

Table 5.7: Frequency bands

The 0^{th} band (marked in red) is used for the LBL setup. It can be observed that the frequency band of 25-29kHz was chosen as the resonance frequency of the transducer lies within this band. Intensive testing with both PSK and FSK settings was done. It was observed that the PSK setting had a higher endurance, robustness and efficiency in the field.

5.4 Software Design

The software used on the glider PC104 board as well as the Lenovo laptops is ROS Fuerte. It is mounted on Ubuntu 12.04. The Robot Operating System (ROS) is a flexible, open source framework for writing robot software. It is basically a collection of tools, libraries and conventions that aim to simplify the task of creating software for complex robotic architectures. It comes with powerful and relatively easy to use developer tools which can help build a prototype quickly and test it. Due to it being open source, the source code for hundreds of robotic projects built with ROS can be examined and studied. This helps in quickly understanding and implementing best practices for robot software. In this project

Qtcreator was used as the tool for version and repository management.

The initiative to use ROS as the primary software for the glider was taken up by Lazslo Techy, from Virginia Tech. The integration of the goby libraries and drivers with ROS for VTUG was done by Jake Quenzer, from University of Washington.

Software in ROS is organized into *packages*. Packages are the smallest individual things that can be built with ROS, and it is the way software is bundled for release. A package might be a ROS node (described later), a library, a dataset, configuration file, a third party piece of software such as a GUI, or anything that logically constitutes a useful module.

ROS architecture is based on graph concepts. It is built on the grounds that every robot's functions can be divided into nodes, like those in a graph. ROS starts with a master node called *roscore*. It is a collection of nodes and programs that are prerequisites of a ROS-based system. The master node allows all other nodes to find and talk to each other. It is executed first, followed by the other nodes. A node is basically an executable file within a ROS package. It performs a specific computation within the robot. For example, a communications node collects data from all the communication hardware such as radio, acoustic modem etc, a motor control node controls the motor actuators to drive them, and a guidance node implements guidance algorithms. The node talks to the hardware using drivers, which have been built for them using a ROS package. Nodes talk to each other using topics, or remote procedure call (RPC) services. Topics are named buses over which nodes exchange messages. Messages are a type of ROS data-type. A node can subscribe or publish a topic. Subscribing to a topic means getting a message from another relevant node. For example a communications node (comms) subscribes to data from the acoustic communications node (acomms). Publishing a topic means to generate messages which can be used by another relevant node. For example, acomms node publishes modem one-way-travel-times to the comms node. Every code for a ROS node begins with information about

the topics it is publishing and topics it is subscribing to. Topics are unidirectional, streaming communication. If a node wants to perform RPCs, it uses services instead. In case of services a node can request data and get a response back. Hence, they are bidirectional. Table 5.8 gives a brief of the ROS architecture, and Figure 5.19 illustrates basic ROS architecture.

Component	**Function**
package	smallest unit of software that can be built and released
roscore	master node that is a prerequisite for a ROS based system
node	executable that performs a specific computation
topic	named bus over which nodes exchange messages
service	RPC request reply interaction

Table 5.8: ROS architecture

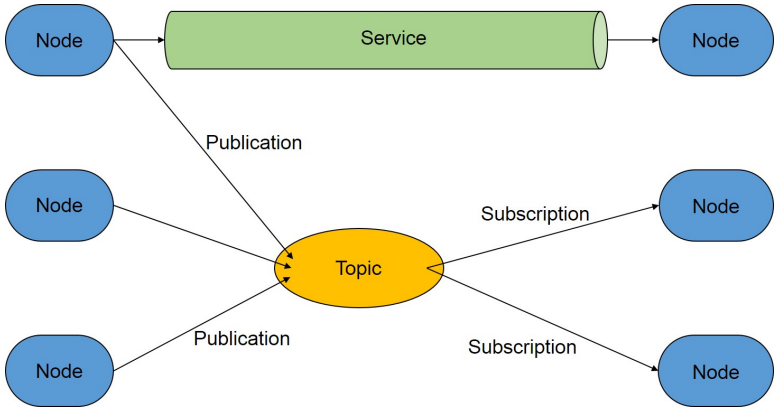


Figure 5.19: Illustration of ROS architecture

Before describing the software design of the LBL positioning system, it is important to mention a key ROS library specifically developed for acoustic communications called *goby* [46]. This open-source ROS library, compatible only with the WHOI Micro-Modem was developed under an open-source project called "Goby Underwater Autonomy Project." The head architect of this project is Toby Schneider. The guide to use the *goby* ROS library is documented in [47], and the dynamic compact control language (DCCL) that was specifically developed for acoustic communications with the WHOI modem is documented in [48]. The

details of the design of goby software and the DCCL is beyond the scope of this project, refer [46] and references therein for more information. The integration of the goby libraries and drivers with ROS for VTUG was done by Jake Quenzer, from University of Washington.

The LBL positioning system developed for the VT glider uses a node and topic structure illustrated in Figure 5.20. Apart from these the glider has many other nodes for motor actuation, buoyancy control, guidance, monitoring etc. It can be observed that there are a total of 6 nodes and 6 topics. The nodes and the topics they publish and subscribe is described below:

1. **/acomms**: The acomms node talks to acoustic modem and gathers data from it. The modem is connected to the acoustic transducer as shown in Figure 5.16. The modem issues commands to the transducer on the glider to ping each beacon successively, and computes one way travel time. It publishes the topic /modemRx to the comms node. This data message contains the one-way-travel-time, source, destination, sequence number etc.

