INTER-GLIDER UNDERWATER COMMUNICATION AND COORDINATION FOR OCEAN MONITORING AND COASTAL TACTICAL SURVEILLANCE

BY BAOZHI CHEN

A dissertation submitted to the

Graduate School—New Brunswick

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

Graduate Program in Electrical and Computer Engineering

Written under the direction of

Professor Dario Pompili

and approved by

____

____

____

____

New Brunswick, New Jersey

October, 2012

ABSTRACT OF THE DISSERTATION

Inter-glider Underwater Communication and Coordination for Ocean Monitoring and Coastal Tactical Surveillance

by Baozhi Chen

Dissertation Director: Professor Dario Pompili

In order to achieve efficient and cost-effective sensing of the vast under-sampled 3D aquatic volume, intelligent adaptive sampling strategies involving teams of Autonomous Underwater Vehicles (AUVs) endowed with underwater wireless communication capabilities become essential. These autonomous vehicles should coordinate and steer through the region of interest, and cooperatively sense and transmit multimedia data to onshore stations for real-time data processing and analysis. Because of the propagation limitations of Radio Frequency (RF) and optical waves, the typical wireless physical-layer communication technology in underwater networks, for distances above a hundred of meters, relies on acoustic waves.

Due to the stringent constraints of the underwater acoustic channels, as of today existing works on underwater acoustic communications are mostly focused on enabling delay-tolerant low-bandwidth applications tailored for measuring only scalar physical phenomena. Hence, it is necessary to design solutions for reliable, high data-rate multimedia underwater acoustic communications and to seamlessly integrate the control and communication of AUVs. In this dissertation, I propose solutions to improve the performance of inter-vehicle acoustic communication and coordination among AUVs. In particular, these solutions are based on underwater gliders and can be extended to other classes of AUVs following predictable trajectories.

Due to the inaccessibility of Global Positioning System (GPS) signal underwater, location estimates of a node may be inaccurate. Inaccuracies in models for deriving position estimates, self-localization errors, and drifting due to ocean currents, however, cause uncertainty when estimating an AUV's position. In this dissertation, I first propose a statistical model to estimate an AUV's position and its associated position uncertainty. Then, the position uncertainty under the influence of ocean currents is further predicted using the Unscented Kalman Filter. Based on this model, in order to optimize the inter-vehicle communications, I propose a delay-tolerant networking solution exploiting the predictability of AUV trajectories and the directional radiation pattern of transducers, a reliable geocasting solution for AUVs with high position uncertainty, and an under-ice localization solution that can minimize localization uncertainty and communication overhead.

Based on these underwater communication techniques, I also propose efficient team-formation and -steering algorithms for underwater gliders in order to take measurements in space and time from the under-sampled vast ocean. Team formation and steering algorithms relying on underwater acoustic communications are proposed to enable glider swarming that is robust against ocean currents and acoustic channel impairments. These algorithms use real underwater acoustic modems and are combined with realistic underwater communication models. Additionally, novel bio-inspired underwater acoustic communication techniques are also proposed to improve the coordination performance.

I also designed and implemented an underwater network emulator using WHOI Micro-Modems, and the performance of the proposed solutions is evaluated using this emulator as well as software simulations. Communication protocols were also implemented on acoustic modems and tested in ocean experiments.

Acknowledgements

The work in this dissertation would have been impossible without the kind, patient and generous help from many people with whom I have been lucky to work during the years of my doctoral study.

I feel deeply indebted to my Ph.D. advisor, Dr. Dario Pompili, an exceptional researcher and mentor who spent countless hours in advising my research. His encouragement and direction helped me set and achieve goals beyond what I imagined. He has provided me a unique environment to learn new knowledge and skills, broaden my vision, and pursue my research. Dr. Pompili's ability to delve into a broad range of research areas that range from underwater robotics to autonomic computing and his success in building collaborations with researchers from different departments speak not only of his insights in different research areas, but also of his ability to perform outstanding research. He has not only given numerous comments to help me improve my research work, but also given me a lot of advice on teaching and mentoring undergraduate and graduate students. Dr. Pompili's attention to details and dedication to exceptional scientific research have greatly improved the quality of my manuscripts. Few researchers could ever direct such a diverse research program and also deliver productive research results. His attitude to original and high quality research work will always be my role model of my future work.

I also thank Dr. Manish Parashar, Dr. Ivan Marsic and Dr. Ulrich Kremer for accepting to be the committee members in my dissertation defense. They have given me a lot of help and advice during collaboration. They also provided me with a lot of insightful comments to improve my Ph.D. dissertation. Thanks should also go to Dr. Zoran Gajic for the guidance on graduate study as well as to ECE staff including Noraida Martinez, Lynn Ruggiero, and Barbara Klimkiewicz for their help on many other issues.

As a member of Cyber-Physical Systems (CPS) lab, I am also grateful to get a lot of help from other members and alumni of the CPS lab. I would like to thank current members Hariharasudhan Viswanathan, Eun Kyung Lee, Parul Pandey, Cagdas Karatas, Mike Puntolillo, Jennifer Padgett, Kalyan Yalamanchi, Chihwei Ching, and Seetha Annamraju. I would also like to thank the lab alumni William Somers, John Paul Varkey, and Patrick C. Hickey. They have helped me a lot in improving and implementing my proposed solutions, evaluating the performance of these solutions, and proofreading my manuscripts. Without their collaboration and support, my Ph.D. study would have been much more difficult. I would also like to thank David Aragon, Hans Christian Woithe, and Qiaojun Wang for their help during collaboration.

I would also like to thank many of my friends who helped me a lot during my study in the United States and it is simply not possible to name them all. Special thanks should go to Grant P. Shieh, Yihsiang Kevin Chen, Rui Zeng, Tim Wu, Wilfred Cheung, Dong Guo, Juhua Li, and Jia Zeng, who gave me a lot of support during my difficult times.

Finally, I would like to thank my family for their understanding and support of my graduate study. Without their support, I would not have been able to go so far. Specifically, I would like to thank my wife for her sacrifice for the family during my study. I would also like to thank my parents and sisters for their continuous support to me without expectation in return.

Dedication

To my family and my friends

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Recently UnderWater Acoustic Sensor Networks (UW-ASNs) [2], which consist of underwater sensors and vehicles, have been deployed to carry out collaborative monitoring tasks such as oceanographic data collection, disaster prevention, and navigation. To ensure sufficient coverage of the vast undersampled 3D aquatic volume, Autonomous Underwater Vehicles (AUVs) with underwater sensing and wireless communication capabilities become essential. These AUVs, which are divided into two main classes – propeller-less/buoyancy-driven (e.g., gliders) and Propeller-Driven Vehicles (PDVs) – rely on local intelligence with minimal onshore operator dependence. These AUVs should coordinate and steer through the region of interest, and cooperatively sense and transmit multimedia data to onshore stations for real-time data processing and analysis.

Because of the propagation limitations of Radio Frequency (RF) and optical waves, i.e., high medium absorption and scattering, respectively, the typical wireless physical-layer communication technology in underwater networks, for distances above a hundred of meters, relies on acoustic waves. Yet, several unique characteristics of the underwater acoustic medium such as frequency-dependent transmission loss, colored ambient noise, multipath, Doppler frequency spread, very high propagation delay, and variable and distance-dependent channel bandwidth make reliable, high-data rate multimedia communications an extremely challenging research problem. For these reasons, as of today existing works on acoustic communications are mostly focused on enabling delay-tolerant low-bandwidth applications tailored for measuring only scalar physical phenomena.

Besides the need for control and coordination, position information is paramount in mobile underwater sensor networks as the data collected has to be associated with the 3D location in order to be spatially reconstructed onshore. In terrestrial sensor networks, the position of a node can generally be characterized by a single point because localization error can be very small by using Global Positioning System (GPS), which however does not work underwater. In contrast, inaccuracies in models for position estimation, self-localization errors, and drifting due to ocean currents will significantly increase the uncertainty in position of an underwater vehicle. Hence, using a deterministic point is not enough to characterize the position of an AUV. Furthermore, such a deterministic approach underwater may lead to problems such as errors in inter-vehicle communications using geographical routing, vehicle collisions, loss of synchronization, and mission failures [3].

In order to address the problems due to position uncertainty, in this work I introduce a probability region to characterize stochastically a node's position. Depending on the view of the different nodes, two forms of position uncertainty are needed, i.e., *internal uncertainty*, the position uncertainty associated with a particular entity/node (such as an AUV) *as seen by itself*, and *external uncertainty*, the position uncertainty associated with a particular entity/node *as seen by others*. These two notions introduce a shift in AUV localization: from a deterministic to a probabilistic view. This shift can then be leveraged to improve the performance of solutions for problems in a variety of research areas. For example, in *UW-ASNs*, using the external-uncertainty notion, geographical routing errors can be decreased and a node can estimate better the transmission power to guarantee the Signal-to-Noise Ratio (SNR) for correct data reception by taking into account not only channel impairments but also position uncertainty. The notion of external uncertainty can also be used in *underwater robotics* to minimize the risk of vehicle collisions, in *underwater localization* to decrease the position uncertainty by selecting a proper subset of nodes to be used as "anchors" (i.e., reference nodes to estimate position). Last, but not least, in *task allocation*, i.e., the problem of selecting a subset of AUVs to accomplish reliably the mission under specific requirements, this notion can be used to improve the performance of multicasting the mission details to AUVs within a certain region. Finally, it can be used to improve the quality of 3D data reconstruction as the vehicle deviation from the original mission path can be estimated and factored in during *data processing/visualization*.

To enable these applications, nodes need to be able to estimate the external uncertainty of other nodes. To do this, they need to first estimate their internal uncertainty and then broadcast it to other neighbors. Due to the communication delays (including propagation delay and transmission delay) and information loss, this received uncertainty information is a delayed version of a node's internal uncertainty and is used as the base for the neighbors to estimate the sender's uncertainty (i.e., external uncertainty). As a result, these two forms of uncertainty are generally different. I first propose a statistical model to estimate an AUV's position and its associated internal uncertainty. Then based on this internal uncertainty estimate, external uncertainty is predicted using Unscented Kalman Filter where the influence of ocean currents is taken into account. Based on this model, to improve the performance of the inter-vehicle communications, I propose: 1) a delay-tolerant networking solution exploiting the predicable AUV trajectories and directional radiation pattern of transducers, 2) a reliable geocasting solution (i.e., the transmission of data packets to nodes located in a certain geographic region) for AUVs with high position uncertainty, and 3) an under-ice localization solution (where position error becomes more severe due to the increased difficulty in accessing GPS as well as that in deploying localization infrastructure) that can minimize localization uncertainty and communication overhead. Respectively, *these solutions can satisfy the following need to facilitate the coordination of multiple AUVs*: 1) the need to optimize acoustic communications among AUVs, 2) the need to send query or commands to these AUVs, e.g., multicasting to AUVs located in a specified region; and 3) the need to minimize localization uncertainty, e.g., for the under-ice environment where deployment of localization infrastructure is difficult.

Using these underwater communication techniques, I also propose efficient team formation and steering algorithms for underwater gliders. In order to take measurements in space and time from the under-sampled vast ocean, these vehicles need to form a team in a specific formation (team formation), steer through the 3D region of interest (team steering), and take application-dependent measurements such as temperature and salinity. I propose team formation and steering algorithms relying on underwater acoustic communications in order to enable glider swarming that is robust against ocean currents and acoustic channel impairments (e.g., high propagation and transmission delay, and low communication reliability). The proposed team formation and team steering algorithms use real underwater acoustic modems and are combined with realistic underwater communication models. Additionally, I designed novel bio-inspired underwater acoustic communication techniques to improve the coordination performance. This reduces the need for localization packets and thus saves communication overhead.