2. **/comms:** The comms node gathers data from the radio and modem, and combines them into a message and publishes the topic named /commsLBLData. For doing this it needs to subscribe to /radioRx and /modemRx.

3. **/sensor_node:** The sensor node subscribes to the topics published by all the sensors on-board the glider. For the LBL system, it subscribes to the /GPSfix topic for GPS data, and /commsLBLData described above. It gathers all this data into the /sensorData topic, which it publishes.

4. **/gpsdClient:** This node uses a custom made driver that talks to the GPS on-board the glider. The driver works only with certain GPS hardware available in the market. It publishes the /GPSfix topic to the sensor_node.

5. **/microstrain_3DM_IMU:** This node corresponds to the IMU. It collects data from the IMU at 1Hz. It publishes the /imu/data topic to the sensor_node. This topic contains the IMU orientation angles in the x,y,z directions of the body axis, as well as velocities and accelerations in these directions. Among these, the heading angle is of special interest and is used in dynamic estimation algorithms of Section 4.2.

6. **/data_collect:** The data_collect node is responsible for logging and storing data on-board the glider PC104 computer. The PC104 has an expansion slot for a compact flash storage device. Using this data can be collected during a run and analyzed after. For LBL data it subscribes to the /sensorData topic. The data is stored in tab-separated format with labeled header columns. The log files are time stamped to the second.

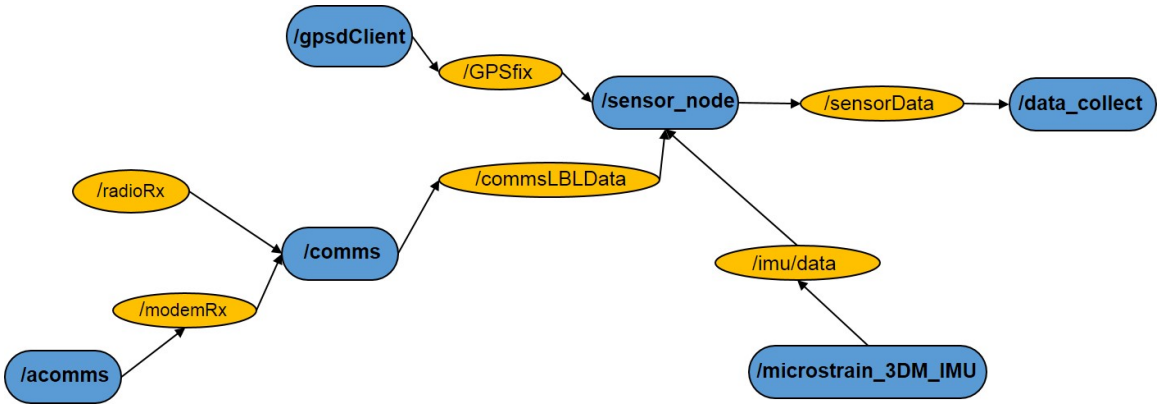


Figure 5.20: ROS nodes for LBL positioning

As mentioned before the WHOI Micro-Modem is interfaced with the ROS architecture using the goby software library. The following NMEA commands (Table 5.9) are used to configure the glider and perform modem to modem pinging.

Command	Description
$CCCFQ,NNN	Query NVRAM configuration parameter Eg. $CCCFQ,AGN: Prints the current value of Analog Gain Eg. $CCCFQ,ALL: Prints a list of all current configuration parameters
$CCCFG,NNN,vv	Used to set NVRAM configuration parameter, host to modem NNN: Name of the NVRAM parameter to set vv: New value Eg. $CCCFG,AGN,250: Sets the analog gain parameter to 250
$CCMPC,SRC,DEST	Mini-packet ping command To ping a beacon to compute one-way-travel-time SRC: Source (unit designated as ping originator) DEST: Destination (unit designated as receiver of ping)
$CAMPA,SRC,DEST	A ping has been received When a ping is received at the destination beacon, the modem sends this command to the user SRC: Source (unit designated as transmitter) DEST: Destination (unit designated as receiver of ping)

Table 5.9: NMEA commands

Other than the above NMEA sentences the modem also generates some acknowledgment messages as well as error messages in case of failure to transmit. Refer [44] for more information.

Figure 5.21 illustrates the UML (Unified Modeling Language) timing diagram for LBL ranging. The boxes with black border denote the ROS nodes involved.

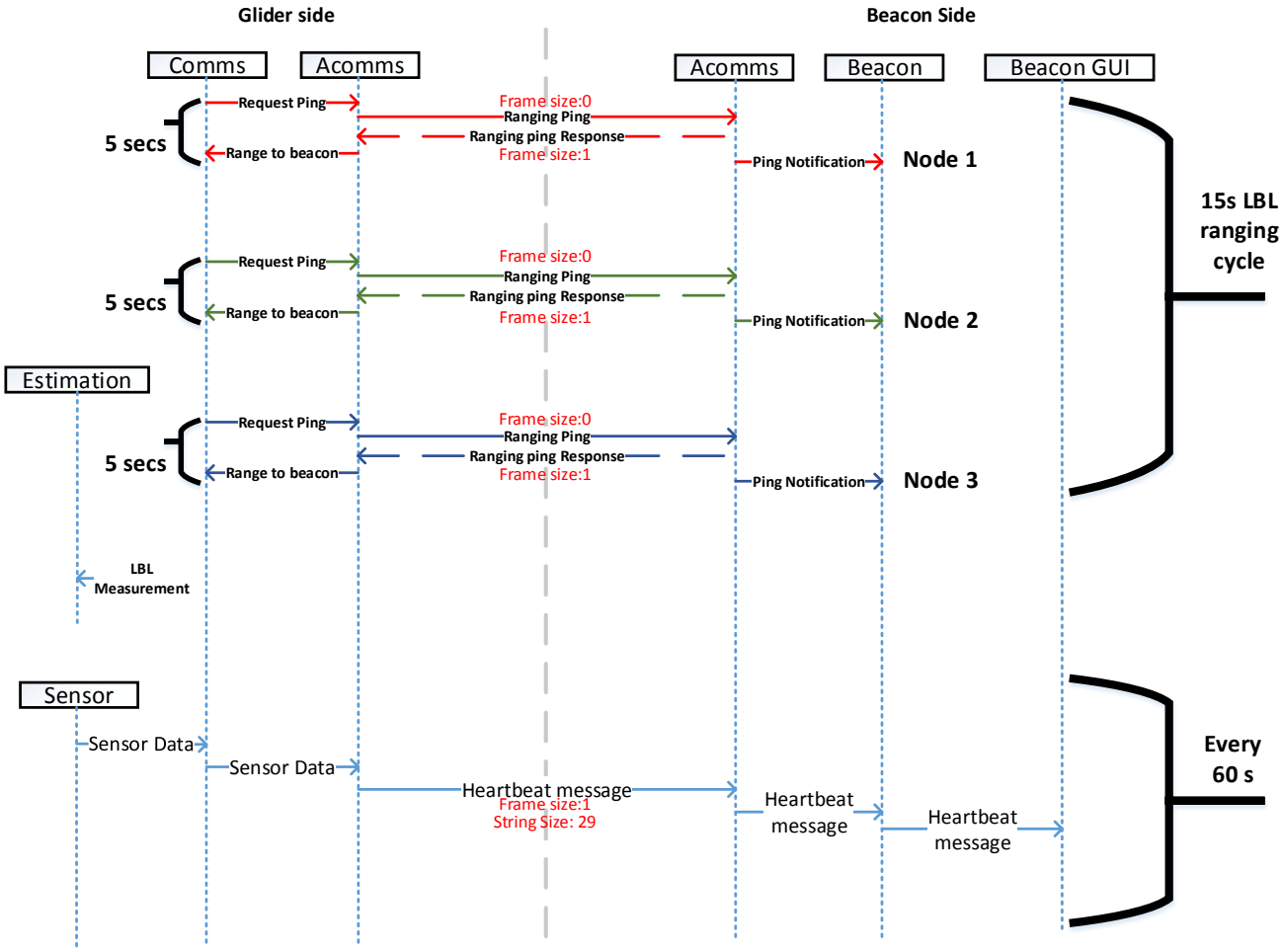


Figure 5.21: UML diagram for LBL ranging

Chapter 6

Experimental Results

This chapter describes experiments done at Claytor Lake, VA on 11th November 2014. Figure 6.1 shows the Claytor Lake map with the area of operation obtained from Google maps.

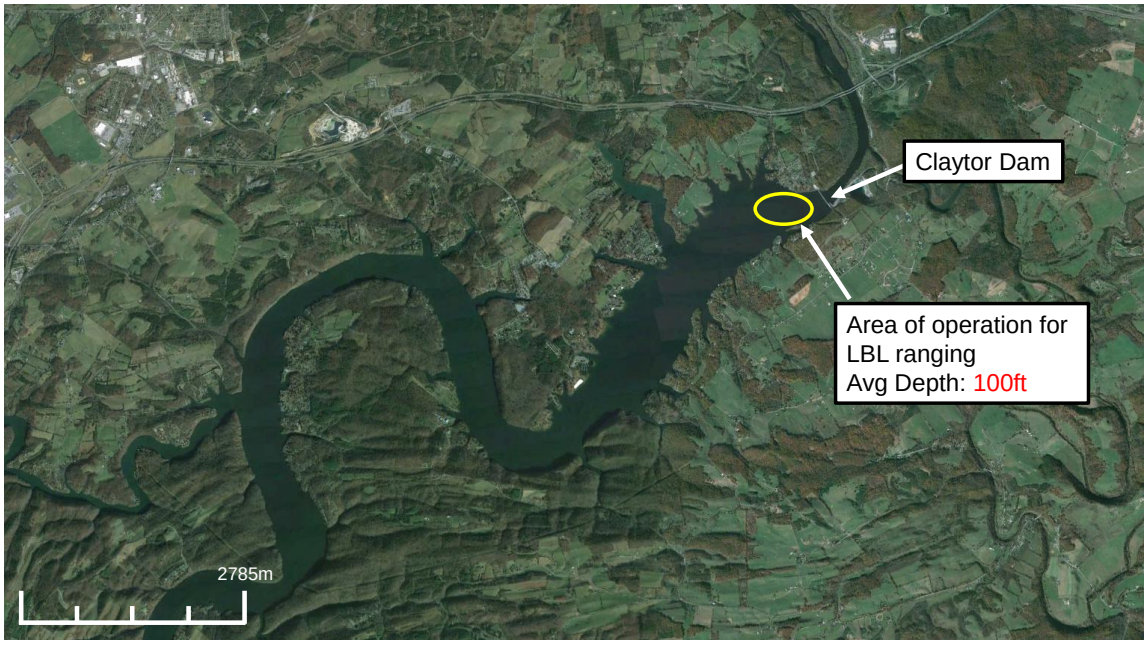


Figure 6.1: Claytor Lake, VA
https://www.google.com/maps/place/Claytor+Lake+State+Park/@37.0493806,-80.6255717,9573m/data=!3m1!1e3!4m2!3m1!1s0x884df07e7e75c0ed:0x989e45b25b94bc71

The experimental setup included the three LBL nodes deployed in their rafts, and the glider on the boat. The experiment was done as a validation test of the LBL system. During the test the glider was towed on the host vessel and the transducer was submerged in the water using the Teledyne wet-mateable cable. The depth of the transducer of the glider node was 5ft. The transducers of the rafts were submerged at a depth of 15ft. This setup yielded the best response to pings. Since the glider was on the host vessel, continuous GPS position of the glider was logged and recorded. The GPS positions of the floating beacons was also logged. Hence, acoustic range measurements could be compared with GPS measurements. This way the accuracy of the system could be established.