In the rest of this chapter, I first introduce the basics on underwater acoustic communications and AUVs, and then I outline the structure of the dissertation, which includes the summary of existing works and our contribution, introduction of two novel position uncertainty concepts, an overview of the proposed inter-vehicle communication solutions, and team formation and steering algorithms.

1.1 Underwater Acoustic Communications

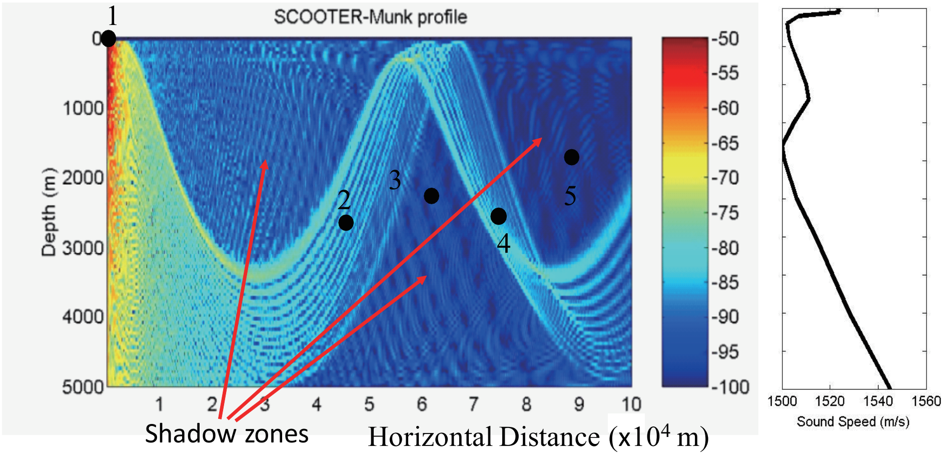


Figure 1.1: Shadow zone scenario.

Communications between nodes in UW-ASNs rely on underwater acoustic communication technology. Models have been proposed for the propagation of acoustic waves underwater. The Urick model is a coarse approximation for underwater acoustic wave propagation, whose transmission loss $TL(l, f)$ [dB] can be model as,

$$TL(l, f) = \kappa \cdot 10 \log_{10}(l) + \alpha(f) \cdot l, \tag{1.1}$$

where l is the distance between the transmitter and receiver and f is the carrier frequency. Spreading factor κ is taken to be 1.5 for practical spreading, and $\alpha(f)$ $[\text{dB/m}]$ represents an absorption coefficient that increases with f [4]. In (1.1), the first term accounts for *geometric spreading*[1], which is the spreading of sound energy caused by the expansion of the wavefronts. It increases with the propagation distance and is independent of frequency.

In reality, sound propagation speed varies with water temperature, salinity, and pressure, which causes wave paths to bend. Acoustic waves are also reflected from the surface and bottom. Such uneven propagation of waves results in convergence (or shadow) zones which may receive much less (or more) transmission loss than that predicted by the Urick model. Details regarding to the

[1] There are two kinds of geometric spreading: *spherical* (omni-directional point source, spreading coefficient $\kappa = 2$), and *cylindrical* (horizontal radiation only, spreading coefficient $\kappa = 1$). In-between cases show a spreading coefficient χ in the interval $(1, 2)$, depending on water depth and link length.

explanation of these phenomena can be found in [5].

Due to these phenomena, the Urick model is not sufficient to describe the underwater channel for simulation purposes. The Bellhop model is based on ray/beam tracing, which can model these phenomena more accurately. This model can estimate the transmission loss by two-dimensional acoustic ray tracing for a given sound speed depth profile or a given sound speed field, in ocean waveguides with flat or variable absorbing boundaries. Transmission loss is calculated by solving differential ray equations, and a numerical solution is provided by HLS Research [6]. Because the Bellhop model requires more information about the environment than an underwater vehicle will have, it is only used to simulate the acoustic environment for testing.

An example plotted using the Bellhop model is shown in Fig. 1.1, where the left figure is the transmission loss of node 1 located at the origin and the right figure is the sound speed profile used to derive the transmission loss (the y-axis is the depth with the same range as the left figure). Suppose that node 1 at the origin wants to send a packet to node 5 in this example. This transmission will fail since node 5 is in the shadow zone and therefore cannot receive acoustic waves.

We adopt the empirical ambient noise model presented in [4], where a ‘V’ structure of the power spectrum density (psd) is shown. The ambient noise power is obtained by integrating the empirical psd over the frequency band in use. Note that in underwater acoustics, power (or source level) is usually expressed using decibel (dB) scale, relative to the reference pressure level in underwater acoustics $1\mu Pa$, i.e., the power (or source intensity) induced by $1\mu Pa$ pressure. The conversion expression for the source level SL re μPa at the distance of $1\ \mathrm{m}$ of a compact source of P watts is shown in [5] as $SL = 170.77 + 10\log_{10} P$. Note that here P is the transmit or receive acoustic power at the transducer. The conversion between the sound source level and electrical energy depends on the characteristics of the underwater transducer. At a frequency, the ability to convert electrical voltage or current into sound pressure level depends on the so-called *Transmit Voltage Response (TVR)* or the *Transmit Current Response (TCR)* - defined as the ratio of sound pressure level measured at 1 meter and the *voltage* or *current* applied to the transducer, while the ability to convert the sound pressure level into electrical voltage depends on the receive sensitivity of the hydrophone (or transducer), i.e., Free Field Voltage Sensitivity (FFVS) - the ratio of output open circuit voltage and the free-field sound pressure level at the hydrophone (or transducer), alternatively called as the Open Circuit Voltage Sensitivity (OCVS).

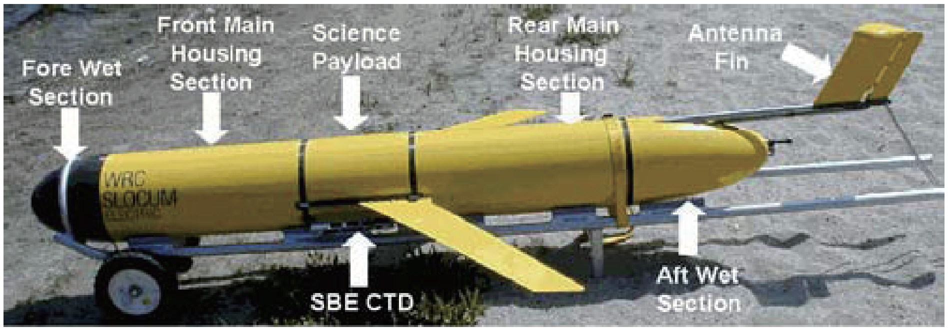


Figure 1.2: A SLOCUM glider (Courtesy of WHOI).

1.2 Autonomous Underwater Vehicles

Autonomous Underwater Vehicles (AUVs) are programmable robots that, depending on their design, can drift, drive, or glide through the water without the real-time control by human operators. Generally, AUVs are equipped with onboard computers and sensors, which allow them to sense the underwater environment and collect the underwater data as they move through the water, and make decisions on their own. For example, they can change their navigation trajectories or mission profiles based on the collected environmental data. To enable some level of manual control, they are generally equipped with satellite modems or underwater acoustic modems so that they can communicate with operators on shore or on boad periodically or continuously. For localization, they are generally equipped with GPS devices so that they can get the global location when surfaced, while they rely on underwater positioning algorithms or systems such as dead reckoning and long baseline navigation systems when submerged.

AUVs allow oceanographic researchers to study underwater environments that are typically impossible or difficult to reach. Moreover, they can stay underwater and continuously collect environmental data for a long period of time (up to months). Therefore they are widely believed to be a revolutionary oceanographic technology. Ever since their appearance, hundreds of AUVs have been designed. These vehicles range in size from portable lightweight AUVs to vehicles of over 10 meters in length. Small vehicles generally require low logistics while large vehicles have large sensor payload capacity and long operation period.

Most AUVs today are powered by rechargeable batteries (e.g., lithium ion, lithium polymer, nickel metal hydride), which enable propellers or thrusters to move them through the water, and power up onboard sensors. To extend the battery life, energy harvesting solutions have been proposed to use renewable energy sources such as solar, thermal and kinetic (e.g., wave) energy [7, 8]. Depending on the power source and the mission, AUVs can operate over hours, days or even months before recharging the battery.

Typically, AUVs can move at speeds ranging from 0.2 to 4.0 $\mathrm{m/s}$. To enable high accuracy navigation, they rely on underwater acoustic positioning systems such as long baseline, short baseline systems. The Long Baseline (LBL) system require transponders deployed at fixed positions on the sea floor. When a surface reference such as a ship is available, the Ultra-Short Baseline (USBL) or Short Baseline (SBL) positioning system is used to estimate the AUV position using acoustic range measurements. When operating completely autonomously, an AUV will surface and take its own GPS fix. Between GPS fixes, an Inertial Navigation System (INS) onboard - which is generally expensive equipment - can be used to measure the acceleration while the vehicle speed can be measured using Doppler velocity technology, and the depth (vertical position) is measured using a pressure sensor. To reduce the cost incurred by the INS system, an alternative is to use the Dead Reckoning (DR) method between GPS fixes, as adopted by the SLOCUM underwater glider today.

AUVs are typically equipped with sensors to measure the chemical and physical properties of the underwater environment. These include the Conductivity, Temperature, Depth (CTD) sensors, Acoustic Doppler Current Profiler (ADCP) (used to measure the ocean current speed), magnetometers, side-scan and other sonars, image and video cameras, pH and dissolved oxygen sensor, radiometer, and spectrophotometer for harmful algal blooms. To support the navigation and localization of the vehicles, GPS, magnetic compass, tilt sensor, and altimeter are also installed.

Acoustic, radio frequency, and satellite communication modems installed on the AUV offer the capability to monitor and control the vehicle remotely (e.g., onshore or on boat). Moreover, these communication technologies make it possible to acquire and process the high-resolution, high quality, and heterogeneous sensor data in near real time for advanced ocean exploration. Recent advance in electronics, computer and communication technologies has make it possible to develop more and more capable and affordable AUVs to explore the ocean. Consequently these vehicles are expected to be widely used for advanced ocean monitoring and coastal tactical surveillance in the near future.

AUVs can be divided into two main classes: Propeller Driven Vehicles (PDVs) and buoyancy-driven gliders. PDVs are one class of AUVs that are driven by propeller(s). As the propulsion system consumes a lot of energy, the lifespan and operation coverage are limited due to the limited battery capacity. In general, PDVs have a life span ranging from several hours to a few days and can traverse distances of several hundred kilometers. The main benefit of using PDVs is that they can cover long distance in a short time (due to the high speed brought by the propulsion system). In addition, they generally have high maneuverability as the onboard computer can change the course or turn in time.

On the other hand, gliders are another class of AUVs that are not driven by propeller. For propulsion, they change their buoyancy in a small amount using a pump and rely on wings to convert vertical velocity into horizontal motion as they rise and fall through the ocean, forming a sawtooth trajectory. As a result, in contrast to PDVs, gliders spend very small amount of energy in propulsion and hence they can operate over long periods of time (weeks or months), even though they are not as fast as PDVs. They travel at a fairly constant horizontal speed, typically 0.25 m/s [2]. The impressive operation time and energy efficiency of these gliders make them a popular choice for oceanographic researchers. Quite a few gliders have been produced and then used for scientific research. These include the SLOCUM glider (which uses electric buoyancy engine) and SLOCUM Thermal glider (propelled by extracting heat from the oceans thermal stratification) by Webb Research, the Seaglider (whose drag is higher at low speeds and lower at high speed than SLOCUM) by University of Washington, and the Spray glider (designed for efficient deep-water performance) by Scripps Institution of Oceanography. The pitch angle of the vehicle is controllable by movable internal ballast (usually battery packs), and steering is accomplished either with a rudder (as in SLOCUM) or by moving internal ballast to control roll (as in Spray and Seaglider). Buoyancy is adjusted either by using a piston to flood/evacuate a compartment with seawater (SLOCUM) or by moving oil in/out of an external bladder (Seaglider, Spray, and SLOCUM Thermal). Gliders control their heading toward their predefined waypoints using a magnetic compass and may surface periodically to acquire their location using GPS and communicate with the operators using a radio frequency or satellite connection.