Only static estimation algorithms were implemented. Dynamic estimation algorithms could not be implemented as the glider did not perform any glides underwater. Hence, the model developed in Section 4.1 cannot be used. All the data was collected using the /data_collect node of ROS, and the processing was done offline.

The host vessel was moved in a lawn mower pattern to cover all the regions of the triangular topology of the node deployment. The glider path and node positions on the lake are shown in Figure 6.2.

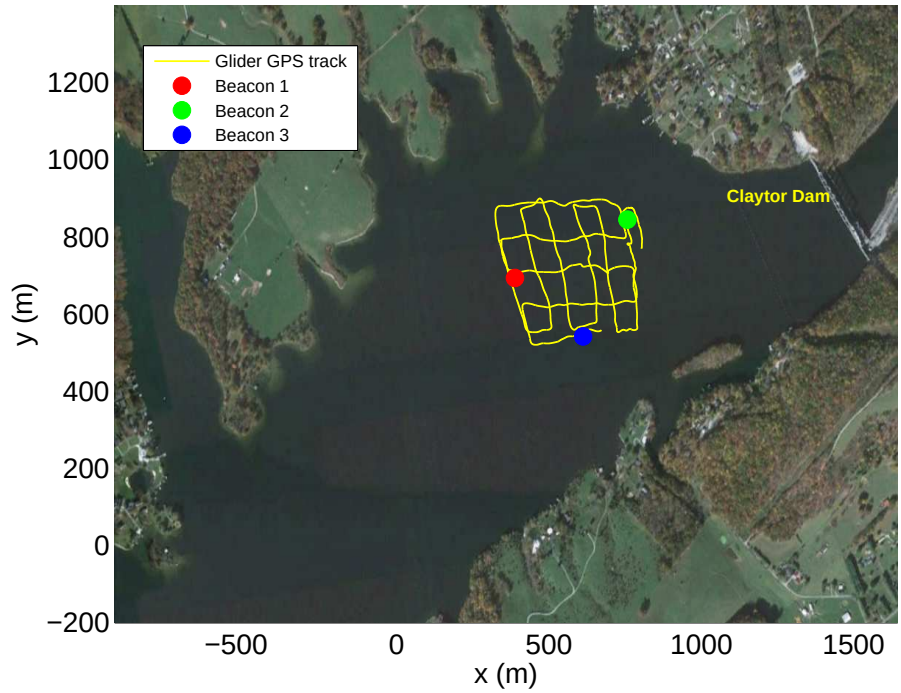
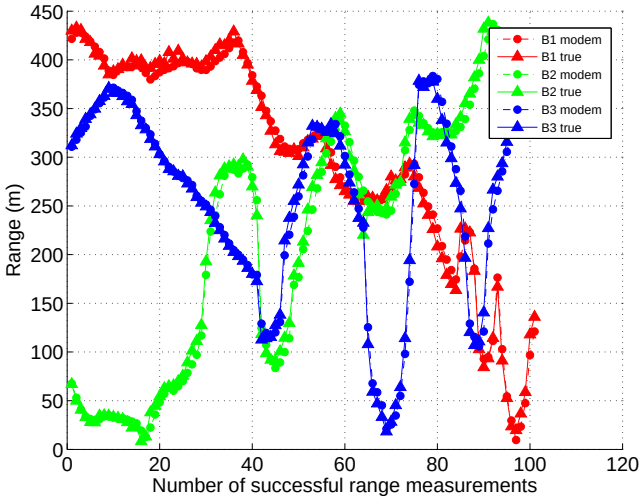
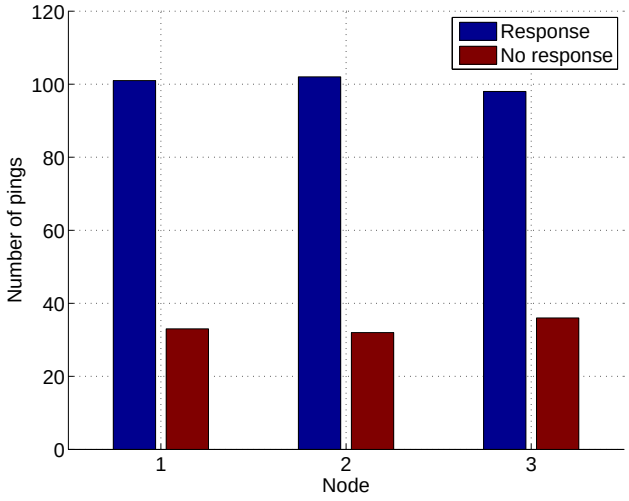


Figure 6.2: Glider path and node positions on the lake
https://www.google.com/maps/place/Claytor+Lake+State+Park/@37.0493806,-80.6255717,9573m/data=!3m1!1e3!4m2!3m1!1s0x884df07e7e75c0ed:0x989e45b25b94bc71