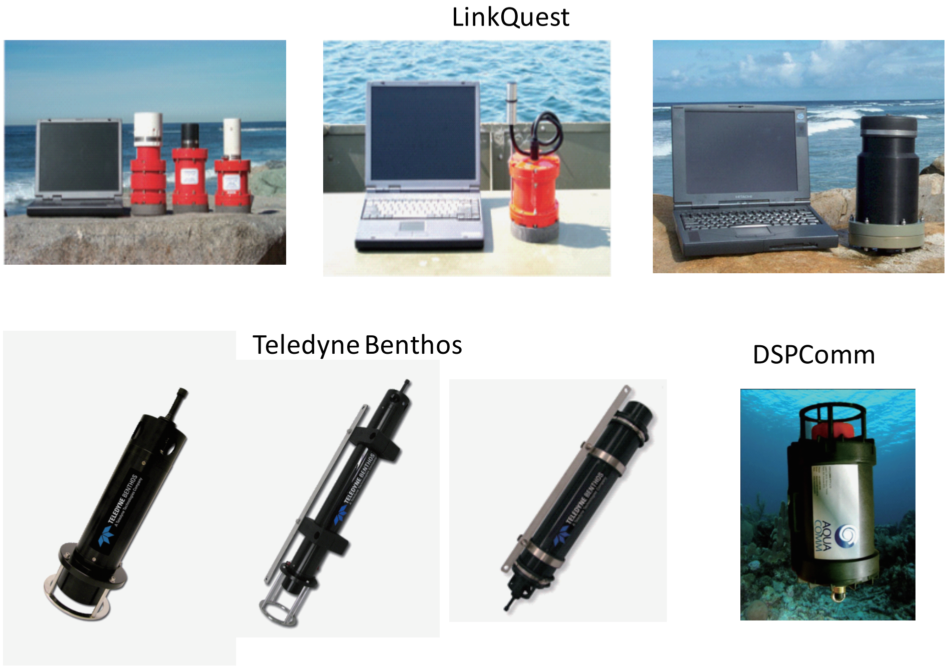


Figure 1.3: Underwater acoustic modems by different manufacturers.

1.3 Underwater Acoustic Modems

As of today, many acoustic modems - such as those designed and manufactured by companies like LinkQuest, Teledyne Benthos, DSPComm- are commercially available to provide communication capabilities in different underwater environments. These modems uses communication techniques such as Frequency-Shift Keying (FSK), Phase-Shift Keying (PSK), Direct Sequence Spread Spectrum (DSSS) and Orthogonal Frequency-Division Multiplexing (OFDM), offering data rates up to 38.4 kbps over different communication ranges, i.e., short range (up to about 500 m), medium range (up to about 4000 m), and long range (up to about 10000 m) in different underwater environments (shallow water or deep water) for different communication link setups (vertical or horizontal communication link). Figure 1.3 shows some modems produced by these manufacturers.

These modems have been used in different underwater communication networks. However, they are generally big in size, which is not suitable for underwater vehicles such as the SLOCUM glider. Due to the size constraint, the popular choice for underwater gliders today is the Micro-Modem produced by Woods Hole Oceanography Institution (WHOI), as shown in Fig. 1.4. The WHOI Micro-Modem is currently the state-of-the-art modem used on the SLOCUM glider. It is compact in size (including the transducer), offering data rates from 80 to 5300 bps with communication

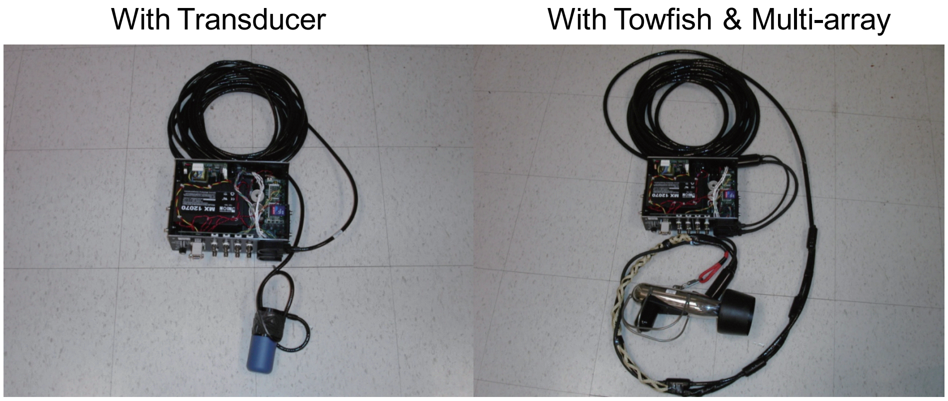


Figure 1.4: WHOI Micro-Modem connected to different transducers.

range of up to a few kilometers. Such feature makes it an appropriate choice for AUVs like underwater gliders. Our proposed solutions are mainly based on the WHOI Micro-Modem yet they are extensible to other acoustic modems.

1.4 Overview of the Dissertation

Existing research works on UW-ASNs, focus either on problems from the perspective of communications and networks [2], or on problems from the robotics perspective. On one hand, researchers aim at improving the performance of underwater communications and networks without considering the impact on the control of underwater vehicles. On the other hand, robotic researchers aim at improving the control of underwater vehicles without considering the physical constraints of underwater communications. Few researchers worked on problems that jointly consider the communication and coordination among a team of underwater vehicles. Existing works either rely on using the satellite communications to control a team of vehicles where real-time control is impossible for submerged vehicles [9], or on controlling the team assuming ideal acoustic communications (i.e., relying on ideal graph theory models). Moreover, these works generally assume ideal underwater localization, which may not be true when localization infrastructure is unavailable and the vehicles drift due to the influence of ocean currents. To address these problems, in this dissertation, we first propose a solution to model and predict position uncertainty for underwater vehicles. Then based on this position uncertainty model, novel solutions are proposed to improve the performance of communication and coordination among mobile AUVs. The contribution of the proposed solutions, in summary, is that they are designed to improve the communication and coordination performance for AUVs with high position uncertainty (e.g., due to influence of ocean currents).

In the rest of this section, an overview of the proposed solutions for inter-vehicle communications and coordinations is given. We first distinguish position uncertainty into two classes, and propose a statistical model to estimate and predict the position uncertainty of AUVs, which can be applied to underwater gliders and AUVs whose trajectories are predictable. Then we propose inter-vehicle communication solutions for AUVs, which include an optimization framework for delay-tolerant traffic, a reliable geocasting solution (i.e., the transmission of data packets to nodes located in a certain geographic region) for AUVs with high position uncertainty, and an under-ice localization solution (where position error becomes more severe due to the increased difficulty in accessing GPS as well as that in deploying localization infrastructure) that can minimize localization uncertainty and communication overhead. These communication solutions are then applied to facilitate the coordination among AUVs. Specifically, *the localization solution can be used to improve the localization performance of AUVs where localization infrastructure is not available; the geocasting solution can be used to send queries or commands to AUVs within a specified 3D region; while the optimization communication framework for delay-tolerant traffic can be used for energy-efficient communications among AUVs.*

Based on these underwater communication techniques, we also propose efficient team formation and steering algorithms for underwater gliders. In order to take measurements in space and time from the under-sampled vast ocean, these vehicles need to form a team in a specific formation (*team formation*), steer through the 3D region of interest (*team steering*), and take application-dependent measurements such as temperature and salinity. We propose team formation and steering algorithms relying on underwater acoustic communications in order to enable glider swarming that is robust against ocean currents and acoustic channel impairments (e.g., high propagation and transmission delay, and low communication reliability). The proposed team formation and team steering algorithms use real underwater acoustic modems and are combined with realistic underwater communication models. Additionally, we design novel bio-inspired underwater acoustic communication techniques to improve the coordination performance. This reduces the need for localization packets and thus saves communication overhead.

We summarize our proposed solutions and their contributions as the following.

1) Statistical Modeling and Prediction of the AUV Position: To address the problems due to position uncertainty, in this work we introduce a probability region to characterize stochastically a node's position. Depending on the view of the different nodes, two forms of position uncertainty are needed, i.e., *internal* and *external uncertainty*, the position uncertainty associated with a particular entity/node (such as an AUV) *as seen by itself*, and *as seen by others*, respectively. To estimate the external uncertainty, in this work we first propose a statistical approach to model the internal uncertainty of AUVs whose trajectories are predictable. Based on the estimated internal uncertainty, we then propose a solution using the Unscented Kalman Filter (UKF) to predict the external uncertainty. Note that our solution is not a localization algorithm. Rather, we provide a way to estimate the uncertainty that is associated with one localization technique and leverage this information for performance improvement.

Specifically, the contributions of this work include: 1) the distinction between two forms of position uncertainty (internal and external, depending on the view of the different nodes); 2) a statistical algorithm to characterize the internal uncertainty of a general AUV (including underwater gliders) whose trajectory is predictable; 3) a solution to predict the external uncertainty based on UKF; 4) an algorithm to dynamically adjust update interval in order to minimize overhead for position information; and 5) guidelines on how to apply the estimated external-uncertainty regions in communication and distributed robotic solutions for performance improvement.

2) An inter-vehicle communication optimization framework for delay-tolerant traffic: Based on the estimated external uncertainty, we propose QUO VADIS[2], a **Q**oS-aware **u**nderwater **o**ptimization framework for inter-**v**ehicle communication using **a**coustic **di**rectional transducer**s**. QUO VADIS is a cross-layer optimization framework for delay-tolerant UW-ASNs that jointly considers the end-to-end (e2e) delay requirements and constraints of underwater acoustic communication modems, including transducer directivity, power control, packet length, modulation, and coding schemes. Specifically, the proposed framework uses the external-uncertainty region estimates of the gliders and forwards delay-tolerant traffic with large maximum e2e delay, which includes *Class I (delay-tolerant, loss-tolerant)* traffic and *Class II (delay-tolerant, loss-sensitive)* traffic [10]. Moreover, our cross-layer communication framework exploits the frequency-dependent radiation pattern of underwater acoustic transducers. By decreasing the frequency band, transducers can change their "directivity" turning from being almost omnidirectional (with a gain of ≈ 0 dBi) – which is a desirable

[2] "Quo vadis?" is a Latin phrase meaning "Where are you going?".

feature to support neighbor discovery and multicasting, geocasting, anycasting, and broadcasting) – to directional (with gains up to 10 dBi) – which is useful for long-haul unicast transmissions.

The contributions of this work are as follows: 1) We propose a distributed communication framework for delay-tolerant applications where AUVs can conserve energy by waiting for a ‘good’ network topology configuration, e.g., a *favorable alignment*, before starting to communicate. 2) We exploit the frequency-dependent directivity of the acoustic transducer that is originally used as omnidirectional transducer at one frequency to optimize network performance. 3) We propose a cross-layer optimization framework that jointly considers the constraints of underwater acoustic modems and the trajectory of the AUV for energy minimization.

3) A reliable geocasting solution for UW-ASNs: In this work, based on different degrees of neighbor information, we propose two versions of an underwater geocasting solution whose objective is to reach the highest number of nodes within a pre-defined directional 3D region in a given amount of time *when the positions of the nodes are uncertain*. Based on the external uncertainty estimates, packets are forwarded along the path that can reach the nodes in the region along the specified direction in minimal time while maximizing link reliability. Moreover, packet transmissions are scheduled in an optimal manner in order to avoid collisions and save the number of transmissions.

To the best of our knowledge, our approach is the first geocasting solution for UW-ASNs that accounts for position uncertainty. Specifically, our contribution includes the design and implementation of: (1) prioritization and scheduling mechanisms to maximize the link reliability while minimizing the time for geocasting; (2) a mechanism to save the number of transmissions by partitioning neighbor nodes into two sets – *forwarding nodes* and *non-forwarding nodes* – so that retransmissions can be minimized; and (3) a distributed solution that can be used for existing underwater acoustic modems using random-access MAC protocols (e.g., Benthos, WHOI). Note that our solution relies only on the use of timers (without requiring synchronization among nodes), and only local neighbor information is used for packet scheduling and forwarding.

4) An under-ice/underwater localization solution for minimization of position uncertainty: Existing localization schemes underwater generally rely on the deployment of transponders or nodes with underwater communication capabilities as reference points, which requires either much deployment effort or much communication overhead. Moreover, these schemes are not able to estimate the uncertainty associated with the calculated position, which is high in under-ice environments, and thus are not able to minimize position uncertainty.

To address this problem, we propose a solution that uses only a subset of AUVs without relying on localization infrastructure. Specifically, a position uncertainty model in [11] is introduced to characterize an AUV's position. This model is extended to estimate the uncertainty associated with the standard distance-based localization technique, resulting in the distance-based localization with uncertainty estimate (DISLU). We further propose a Doppler-based technique with uncertainty estimation capability, which is called Doppler-based localization with uncertainty estimate (DOPLU). DISLU relies on packets (i.e., communication overhead) to measure the inter-vehicle distances (i.e., ranging), which, in conjunction with positions of reference nodes (in general *other AUVs*), are utilized to estimate the position. On the other hand, DOPLU, which measures Doppler shifts from ongoing communications and then uses these measurements to calculate velocities for localization, removes the need for ranging packets. As DOPLU only relies on relative measurements, it may not be able to fix displacement errors introduced by the rotation or position translation of the AUV group. In this case, DISLU is executed to bound such localization errors. Considering these tradeoffs, using the uncertainty model, the localization error and communication overhead of DISLU and DOPLU can be jointly considered and algorithms are devised to minimize the localization uncertainty and communication overhead while satisfying localization error requirement.

Our solution offers a way to estimate the degree of uncertainty associated with a localization technique and based on this estimation it further minimizes both position uncertainty and communication overhead. The contributions of this work include: 1) a probability model to estimate the position uncertainty associated with localization techniques; 2) an algorithm to minimize localization uncertainty by selecting an appropriate subset of reference nodes; 3) an algorithm to optimize the localization interval in order to further minimize the localization overhead; and 4) a Doppler-based localization technique that can exploit ongoing communications for localization.

5) A team formation and steering solution for AUVs: existing solutions such as [12–15] have been proposed by underwater robotics researchers to steer a team of autonomous vehicles along a specified path and thus performing a mission such as adaptive sampling. For many of these solutions, inter-vehicle communications are assumed to be ideal (i.e., no packet loss, no delay, etc.) or are based on ideal graph theory models. Therefore, it is not clear how well they perform using real underwater communications. There are also some solutions such as [9] that rely on terrestrial communication techniques (e.g., satellite communications) to exchange inter-vehicle information. In this case, these vehicles have to surface, thus wasting more energy and time (not to mention the risk that – as it has happened – the vehicle is stolen by pirates or damaged by vandals).

To overcome the limitation of using theoretical communication models and relying on radio communication techniques, we introduce innovative coordination algorithms using underwater communication techniques to support swarming of a realistically limited number of underwater gliders (less than ten). Specifically, we propose: 1) a *team formation algorithm* to move the gliders into the specified geometry in minimal time and without collisions, and 2) an *attraction and repulsion swarming algorithm* to steer gliders while maintaining the formation. Underwater acoustic communication techniques are combined with these algorithms in order to improve the performance of vehicle coordination. For team formation, a packet type that performs well for long-range communications is used. For team steering, the relative locations of AUVs are estimated from the Doppler shifts extracted from ongoing opportunistic inter-vehicle communications.

The contribution of our solution is the following: 1) Our team formation and team steering algorithms use real underwater acoustic modems and are combined with more realistic underwater communication models. Therefore our solution is closely integrated with realistic underwater communications. 2) We design novel underwater acoustic communication techniques to improve the performance of inter-vehicle communications. For example, reliable short FSK-modulated packets are used for long-range communication during team formation. 3) We propose a hybrid team steering scheme based on the Doppler shifts extracted from ongoing opportunistic inter-vehicle communications. These Doppler shifts are then used to estimate the relative locations of the AUVs, which are then fed back for distributed steering control.

The structure of this dissertation is outlined here as the following. In Chapter 2, we review the existing solutions for UW-ASNs, especially for geographic routing, underwater delay-tolerant networks, geocasting, and underwater localization. We also review and discuss the existing work in team coordination algorithms for AUVs. The difference between our solution and existing solutions will be presented and discussed.

In Chapter 3, we introduce the distinction between two forms of position uncertainty and how we model the position uncertainty using statistical methods - first for underwater gliders and then for general types of AUVs whose trajectories are predictable. Then based on this uncertainty model, we propose an energy consumption minimization framework that can serve two-classes of delay-tolerant traffic. The proposed framework also uses acoustic directional transducers to reduce the computation and communication overhead for inter-vehicle data transmission. We then propose an efficient geocasting solution that can reliably forward packets to AUVs within a specified region. An under-ice/underwater localization solution is also proposed to minimize the localization uncertainty underwater, especially for scenarios where localization infrastructure is unavailable. We further propose a team formation and steering solution for underwater gliders using underwater acoustic modems, which is robust against acoustic channel impairments and ocean currents. Details of the proposed solutions are presented and discussed.

In Chapter 4, we first propose a novel underwater communication emulator for performance evaluation. This emulator is then used to evaluate the solutions we proposed for inter-vehicle communications and coordination. We first evaluate the accuracy of our statistical model in estimating the position uncertainty. It is shown that our inter-vehicle optimization framework is effective in providing QoS services to the loss-sensitive and loss-tolerant traffic in delay-tolerant applications. Performance of the inter-vehicle communication optimization framework is improved over some well-known geographic routing and DTN approaches. The performance of the geocasting solution and underwater localization solution are also compared against existing solutions. Our team formation and steering algorithms are shown to be robust against acoustic channel impairments and ocean currents. It is also shown that our coordination solution using acoustic communications is performing better than existing solutions using only satellite communications.

In Chapter 5, we draw conclusions on our work. We proposed a statistical approach to model an AUV's position uncertainty. And based on this model, inter-vehicle communication solutions - which include an energy-efficient optimization framework, reliable geocasting and position uncertainty minimization underwater localization - and team coordination solution are proposed. Future work includes implementation of these communication and coordination solutions on different types of AUVs and evaluation of their performance, design of adaptive sampling solutions to efficiently sample the ocean, and integration of the AUVs with static UW-ASNs and existing ocean observation infrastructures.

Chapter 2

Related Work

In this chapter, we review the related work of our dissertation, first the related approaches for geographic routing, delay-tolerant networks, geocasting, underwater localization, and then the related solutions for team coordinations. An overview of communication solutions for different layers of UW-ASNs has been provided in [2, 16]. Here we focus on the work closely related to our solutions.

2.1 Geographic Routing Protocols

Geographic routing protocols rely on geographic position information for message forwarding, which requires that each node can determine its own location and that the source is aware of the location of the destination. Many geographical routing schemes, including some well-known ones such as Most Forward within Radius (MFR) scheme [17], Greedy Routing Scheme (GRS) [18] and Compass Routing Method (CRM) [19], have been proposed for terrestrial wireless networks. In MFR, the message is forwarded to the neighbor that is closest to the destination, while in GRS a node selects the neighbor whose projection on the segment from the source to destination is closest to the destination. In the Compass Routing Method (CRM) [19], a message is forwarded to a neighbor whose direction from the transmitter is the closest to the direction to the destination. In [20], a scheme called Partial Topology Knowledge Forwarding (PTKF) is introduced, and is shown to outperform other existing schemes in typical application scenarios. Based on the estimate using local neighborhood information, PTKF forwards packet to the neighbor that has the minimal e2e routing energy consumption. These solutions are proposed for terrestrial wireless networks. In UW-ASNs, they may not work well since propagation of acoustic signals is quite different from that of radio signals. Moreover, localization underwater is generally more difficult than in the terrestrial environment.

In [21], a geographic routing protocol based on Dynamic Source Routing (DSR) with location awareness is proposed. It employs the range information to estimate local network topology, showing improved network capacity over blind flooding and DSR protocols as node distance becomes large. Focus Beam Routing (FBR) protocol [22] employs a directional beamforming technique with the help of location information for packet forwarding. It is shown that FBR has energy consumption performance close to Dijkstra's algorithm while additional burden of dynamic route discovery is minimal.

2.2 Delay-Tolerant Networking Approaches

Solutions for DTNs have been proposed for communications within extreme and performance-challenged environments where continuous e2e connectivity does not hold most of the time [23,24]. Many approaches such as Resource Allocation Protocol for Intentional DTN (RAPID) routing [25], Spray and Wait [26], and MaxProp [27], are solutions mainly for intermittently connected terrestrial networks. RAPID [25] translates the e2e routing metric requirement such as minimizing average delay, minimizing worst-case delay, and maximizing the number of packets delivered before a deadline into per-packet utilities. At a transfer opportunity, it replicates a packet that locally results in the highest increase in utility. Spray and Wait [26] "sprays" a number of copies per packet into the network, and then "waits" until one of these nodes meets the destination. In this way it balances the tradeoff between the energy consumption incurred by flooding-based routing schemes and the delay incurred by spraying only one copy per packet in one transmission. MaxProp [27] prioritizes both the schedule of packets transmissions and the schedule of packets to be dropped, based on the path likelihoods to peers estimated from historical data and complementary mechanisms including acknowledgments, a head-start for new packets, and lists of previous intermediaries. It is shown that MaxProp performs better than protocols that know the meeting schedule between peers. These terrestrial DTN solutions may not achieve the optimal performance underwater as the characteristics of underwater communications are not considered. Hence, in the rest of this section, we focus on related solutions for UW-ASNs.

Several DTN solutions for UW-ASNs have been proposed in [28–31]. In [28], an energy-efficient protocol is proposed for delay-tolerant data-retrieval applications. Efficient erasure codes and Low Density Parity Check (LDPC) codes are also used to reduce Packet Error Rate (PER) in the underwater environment. In [29], an adaptive routing algorithm exploiting message redundancy and resource reallocation is proposed so that ‘more important’ packets can obtain more resources than other packets. Simulation results showed that this approach can provide differentiated packet delivery according to application requirements and can achieve a good e2e performance trade-off among delivery ratio, average e2e delay, and energy consumption. A Prediction Assisted Single-copy Routing (PASR) scheme that can be instantiated for different mobility models is proposed in [30]. An effective greedy algorithm is adopted to capture the features of network mobility patterns and to provide guidance on how to use historical information. It is shown that the proposed scheme is energy efficient and cognizant of the underlying mobility patterns.

In [31], an approach called Delay-tolerant Data Dolphin (DDD) is proposed to exploit the mobility of a small number of capable collector nodes (namely dolphins) to harvest information sensed by low power sensor devices while saving sensor battery power. DDD performs only one-hop transmissions to avoid energy-costly multi-hop relaying. Simulation results showed that limited numbers of dolphins can achieve good data-collection requirements in most application scenarios. However, data collection may take a long time as the nodes need to wait until a dolphin moves into the communication ranges of these nodes.

Compared to the number of approaches using directional antennae for terrestrial wireless sensor networks, solutions using directional transducers for UW-ASNs are very limited due to the complexity of estimating position and direction of vehicles underwater. Moreover, these solutions generally assume the transducers are ideally directional. That is, they assume the radiation energy of the transducer is focused on some angle range with no leaking of radiation energy outside this range. For example, such transducers are used for localization using directional beacons in [32] and for directional packet forwarding in [33]. These solutions also use only one frequency. In this work, rather than using the ideal transducer model, we consider the radiation patterns of existing real-world transducers at different frequencies in order to minimize energy consumption.