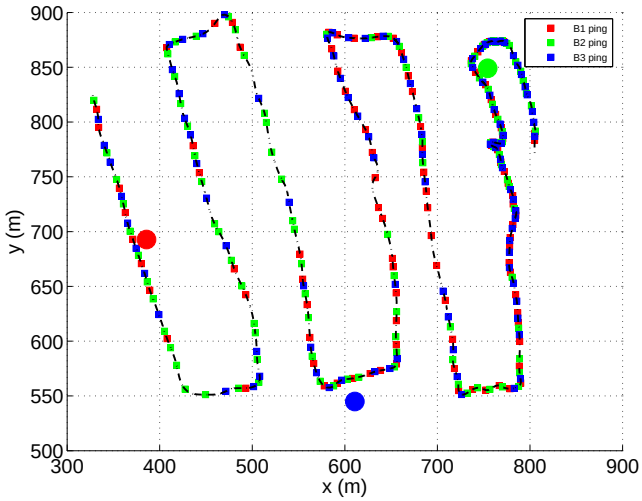
The full track of the glider (or boat) was divided into two separate tracks for analysis. The first track (Track 1) comprised of the vertical lawn mower paths, and the second track (Track 2) comprised of the horizontal ones. Figure 6.3 shows the estimation results of the vertical lawn mower paths.



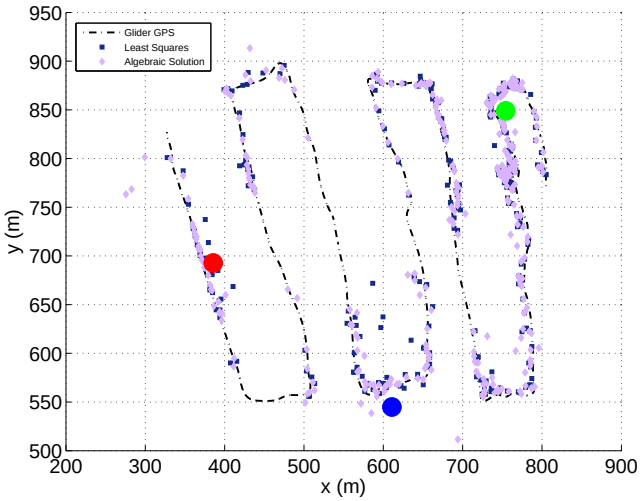
(a) True vs Modem ranges



(b) Histogram of ranges



(c) Ping locations on glider path



(d) Planar static estimation

Figure 6.3: Track 1 results

Figure 6.3(a) shows a comparison of the modem ranges and the true ranges obtained by using the GPS data from the nodes and glider. It can be seen that they both match quite well.

Figure 6.3(b) shows a histogram of pings with and without a successful response from the

nodes. For the duration of the run, 134 successful pings were expected from each node. However, only 101, 102, and 98 successful pings were obtained from Nodes 1,2, and 3 respectively. This shows some consistency in the performance of all the three nodes. However, the causes of the lost pings cannot be easily identified. It could be quite possibly due to the properties of the underwater acoustic environment. This is because extensive testing was done to ensure the robustness of the hardware and software, and in a smaller water body such as a tank no pings were lost.

Figure 6.3(c) shows the ping locations on the glider path. It was observed that three consecutive successful pings in the 15s LBL range cycle were not always obtained. Most of the time 2 successful pings in the 15s cycle were obtained. Whenever 3 consecutive range measurements were available localization could be performed with higher accuracy. However, two successive range measurements were sufficient to localize the glider position because,

1. Only planar estimation is being performed, since the depth of the transducers is known *a priori.*

2. Due to the availability of GPS data of the glider, one of the solutions obtained via the algebraic solution could be eliminated as knowledge of the baseline was available.

Figure 6.3(d) shows the results of implementing the Least Squares and Bancroft algorithms. The estimates with three range measurements and those with two are combined together in the plot shown. As expected, their performances are similar. The results of the run are detailed in Table 6.1.

Parameter	**Value**
Duration of run	34min
Number of expected pings per beacon	134
Avg. error* for LS with 3 range measurements	10m
Avg. error for LS with 2 range measurements	13m
Avg. error for Algebraic Solution with 3 range measurements	9m
Avg. error for Algebraic Solution with 2 range measurements	12m

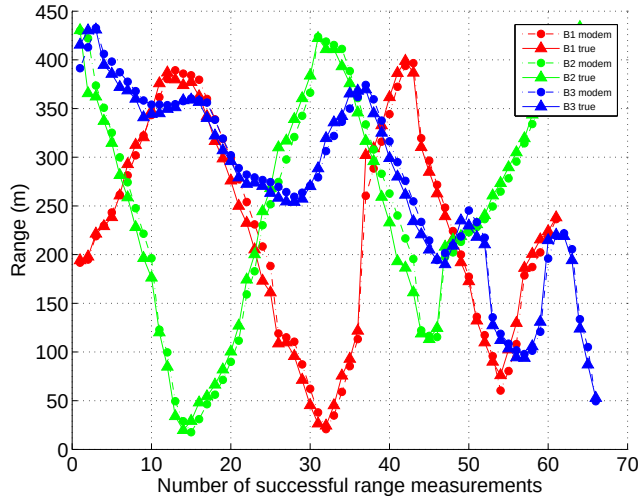
Table 6.1: Results of Track 1

*error: difference between range calculated using GPS, and range from WHOI Micro-Modems

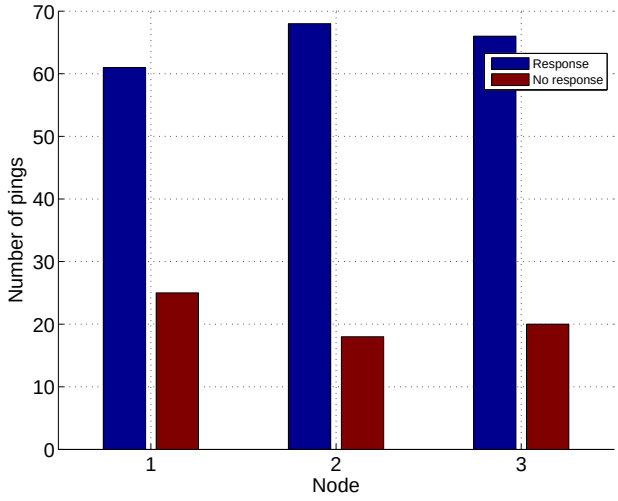
Figure 6.4 shows the results for Track 2. The results are comparable with Track 1. Results are detailed in Table 6.2.

Parameter	**Value**
Duration of run	22min
Number of expected pings per beacon	86
Avg. error* for LS with 3 range measurements	12m
Avg. error for LS with 2 range measurements	16m
Avg. error for Algebraic Solution with 3 range measurements	12m
Avg. error for Algebraic Solution with 2 range measurements	15m

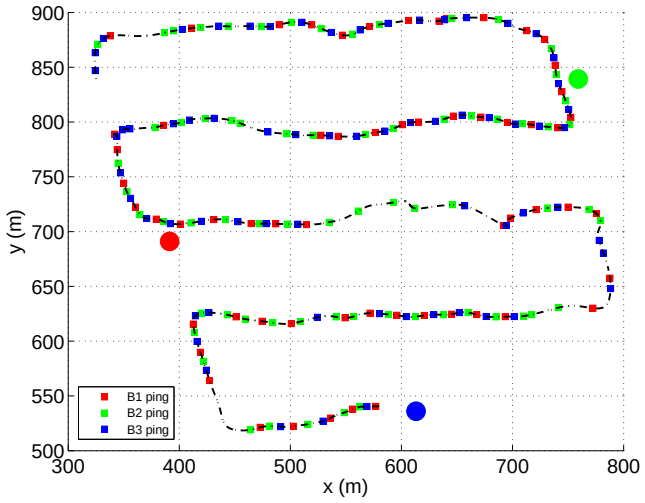
Table 6.2: Results of Track 2



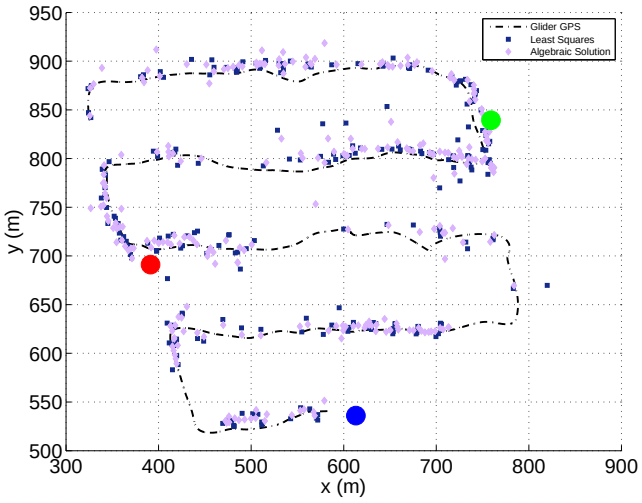
(a) True vs Modem ranges



(b) Histogram of ranges



(c) Ping locations on glider path



(d) Planar static estimation

Figure 6.4: Track 2 results

Chapter 7

Conclusions

7.1 Contributions

In this project a long baseline acoustic positioning system was developed and tested in a lake environment. The experimental results obtained so far served as a validation of the system. During these experiments the glider was towed on the host vessel with only the transducer submerged in the water using a long wet-mateable cable as described in Chapter 6. The glider did not perform any glides underwater while the LBL system was active. Since glider GPS data was also available for the whole run, the LBL range data could be compared with true data and observe its performance. All of the data processing was done offline.

Theory behind range-based position estimation of AUVs using static and dynamic algorithms was studied. Two 3D static estimation algorithms, namely an iterative nonlinear least squares algorithm, and the Bancroft algorithm which provides a direct algebraic solution were studied and implemented in Matlab simulations. They were also implemented offline with range data obtained from the experiment detailed in Chapter 6. Dynamic esti-

mation was studied using the EKF and RTS smoothing algorithms. These algorithms were only implemented in Matlab simulations as shown in Section 4.4.

Hardware and software integration of the components on-board the floating buoys was done successfully. The robustness of the system was verified. The endurance of the glider acomms system was thoroughly tested and the results were satisfactory. Endurance here refers to the amount of time the LBL system can be deployed without failure. The effectiveness and accuracy of the range data was established. The logistics of deployment and retrieval of the nodes have been well documented. Pre-flight and post-flight upkeep of the glider and the nodes has been standardized.

7.2 Future Work

Future tasks which could benefit the system in the order they could be implemented are listed below,

- Collection of LBL range data while the glider performs sawtooth glides underwater.
- Implementation of EKF and RTS smoothing estimation algorithms for range data offline (only simulations have been performed). Verification of their performance and comparison with static estimation algorithms.
- Testing the LBL system in a shallow coastal underwater environment with sandy bottom. Coastal environments have underwater currents, and the EKF algorithm developed is capable of estimating them.
- Implementation of the iterative least squares and the EKF algorithm as part of the ROS code and perform LBL localization online.