A cross-layer optimization solution for UW-ASNs has been proposed in [10], where the interaction between routing functions and underwater characteristics is exploited, resulting in improvement in e2e network performance in terms of energy and throughput. A study on the interaction between physical and Medium Access Control (MAC) layers is presented in [34], where a method is proposed to estimate the battery lifetime and power cost for shallow-water UW-ASNs. In this way, the energy consumption is equalized and the network lifetime is prolonged. A cross-layer approach that improves energy consumption performance by jointly considering routing, MAC, and physical layer functionalities is proposed in [22]. These solutions, however, do not consider uncertainty in the AUV positions and are implemented and tested only by software simulation platforms. On the contrary, we propose a practical uncertainty-aware cross-layer solution called QUO VADIS (see Chapter 3.4) that incorporates the functionalities of the WHOI Micro-Modem [35] to minimize energy consumption. Moreover, our solution is implemented on real hardware and tested in our emulator integrating WHOI underwater acoustic modems.

2.3 Geocasting Algorithms

Compared to terrestrial wireless networks, where a number of geocasting protocols have been proposed, geocasting for underwater sensor networks is a relatively unexplored area. Geocasting algorithms designed for terrestrial networks do not work well in the underwater environment due to its unique propagation characteristics. So far no geocasting solution has been tailored for underwater sensor networks. In this section, we review the solutions for terrestrial networks and discuss why they would fall short in the underwater environment.

Ko and Vaidya proposed two location-based multicast algorithms in [36], where only nodes within the so-called *forwarding zone* are allowed to relay the packets to the geocasting region. In this way, the number of nodes for forwarding packets is reduced, resulting in reduced delivery overhead. Moreover, the authors proposed GeoTORA in [37], which is based on the Temporally-Ordered Routing Algorithm (TORA) unicast routing protocol [38]. Flooding is incorporated into TORA, but it is limited to nodes within a small region. This integration significantly reduce the overhead of geocasting delivery, while maintaining high accuracy. These two solutions focus mainly on limiting the traffic in a region and on selecting appropriate routes for geocasting without considering the communication link properties. Hence, due to the communication impairments of the underwater acoustic channel, they may not be able to achieve minimum geocasting delay.

Two geocasting algorithms are presented in [39] to forward packets to the neighbors that may be closest to a possible location of the destination. These neighbors are determined using Voronoi diagrams; the algorithm is further modified to ensure that the routes are loop free. Simulations are performed for two basic scenarios, one for geocasting and reactive routing, and the other for proactive routing; both showed to have high success rate and low flooding rate compared to other similar methods. They are both designed using graph theory and consider only the connectivity.

Three approaches of geocasting that can guarantee delivery are designed and discussed in [40], two of which are face traversal schemes based on depth-first search of the face tree and traversal of all faces that intersect the border of the geocasting region, respectively. In the entrance zone multicasting-based approach, the monitoring center divides the entrance ring of a geocasting region into zones of diameter equal to the transmission radius and then the problem is converted into the problem of first multicasting packets to nodes at the center of each zone and then flooding from these nodes. Improvements to all methods are made by applying neighbor or area dominating sets and coverage, and converting nodes that are not selected to sleep mode.

A delivery guaranteed geocasting solution is proposed in [41], which finds a connected planar subgraph of the network and then apply routing algorithms for planar graphs on this subgraph. While the solution requires no duplication of packets or memory at the nodes, a packet is always guaranteed to be delivered to its destinations, as confirmed through simulations. This solution is also designed based on graph theory by considering only the connectivity.

An Obstacle-Free Single-destination Geocasting Protocol (OFSGP) is proposed for mobile ad hoc networks in [42] to keep messages away from obstacles by creating a very small flooding region, and an Obstacle-Free Multi-destination Geocasting Protocol (OFMGP) is proposed for relaying messages from the source to all hosts located in multiple disconnected geographical regions, where a shared path for different destinations is created so that the number of flooding packets can be reduced as much as possible. Simulation results demonstrate that the proposed protocols transmit the message from source host to one or more destination regions with low flooding overhead and with a high success rate.

In [43], a Virtual Surrounding Face Geocasting (VSFG) algorithm is proposed to guarantee message delivery while keeping the flooding overhead low. The network area is partitioned into a set of faces, where a face is a continuous area enclosed by a sequence of edges. In VSFG, all the faces intersecting with a geocasting region are merged into a unique Virtual Surrounding Face (VSF) containing this geocasting region. By traversing all the boundary nodes of VSF and performing restricted flooding within the geocasting region, all nodes are guaranteed to receive the message. The proposed VSFG is evaluated through theoretical analysis and comprehensive simulations showing up to 40% of reduction in the number of transmissions.

Geographic Multicast Routing (GMR) [44] is proposed to construct trees with minimal bandwidth for wireless sensor networks. GMR selects the set of next hop neighbors that minimizes the so-called *cost over progress ratio*, i.e., the ratio of the number of neighbors selected for relaying to the overall reduction of the remaining distances to destinations. In this way the tradeoff between the bandwidth of the multicast tree and the effectiveness of the data distribution is better handled. The simulation results show that GMR achieves less cost of the trees and computation time in a number of networking scenarios than the position based multicast protocol.

All these protocols are designed mainly based on ideal graph theory models that only considers the connectivity. Link characteristics such as bandwidth, delay, and packet loss rate are not considered and hence the impact of link level constraints, which is not negligible in underwater acoustic communications, is ignored. Consequently, their performance may not be optimal in the underwater environment. In this work, we attempt to fill this gap and propose a geocasting solution by taking the large propagation delay, low bandwidth, and high packet loss rate constraints into account.

2.4 Underwater Localization

Localization is essential for underwater vehicle navigation and UW-ASNs, where many localization solutions, as summarized in [45] and [46], have been proposed. Due to space limitation, we just review the work that is most related, i.e., localization in UW-ASNs using AUVs.

Short Baseline (SBL) and Long Baseline (LBL) systems [45] are standard ways to localize vehicles underwater, where external transponder arrays are employed to aid localization. In SBL systems, position estimate is determined by measuring the vehicle's distance from three or more transponders that are, for example, lowered over the side of the surface vessel. LBL systems are similar to SBL, with the difference that an array of transponders is tethered on the ocean bed with fixed locations.

In [47], a localization scheme called AUV Aided Localization (AAL) is proposed, where position estimation is done using a single AUV. In AAL, an AUV navigates a predefined trajectory, broadcasts its position upon a node's request, and fixes its own position at the surface. Each node estimates the distances to the AUV while the AUV is at different locations, using the Round-Trip Time (RTT) between itself and the AUV. Algorithms such as triangulation or bounding box can then be used for position estimate. Another localization solution called Dive-N-Rise Localization

(DNRL) is proposed for both static and mobile networks in [48]. DNRL is similar to AAL, with the difference that ocean currents are considered and time synchronization is required between nodes.

In [49], an online algorithm for cooperative localization of submerged AUVs is designed, implemented, and evaluated through experiments. This algorithm relies on a single surface vehicle called Communication and Navigation Aid (CNA) for autonomous navigation. Using the CNA's GPS positions and basic onboard measurements including velocity, heading and depth, this algorithms can use filtering techniques such as Extended Kalman Filter (EKF) to bound the error and uncertainty of the on-board position estimates of a low-cost AUV.

Among existing underwater localization techniques (which are generally not suitable for under-ice environments), relatively few under-ice localization techniques have been proposed. Despite these efforts, the technology remains expensive and out of reach for researchers. Current techniques employed in the under-ice environment include combinations of either dead-reckoning using inertial measurements, sea-floor acoustic transponder networks such as SBL or LBL, and/or a Doppler Velocity Log (DVL) that can be either seafloor or ice relative [45]. These current approaches require external hardware, are cost prohibitive, and suffer from error propagation. For accurate dead reckoning, highly accurate sensors are required because magnetic navigation systems are subject to local magnetic field variations and gyros are subject to drift over time. Quality inertial navigation sensors often cost more than \$10,000 [45]. In contrast, our solution is much more economical as it does not require these expensive sensors.

Two solutions for underwater collaborative localization using a probability framework are proposed in [50] and [51], where a sum-product algorithm and a Markov process that are based on the so-called factor graph are used to model the joint distribution of multiple nodes. Both solutions require the global information of the nodes that are involved in localization, which leads to high computation complexity and communication overhead. Our solution offers another probability framework that leverages the self-estimated uncertainty distribution for estimation of other nodes. Therefore, global information is not required, resulting in reduced computation complexity and communication overhead.

2.5 Team Coordination Algorithms

Cooperation of a team of AUVs to efficiently complete underwater missions such as adaptive sampling [9, 52] has attracted many researchers. For example, a solution was proposed for cooperative control of multiple vehicles based on virtual bodies and artificial potentials [9]. However, the control is achieved through satellite communication, which is not available underwater. Periodically, these AUVs have to surface to update their GPS location and mission plan. The control of the AUVs is not in real time, therefore team formation and steering error due to unpredictable events such as variations of ocean currents cannot be fixed in real time.

In [13], research work in the European Union Project GREX, which focuses on the coordination and control of cooperating heterogeneous Autonomous Marine Vehicles (AMVs) in uncertain environments, is summarized. A general architecture for cooperative AMV control in the presence of time-varying communication topologies and communication losses is proposed. The simulation results with the networked marine system simulator and the real sea-experiment results are presented and show the efficacy of the algorithms developed for cooperative motion control. Some theoretical and practical implementation issues have, however, been raised.

A leader-follower approach is proposed in [15] for multi-AUV coordination using underwater communications. Specifically, two control algorithms are designed for two scenarios using two AUVs. The effectiveness of both algorithms are verified only in simulations.

In [14], a solution is proposed to address the problem of steering a group of vehicles along given spatial paths while holding a desired geometrical formation pattern (i.e., the path following problem). The solution is built on Lyapunov-based techniques and addresses explicitly the constraints imposed by the topology of the inter-vehicle communications network. By decoupling the path-following and coordinated control system, the dynamics of each AUV can be dealt with by each vehicle controller locally at the path-following control level, while coordination can be achieved using a decentralized control law whereby the exchange of data among the vehicles is kept at a minimum. The effectiveness of the proposed solution is verified by simulations. However, as the communication impairments are based on graph theory, the proposed solution needs to be extended to handle stringent underwater communication constraints. Therefore, it is unclear how well the proposed solution performs in real underwater communication environments.

In [53], the problem of team formation from initial to target formation positions under the influence of external disturbances is studied. An event-based approach is proposed, which relies on an uncertainty model to trigger surfacing events so that AUVs can measure their own position and update their control signal. A method is also proposed to characterize the disturbance set using these events. Communications between AUVs are modeled with network adjacency matrix and are limited to the time when the AUVs surface. Numerical examples on relevant scenarios are also provided.

Many of the above approaches use ideal graph theory models to model underwater acoustic communications. Therefore, it is not clear how well these solutions perform in real underwater acoustic communications. Conversely, here we propose a team steering and coordination solution that is based on realistic underwater acoustic communication models and that uses real underwater acoustic modems. In this way, we are able to assess the impact of the impairments of underwater communications on the coordination of AUVs.

Chapter 3

Proposed Inter-Vehicle Communication and Coordination Solutions

In this chapter, we present the details of our solutions for inter-vehicle acoustic communication and coordination algorithms. We first distinguish position uncertainty into two classes, and we introduce the statistical approach to model the position uncertainty of an underwater glider and extend the model for a general AUV whose trajectory is predictable, then we present our inter-vehicle communication solutions, which include QUO VADIS - the QoS-aware inter-glider communication optimization solution using frequency-dependent directional acoustic transducers, a reliable geocasting solution for AUVs with high position uncertainty, and an under-ice/underwater localization solution that minimizes the localization uncertainty. These solutions can satisfy the following need to facilitate the coordination of multiple AUVs: 1) the need to optimize acoustic communications among AUVs, 2) the need to send query or commands to these AUVs, e.g., multicasting to AUVs located in a specified region; and 3) the need to minimize localization uncertainty, e.g., for the under-ice environment where deployment of localization infrastructure is difficult, respectively. Based on these communication techniques, we present our algorithms for team formation and steering in the end.