Bibliography

[1] A. Wolek, J. Burns, C. Woolsey, J. Quenzer, L. Techy, and K. Morgansen, "A maneuverable, pneumatic underwater glider," in *Oceans, 2012*. IEEE, 2012, pp. 1–7. x, xii, 6, 60, 62, 63, 64, 70

[2] P. Milne, *Underwater acoustic positioning systems*. Gulf Publishing Company, 1983. 1, 10, 12

[3] A. Alcocer, P. Oliveira, and A. Pascoal, "Underwater acoustic positioning systems based on buoys with gps," in *Proceedings of the Eighth European Conference on Underwater Acoustics*, vol. 8, 2006, pp. 1–8. 2

[4] H. G. Thomas, "Gib buoys: an interface between space and depths of the oceans," in *Autonomous Underwater Vehicles, 1998. AUV'98. Proceedings Of The 1998 Workshop on*. IEEE, 1998, pp. 181–184. 2

[5] A. Alcocer, P. Oliveira, and A. Pascoal, "Study and implementation of an ekf gib-based underwater positioning system," *Control engineering practice*, vol. 15, no. 6, pp. 689–701, 2007. 2, 41

[6] S. Bancroft, "An algebraic solution of the gps equations," *Aerospace and Electronic Systems, IEEE Transactions on*, no. 1, pp. 56–59, 1985. 2, 32

[7] W. Nodland, T. Ewart, W. Bendiner, J. Miller, and E. Aagaard, "Spurv ii-an unmanned, free-swimming submersible developed for oceanographic research," in *OCEANS 81*. IEEE, 1981, pp. 92–98. 5, 12

[8] S. A. Jenkins, D. E. Humphreys, J. Sherman, J. Osse, C. Jones, N. Leonard, J. Graver, R. Bachmayer, T. Clem, P. Carroll *et al.*, "Underwater glider system study," Scripps Institution of Oceanography, Tech. Rep. 53, May 2003. 6, 63

[9] K. Vickery, "Acoustic positioning systems. a practical overview of current systems," in *Autonomous Underwater Vehicles, 1998. AUV'98. Proceedings of the 1998 Workshop on*. IEEE, 1998, pp. 5–17. 6, 11, 12, 16

[10] D. B. Kilfoyle and A. B. Baggeroer, "The state of the art in underwater acoustic telemetry," *Oceanic Engineering, IEEE Journal of*, vol. 25, no. 1, pp. 4–27, 2000. 6

[11] Q. Liang and X. Cheng, "Underwater acoustic sensor networks: Target size detection and performance analysis," *Ad Hoc Networks*, vol. 7, no. 4, pp. 803–808, 2009. 6

[12] J. C. Kinsey, R. M. Eustice, and L. L. Whitcomb, "A survey of underwater vehicle navigation: Recent advances and new challenges," in *IFAC Conference of Manoeuvering and Control of Marine Craft*, Lisbon, Portugal, September 2006. 6, 8

[13] V. Chandrasekhar, W. K. Seah, Y. S. Choo, and H. V. Ee, "Localization in underwater sensor networks: survey and challenges," in *Proceedings of the 1st ACM international workshop on Underwater networks*. ACM, 2006, pp. 33–40. 6

[14] D. R. Yoerger, M. Jakuba, A. M. Bradley, and B. Bingham, "Techniques for deep sea near bottom survey using an autonomous underwater vehicle," *The International Journal of Robotics Research*, vol. 26, no. 1, pp. 41–54, 2007. 6

[15] J. Partan, J. Kurose, and B. N. Levine, "A survey of practical issues in underwater networks," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 11, no. 4, pp. 23–33, 2007. 6

[16] L. Stutters, H. Liu, C. Tiltman, and D. J. Brown, "Navigation technologies for autonomous underwater vehicles," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 38, no. 4, pp. 581–589, 2008. 6, 18

[17] A. Y. Teymorian, W. Cheng, L. Ma, X. Cheng, X. Lu, and Z. Lu, "3d underwater sensor network localization," *Mobile Computing, IEEE Transactions on*, vol. 8, no. 12, pp. 1610–1621, 2009. 6

[18] H.-P. Tan, R. Diamant, W. K. Seah, and M. Waldmeyer, "A survey of techniques and challenges in underwater localization," *Ocean Engineering*, vol. 38, no. 14, pp. 1663–1676, 2011. 6, 12, 14

[19] C. E. G. Lapointe, "Virtual long baseline (vlbl) autonomous underwater vehicle navigation by using a single transponder," Master's thesis, Massachusetts Institute of Technology, June 2006. 6, 8, 11

[20] L. Paull, S. Saeedi, M. Seto, and H. Li, "Auv navigation and localization: A review," *Oceanic Engineering, IEEE Journal of*, vol. 39, no. 1, pp. 131–149, 2014. 7, 8, 9, 18

[21] A. J. Kapaldo, "Gyroscope calibration and dead reckoning for an autonomous underwater vehicle," Master's thesis, Virginia Tech, July 2005. 8

[22] W. Somers, "Doppler-based localization for mobile autonomous underwater vehicles," Ph.D. dissertation, Rutgers University, New Brunswick, 2011. 8

[23] M. Morgado, P. Oliveira, C. Silvestre, and J. F. Vasconcelos, "Usbl/ins tightly-coupled integration technique for underwater vehicles," in *Information Fusion, 2006 9th International Conference on*. IEEE, 2006, pp. 1–8. 9, 13, 14

[24] L. L. Whitcomb, D. R. Yoerger, H. Singh, and J. Howland, "Combined doppler/lbl based navigation of underwater vehicles," in *Proc. Int. Symp. on Unmanned Untethered Submersible Technology*, 1999. 9

[25] J. Vaganay, J. J. Leonard, and J. G. Bellingham, "Outlier rejection for autonomous acoustic navigation," in *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, vol. 3. IEEE, 1996, pp. 2174–2181. 12

[26] S. Smith and D. Kronen, "Experimental results of an inexpensive short baseline acoustic positioning system for auv navigation," in *OCEANS'97. MTS/IEEE Conference Proceedings*, vol. 1. IEEE, 1997, pp. 714–720. 13

[27] A. Urruela, J. Sala, and J. Riba, "Average performance analysis of circular and hyperbolic geolocation," *Vehicular Technology, IEEE Transactions on*, vol. 55, no. 1, pp. 52–66, 2006. 19

[28] M. Deffenbaugh, J. G. Bellingham, and H. Schmidt, "The relationship between spherical and hyperbolic positioning," in *OCEANS'96. MTS/IEEE. Prospects for the 21st Century. Conference Proceedings*, vol. 2. IEEE, 1996, pp. 590–595. 19

[29] L. Techy, K. A. Morgansen, and C. A. Woolsey, "Long-baseline acoustic localization of the seaglider underwater glider," in *American Control Conference (ACC), 2011*. IEEE, 2011, pp. 3990–3995. 21, 41, 58

[30] J. M. Mendel, *Lessons in digital estimation theory*. Prentice-Hall, Inc., 1986. 22, 23, 24, 45

[31] B. S. Bingham, "Precision autonomous underwater navigation," Ph.D. dissertation, Massachusetts Institute of Technology, 2003. 23, 41

[32] J. L. Crassidis and J. L. Junkins, *Optimal estimation of dynamic systems*, ser. Chapman & Hall/CRC Applied Mathematics & Nonlinear Science. CRC press, 2011. 31, 41, 44, 48, 50, 51

[33] L. Techy, K. A. Morgansen, and C. A. Woolsey, "Long-baseline ranging system for acoustic underwater localization of the seaglider underwater glider," University of Washington, Aeronautics & Astronautics, Seattle, WA, Tech. Rep. UWAATR-2010-0001, 2010. 32

[34] N. Mahmoudian, J. Geisbert, and C. Woolsey, "Dynamics and control of underwater gliders i: Steady motions," Virginia Center for Autonomous Systems, Virginia Tech, VA, Tech. Rep. VACAS-2007-01, June 2007. 39, 60

[35] N. Mahmoudian and C. Woolsey, "Dynamics and control of underwater gliders ii: Motion planning and control," Virginia Center for Autonomous Systems, Virginia Tech, VA, Tech. Rep. VACAS-2010-2, November 2010. 39, 60

[36] P. Baccou and B. Jouvencel, "Homing and navigation using one transponder for auv, postprocessing comparisons results with long base-line navigation," in *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, vol. 4. IEEE, 2002, pp. 4004–4009. 41

[37] A. S. Gadre and D. J. Stilwell, "A complete solution to underwater navigation in the presence of unknown currents based on range measurements from a single location," in *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on.* IEEE, 2005, pp. 1420–1425. 41

[38] P.-M. Lee, B.-H. Jun, K. Kim, J. Lee, T. Aoki, and T. Hyakudome, "Simulation of an inertial acoustic navigation system with range aiding for an autonomous underwater vehicle," *Oceanic Engineering, IEEE Journal of*, vol. 32, no. 2, pp. 327–345, 2007. 41

[39] R. Carona, A. P. Aguiar, and J. Gaspar, "Control of unicycle type robots tracking, path following and point stabilization," in *International Proceeding of IV Electronics and Telecommunications*, Lisbon, 2008, pp. 180–185. 41

[40] S. Singh, M. Grund, B. Bingham, R. Eustice, H. Singh, and L. Freitag, "Underwater acoustic navigation with the whoi micro-modem," in *OCEANS 2006*. IEEE, 2006, pp. 1–4. 55, 76, 77

[41] A. Wolek and C. Woolsey, "Feasible dubins paths in presence of unknown, unsteady velocity disturbances," *Journal of Guidance, Control, and Dynamics*, pp. 1–5, 2014. 60

[42] W.-B. Yang and T. Yang, "High-frequency fh-fsk underwater acoustic communications: The environmental effect and signal processing," *High Frequency Ocean Acoustics*, vol. 728, pp. 106–113, 2004. 76

[43] L. Freitag, M. Grund, S. Singh, J. Partan, P. Koski, and K. Ball, "The whoi micro-modem: an acoustic communications and navigation system for multiple platforms," in *OCEANS, 2005. Proceedings of MTS/IEEE*. IEEE, 2005, pp. 1086–1092. 76, 77, 78

[44] *Micro-Modem Software Interface Guide*, 3rd ed., Woods Hole Oceanographic Institute, July 2013. 76, 79, 81, 83, 88

[45] M. Grund, J. Partan, P. Koski, and L. Freitag, "Synchronous navigation with the micro-modem," WHOI, Tech. Rep. Revision: D, 2005. 80, 81

[46] T. Schneider and H. Schmidt, "Goby-acomms version 2: extensible marshalling, queuing, and link layer interfacing for acoustic telemetry," in *9th IFAC Conference on Manoeuvring and Control of Marine Craft, Arenzano, Italy*, 2012. 85, 86

[47] *Goby Underwater Autonomy Project*, User manual for v2.1.0 alpha1 ed., April 2014. 85

[48] T. Schneider and H. Schmidt, "The dynamic compact control language: A compact marshalling scheme for acoustic communications," in *Oceans 2010 IEEE-Sydney*. IEEE, 2010, pp. 1–10. 85