3.1 Two Forms of Position Uncertainty

As introduced in Chapter 1, the position uncertainty for underwater vehicles should be considered in order to improve the performance of solutions for problems in a variety of research areas. Due to the propagation delay and loss of position information, the position uncertainty as seen by a node itself and that as seen by other nodes are generally different. Hence it is necessary to distinguish between these two forms of uncertainty. By distinguishing between internal and external uncertainty, the causes of the position uncertainty can be understood and different components of the position uncertainty can be modeled and derived. Solutions can further be designed to reduce the uncertainty

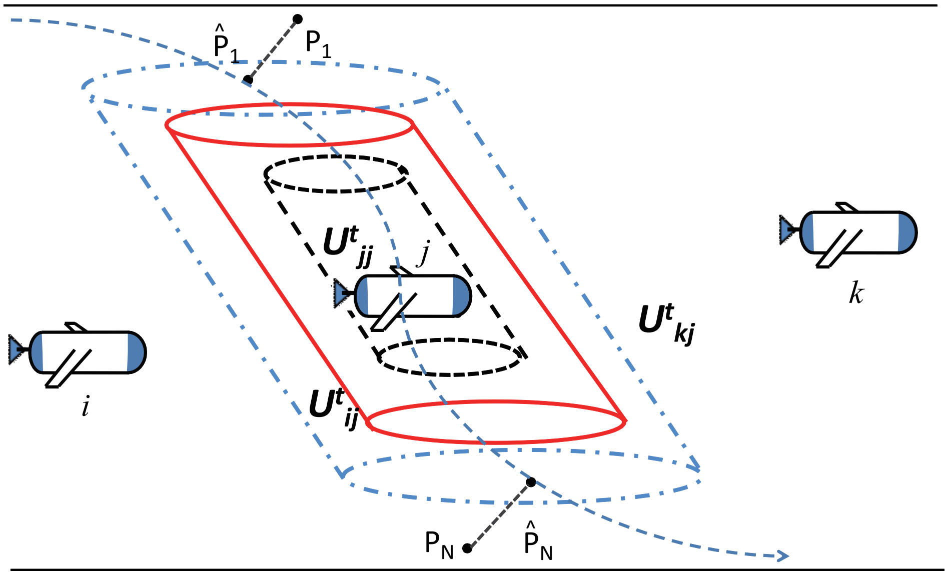


Figure 3.1: Internal and external uncertainty regions (superscript t is used to clarify that they are estimated at time t; this is used when necessary).

of different components so that such uncertainty can be minimized for performance improvement.

In this section, we first introduce the two forms of position uncertainty, discuss the relationship between them, and comment on the benefits of using the external uncertainty in a variety of relevant research areas.

3.1.1 Internal and External Uncertainties

Internal uncertainty is the position uncertainty associated with a particular entity/node (such as an AUV) *as seen by itself*. Many approaches such as those using Kalman Filter (KF) [54, 55] have been proposed to estimate the internal uncertainty assuming that the variables to be estimated have linear relationships between each other and that noise is additive and Gaussian. While simple and quite robust, KF is not optimal when the linearity assumption between variables does not hold. On the other hand, approaches using non-linear filters such as the extended or unscented KF attempt to minimize the mean squared errors in estimates by jointly considering the navigation location and the sensed states/features such as underwater terrain features, which are non-trivial, especially in an unstructured underwater environment.

External uncertainty is the position uncertainty associated with a particular entity/node *as seen by others*. Let us denote the internal uncertainty, a 3D region associated with any node $j \in \mathcal{N}$ ($\mathcal{N}$ is

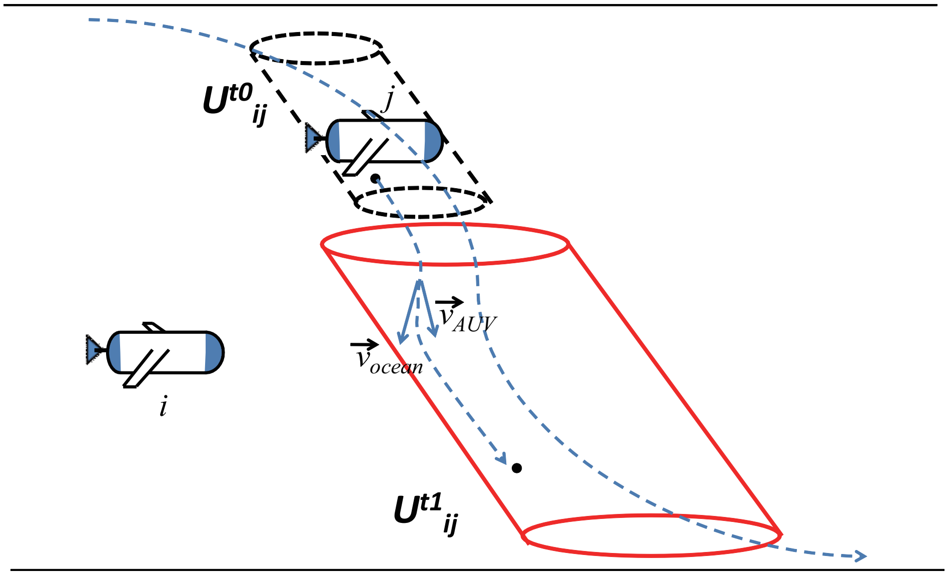
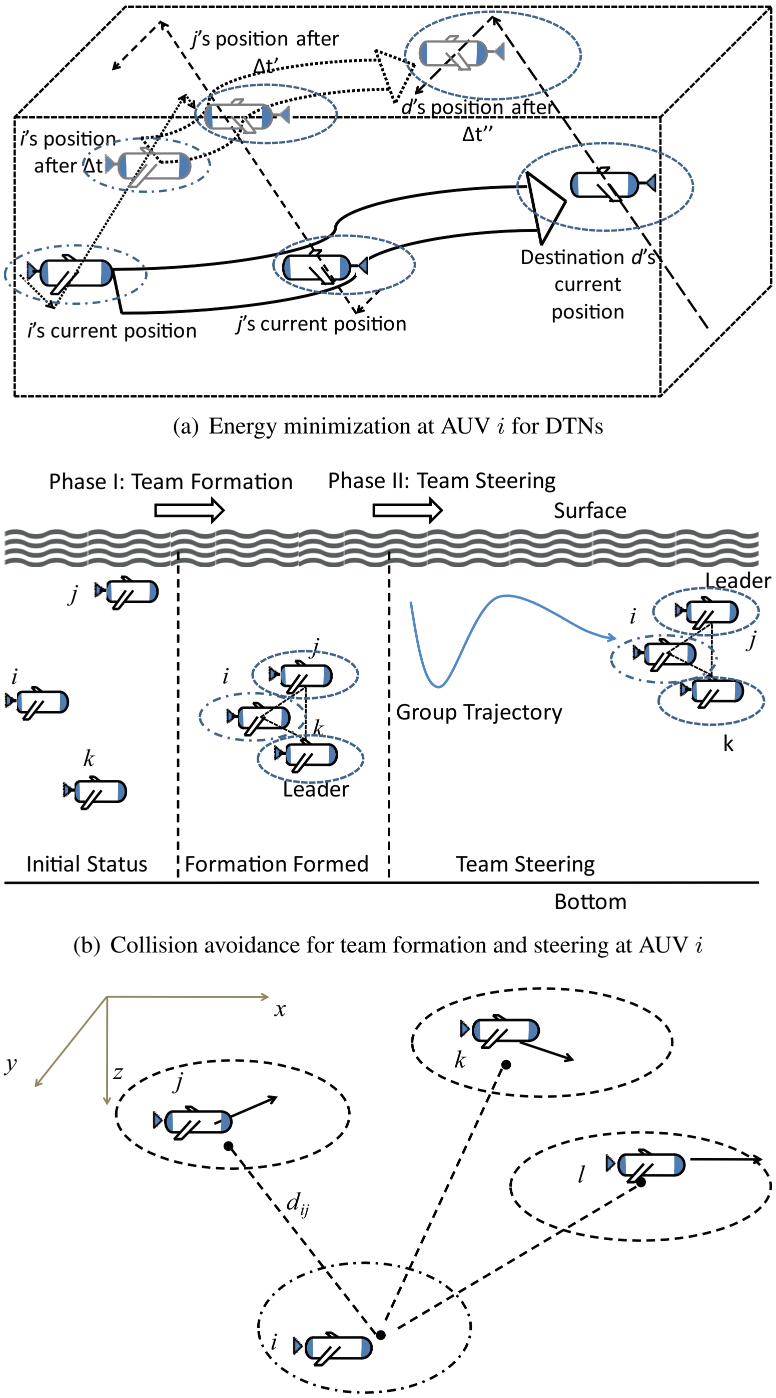


Figure 3.2: Variation of external uncertainty regions over time.



(a) Energy minimization at AUV i for DTNs

(b) Collision avoidance for team formation and steering at AUV i

(c) Uncertainty minimization in underwater localization at AUV i

Figure 3.3: Research areas that can benefit from the notion of external uncertainty (broken-line circles denote external uncertainty and broken-dotted-line circles denote internal uncertainty; note that we use circles instead of 3D shapes for uncertainty regions just for the sake of visualization simplicity).

the set of network nodes), as $\mathcal{U}_{jj}$, and the external uncertainties, 3D regions associated with j as seen by $i, k \in \mathcal{N}$, as $\mathcal{U}_{ij}$ and $\mathcal{U}_{kj}$, respectively ($i \neq j \neq k$). In general, $\mathcal{U}_{jj}$, $\mathcal{U}_{ij}$, and $\mathcal{U}_{kj}$ are different from each other; also, due to asymmetry, $\mathcal{U}_{ij}$ is in general different from $\mathcal{U}_{ji}$. External uncertainties may be derived from the broadcast/propagated internal-uncertainty estimates (e.g., using *one-hop or multi-hop neighbor discovery mechanisms*) and, hence, will be affected by e2e *network latency* and *information loss*.

The estimation of the external-uncertainty region $\mathcal{U}_{ij}$ of a generic node j at node i (with $i \neq j$) involves the participation of both i and j. Figure 3.1 illustrates the internal and external uncertainty regions and their difference. In this figure, j's uncertainty regions seen by j itself (internal uncertainty), by i (i.e., $\mathcal{U}_{ij}$) and by k (i.e., $\mathcal{U}_{kj}$) are all depicted to be different (general case). Note that as shown in Fig. 3.2, in general, the longer the AUV stays underwater, the more uncertain we have for its position (for both external and internal uncertainty). Estimation of $\mathcal{U}_{ij}$ involves estimation of the change of $\mathcal{U}_{jj}$ with time and in this work a solution is proposed to predict $\mathcal{U}_{ij}$ of the general type of vehicle with predictable trajectory based on the statistical estimation of $\mathcal{U}_{jj}$.

3.1.2 Benefits to Underwater Applications

We present here a variety of applications or research areas where the proposed external uncertainty can be applied to improve performance.

Communication protocols for UW-ASNs: In UW-ASNs, external uncertainty can be used to improve the performance of networking solutions. For example, as shown in [11], a solution that considers external uncertainty can be used for Delay-Tolerant Networks (DTNs) [1]. As shown in Fig. 3.3(a), by leveraging the predictability of AUVs' trajectories, a novel solution to dynamically optimize communications by delaying packet transmissions for the optimal network topology (i.e., trading e2e delay for throughput and/or energy consumption) can minimize communication energy consumption for two classes of delay-tolerant traffic (loss-tolerant and loss-sensitive). Specifically, the external uncertainty can be used to predict the position of an AUV and to adjust better the transmission power. Consequently, routing error can be reduced, which decreases energy/bandwidth

[1] Note that the external uncertainty used in [11] is an ideal version of the external uncertainty, assuming it be the same as the internal uncertainty. However, due to the delay incurred by large propagation delay, this external uncertainty estimate is inaccurate. In this work, we propose a solution to predict the external uncertainty from the internal uncertainty estimated for the generalized AUV with predictable trajectory.

utilization.

Underwater robotics: In underwater robotics, a team of AUVs can collaborate to explore a 3D region and take measurements in space and time. In order to derive the spatio-temporal correlation of the measurements, these AUVs need to keep a geometric formation and steer through the region, as shown in Fig. 3.3(b). They also need to keep a distance between each other in order to avoid vehicle collisions. In [56], a solution is proposed to minimize the time to form the geometric formation while avoiding vehicle collisions. However, in that solution, the gliders are assumed to have correct location information, which is a strong requirement for the underwater environment. The solution can be made more robust against ocean currents and acoustic channel impairments by exploiting the concept of external uncertainty, e.g., a control algorithm can be designed to minimize the probability that two AUVs are within the collision region. This concept can also be used for adaptive sampling solutions, where sampling strategy is changed based on the variation of underwater measurements. For example, for real-time ocean forecasting, a team of AUVs can be deployed to take spatial and temporal measurements. To maximize the forecasting performance, the geometry formation and inter-vehicle distance can be dynamically adjusted to measure a dynamically changing region. Using the external uncertainty notion, we can estimate the probability distribution of the measurements (since location of the measurement is the same as the AUV) and then design an optimal strategy to minimize the observation uncertainty for accurate forecasting.

Underwater localization: In the underwater environment, AUVs may need to rely on other anchor nodes (e.g., AUVs) – whose positions may not be accurate – for localization, as shown in Fig. 3.3(c). Localization error, however, may increase if an AUV relies on anchors with large position uncertainty. The external uncertainty notion can be used to decrease localization error and computation complexity. For example, a vehicle can select the optimal subset of anchors (typically with small external uncertainty) so to minimize the new internal uncertainty.

Task allocation: The proposed notion of uncertainty can also be applied in task allocation, whose objective is to choose a subset of vehicles to accomplish reliably a mission with specific requirements. For example, only part of a team of AUVs can be selected to conduct a critical mission while trying to maximize the remaining energy after the mission or to minimize the time to complete the mission [57]. In such an application, using the external uncertainty notion for position estimate, a team of AUVs that are “closer” to the target can be selected, which may lead to less time and/or energy to complete the mission.

Data processing and visualization: Once the measurements taken are received by the onshore stations, oceanographers need to visualize and analyze sensor data for a multitude of ocean science studies. The external-uncertainty notion can improve the quality of 3D data reconstruction because it can be used to show the vehicle deviation from the original mission path.

3.2 Estimation of Trajectory and Uncertainty for Gliders

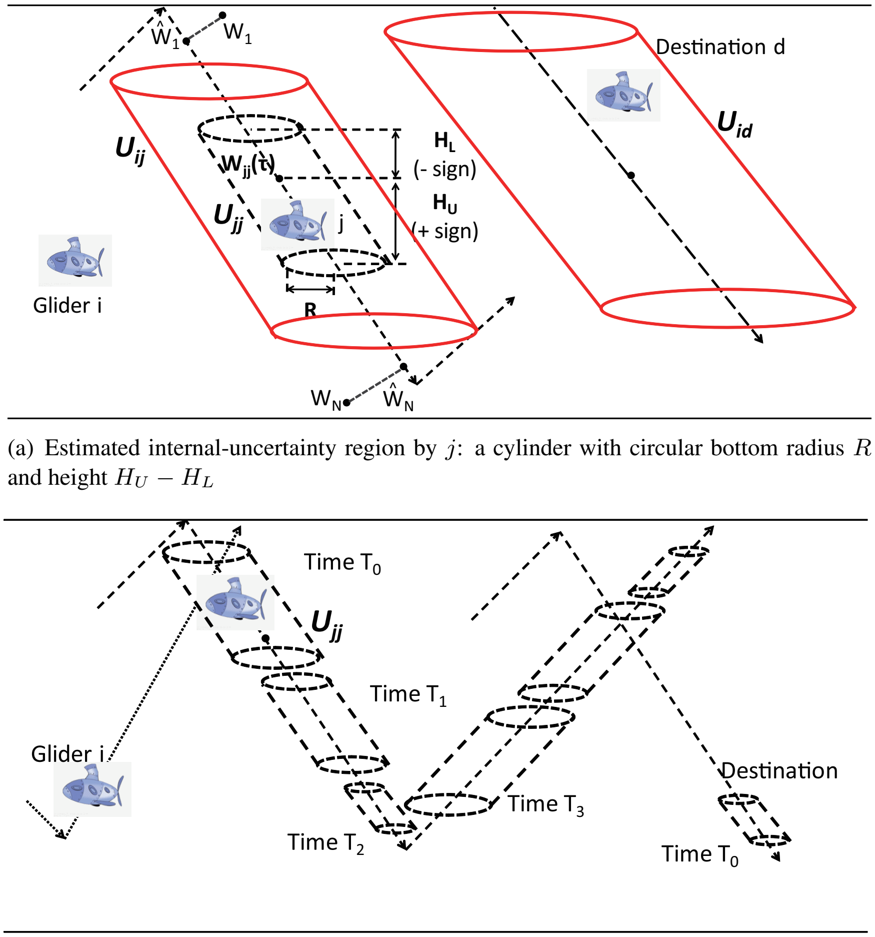
In this section, we present the statistical approach for external-uncertainty region estimation when *gliders are used as AUVs* and ocean currents are unknown.

3.2.1 Internal and External Uncertainty for Gliders

As mentioned before, the estimation of the external-uncertainty region $\mathcal{U}_{ij}$ of a generic node j at another node i (with $i \neq j$) involves the participation of both i and j. Node j will first estimate its positions at different points in time, its trajectory, and its internal-uncertainty region $\mathcal{U}_{jj}$; then, it will broadcast the parameters describing this region in its neighborhood. Upon receiving j's internal-uncertainty region parameters, glider i will estimate the external-uncertainty region of j, $\mathcal{U}_{ij}$. Here, we provide a solution for internal- and external-uncertainty estimation when *1) gliders are used* (which move in a predictable 'sawtooth' trajectory) and *2) ocean currents are unknown* (Fig. 3.4).

Internal-uncertainty estimation at j: Assume gliders estimate their own locations over time using some localization algorithm, e.g., dead reckoning (the localization method used on the SLOCUM glider today). Glider j's estimated coordinates, $W_n = (x_n, y_n, z_n)$ at sampling times t_n ($n = 1 \dots N$), are used to estimate its trajectory line segment. Here $t_n = t_{n-1} + \Delta$, where Δ is the update interval. From statistics, the Orthogonal Least Square (OLS) line gives the best maximum likelihood estimation [58]. In other words, the regression line $(x_0 + at, y_0 + bt, z_0 + ct)$ is the best fitting line such that the sum of the squared distances between it and W_n's is minimal. Here x_0, y_0, z_0, a, b, c are coefficients to be calculated while t is the real-valued variable for this parameterized line.

As shown in Fig. 3.5, denote the distance of W_n to the regression line by d_n, OLS regression



(a) Estimated internal-uncertainty region by j: a cylinder with circular bottom radius R and height $H_U - H_L$

(b) Change of internal-uncertainty region over time.

Figure 3.4: External- and internal-uncertainty regions for gliders under the effect of unknown ocean currents.

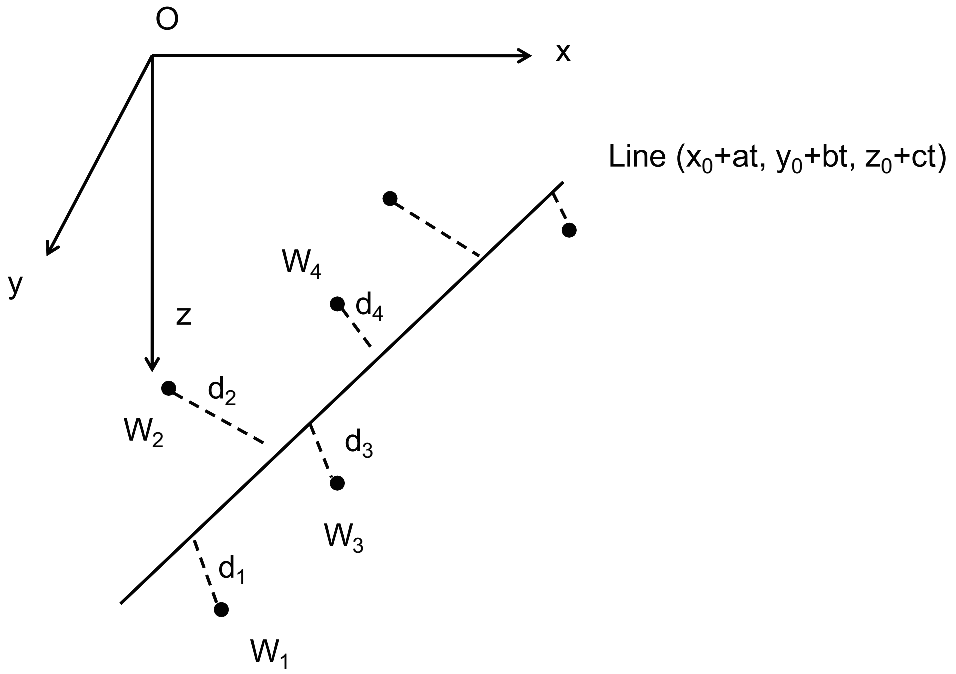


Figure 3.5: Estimated line.

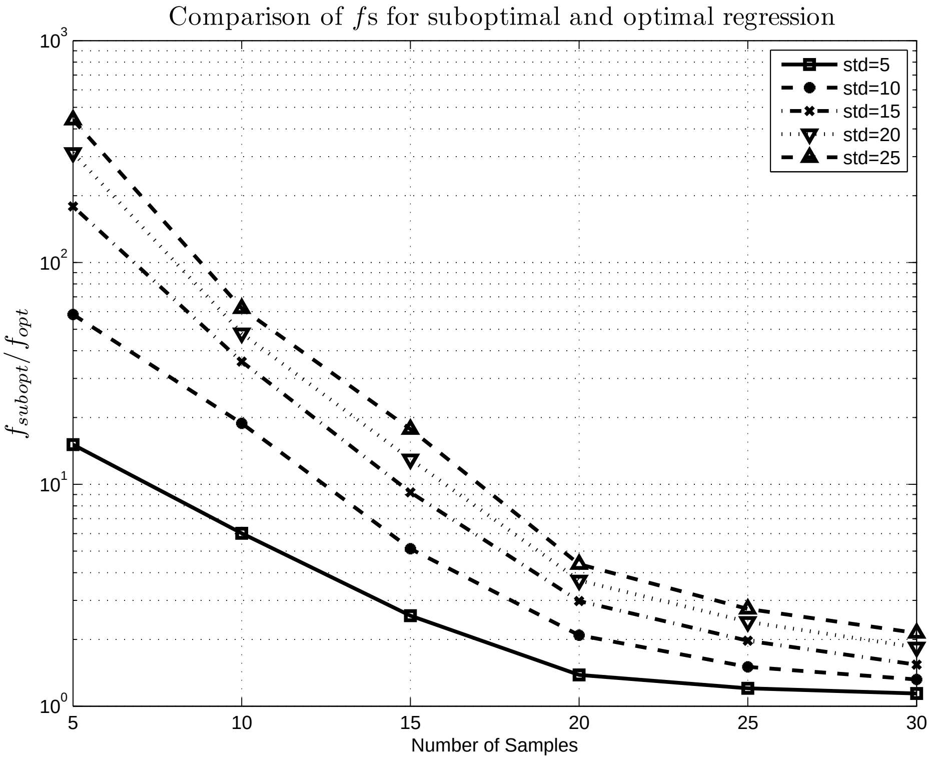


Figure 3.6: Comparison of suboptimal and optimal solution: f_{subopt} and f_{opt} are the minimal values of the suboptimal and optimal solution, respectively.

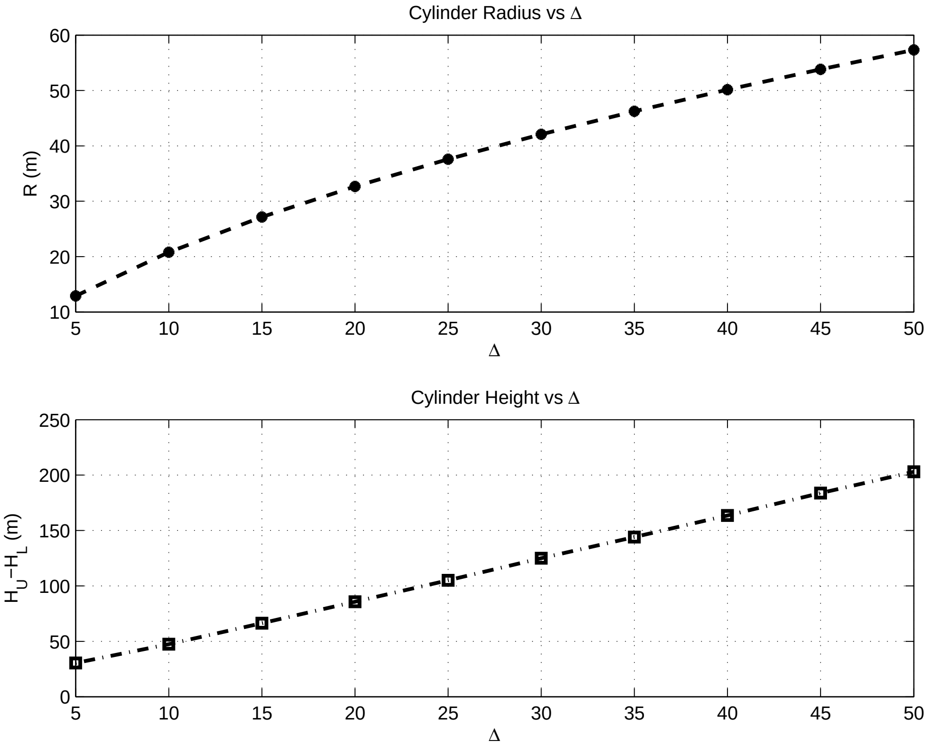


Figure 3.7: Estimated cylinder region for $\alpha = 0.05$: the larger the update interval Δ is, the bigger the estimated region is.

gives a line that minimizes the sum of the squared distances, i.e., it minimizes

$$f(x_0, y_0, z_0, a, b, c) = \sum_{n=1}^{N} d_n^2. \tag{3.1}$$

From geometry, it is not hard to derive

$$\begin{aligned} d_n^2 = \{&[c(y_n - y_0) - b(z_n - z_0)]^2 + [a(z_n - z_0) - c(x_n - x_0)]^2 \\ &+ [b(x_n - x_0) - a(y_n - y_0)]^2\}/(a^2 + b^2 + c^2). \end{aligned} \tag{3.2}$$

As a consequence, the following equations must be satisfied,

$$\frac{\partial f}{\partial x_0} = \frac{\partial f}{\partial y_0} = \frac{\partial f}{\partial z_0} = \frac{\partial f}{\partial a} = \frac{\partial f}{\partial b} = \frac{\partial f}{\partial c} = 0. \tag{3.3}$$

However, this equation group is a cubic equation group with 6 variables for which it is hard to derive a closed-form solution.

To reduce the complexity of this problem, we divide it into two sub-problems by the observation from (3.2) that x_0, y_0, z_0 and a, b, c have similar effect on d_i. Therefore, our *first sub-problem* is to find the x_0^*, y_0^*, z_0^* that minimize f by assuming a, b, c are known. Then, based on the solution on the first problem, i.e., (x_0^*, y_0^*, z_0^*), we will find the optimal a^*, b^*, c^* which can minimize f (*second sub-problem*).

Solutions for these two sub-problems are provided in [59], which shows that $[x_0^*, y_0^*, z_0^*] = [\bar{x}, \bar{y}, \bar{z}]$ and $[a^*, b^*, c^*]^T$ is the corresponding eigenvector to the maximum eigenvalue λ_{max} of $\mathbf{A}^T\mathbf{A}$, where $\bar{x} = \frac{1}{N}\sum_{n=1}^{N} x_n$, $\bar{y} = \frac{1}{N}\sum_{n=1}^{N} y_n$, $\bar{z} = \frac{1}{N}\sum_{n=1}^{N} z_n$, and

$$\mathbf{A} = \begin{bmatrix} x_1 - \bar{x} & y_1 - \bar{y} & z_1 - \bar{z} \\ x_2 - \bar{x} & y_2 - \bar{y} & z_2 - \bar{z} \\ \cdot & \cdot & \cdot \\ x_N - \bar{x} & y_N - \bar{y} & z_N - \bar{z} \end{bmatrix}.$$

Computation complexity for $[a^*, b^*, c^*]^T$ can be further reduced by the following proposition.

Proposition 1 $[a^*, b^*, c^*]^T$ *is the singular vector of* $\mathbf{A}$ *corresponding to its largest absolute singular* *value.*

Proof: Computation complexity can be further reduced using the Singular Value Decomposition (SVD) [60],

$$\mathbf{A} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^T, \tag{3.4}$$

where $\mathbf{\Lambda}$ is the diagonal singular value matrix, $\mathbf{U}$ and $\mathbf{V}$ are unitary matrices. The columns of $\mathbf{V}$ are the eigenvectors corresponding to the singular values in $\mathbf{\Lambda}$. Hence,

$$\mathbf{A}^T\mathbf{A} = (\mathbf{U}\mathbf{\Lambda}\mathbf{V}^T)^T\mathbf{U}\mathbf{\Lambda}\mathbf{V}^T = \mathbf{V}\mathbf{\Lambda}^2\mathbf{V}^T, \tag{3.5}$$

which means that the eigenvalues of $\mathbf{A}^T\mathbf{A}$ are the squares of the singular values of $\mathbf{A}$, and the eigenvectors of $\mathbf{A}^T\mathbf{A}$ are the singular vectors of $\mathbf{A}$. Hence the optimal solution (a^*, b^*, c^*) to the second subproblem is the singular vector of $\mathbf{A}$ corresponding to its largest absolute singular value. Complexity for this decomposition is $O(\min(9N, 3N^2)) = O(9N)$ when $N \geq 3$. ■

After some math manipulation, we can further normalize the coefficients of this regression line $(x_0 + at, y_0 + bt, z_0 + ct)$ so that t can be the time instead of just a parameter. Ultimately, this normalized trajectory segment can be described as $W(t) = \bar{W} + \overrightarrow{\mathbf{v}}(t - \bar{t})$, where $\bar{W} = (\bar{x}, \bar{y}, \bar{z})$ and $\overrightarrow{\mathbf{v}} = \frac{\|\overrightarrow{\widehat{W_1}\widehat{W_N}}\|}{\|(a^*,b^*,c^*)\|\cdot(t_N - t_1)} \cdot (a^*, b^*, c^*)$. Here $\bar{t} = \frac{1}{N}\sum_{n=1}^{N} t_n$ is the average of the sampling times, and $\widehat{W_i}$ is the projection of point W_i on the line segment (Fig. 3.4(a)).

The optimal and sub-optimal solutions with two sub-problems are compared in Fig. 3.6, where std is the standard deviation of the samples from trajectory. When N is small, the optimal solution gives a better fitting line. As N gets larger, f_{opt} and f_{subopt} get closer; yet, the suboptimal solution gives an algorithm that requires much less computation (see Proposition 1).

After trajectory estimation, because gliders have no knowledge about the currents affecting themselves (and the other gliders), the internal-uncertainty region of j is estimated as a *cylindrical* region[2]. This cylinder $\mathcal{U}$ is described by its radius R and its height $H_U - H_L$, where H_U and H_L (as shown in Fig. 3.4) – in general different – are the *signed distances* of the cylinder's top and

[2] If the ocean current moves in any direction in the 3D space, j's drifting can be treated as a 3D Brownian Motion where the deviations in x and y direction are identically independently distributed (i.i.d.), which makes the horizontal projection of j's confidence region circular. And as j moves along its ascending or descending trajectory, the region swept is a cylinder. Although the pressure sensor on j gives a rather accurate vertical position, there still can be vertical uncertainty due to 'upwelling' or 'downwelling' currents. The uncertainty-region shape can be made more realistic if some ocean-current knowledge is available.

bottom surface (i.e., the surface ahead and behind in the trajectory direction, respectively) to glider j's expected location $W_{jj}(\tau)$ on the trajectory at current time τ.

The problem to jointly find the H_L, H_U, and R for $\mathcal{U}_{ij}$ is complicated. We simplify it into two sub-problems that can estimate H_L, H_U, and R separately. We reason that the probability of the glider being in $\mathcal{U}_{ij}$ in the z direction is greater than $1-\alpha$ and the probability of it being within the horizontal cross section of $\mathcal{U}_{ij}$ is greater than $1-\alpha$, i.e.,

$$\begin{cases} \Pr\{H_L \leq H \leq H_U\} \geq 1-\alpha \\ \Pr\{D \leq R\} \geq 1-\alpha \end{cases}. \tag{3.6}$$

With (3.6), given a specified α, H_L, H_U, and R can be estimated using statistical inference theory [58] by relying on the available N samples. H_L and H_U can be solved as the lower and upper bounds of the two-sided confidence interval of H, while R can be solved as the bound of the one-sided confidence interval of D. The samples of H_n (for H) and R_n (for D) can be calculated from the available N position samples W_n. We have the following two propositions.

Proposition 2 *H_L and H_U are estimated by*

$$\begin{cases} H_L = \overline{H} - \hat{t}_{\alpha,N-1}S^{(H)}\sqrt{1+1/N} \\ H_U = \overline{H} + \hat{t}_{\alpha,N-1}S^{(H)}\sqrt{1+1/N} \end{cases}, \tag{3.7}$$

where $\overline{H} = \sum_{n=1}^{N} H_n/N$ is the mean of the current N samples, $S^{(H)} = [\frac{1}{N-1}\sum_{n=1}^{N}(H_n - \overline{H})^2]^{1/2}$ is the unbiased standard deviation, and $\hat{t}_{\alpha,N-1}$ is the $100(1-\alpha/2)\%$ of Student's t-distribution *[58] with $N-1$ degrees of freedom.*

Proof: Assume these H_n's are i.i.d. normal variables with the same unknown mean and variance. From predictive inference theory [58], we have

$$\frac{H-\overline{H}}{S^{(H)}\sqrt{1+1/N}} \sim t_{N-1}, \tag{3.8}$$

where t_{N-1} is the Student's t-distribution with degree of freedom $N-1$ with probability distribution function (pdf)

$$f(t) = \frac{\Gamma(N/2)}{\sqrt{(N-1)\pi}\Gamma((N-1)/2)}(1+\frac{t^2}{N-1})^{-N/2}. \tag{3.9}$$

Here Γ is the well-known *Gamma function.* Solving for H yields the following distribution

$$\overline{H} + S^{(H)}\sqrt{1+1/N} \cdot t_{N-1}. \tag{3.10}$$

Put this into the second inequality in (3.6), we can obtain (3.7). ■

Proposition 3 *R is estimated by*

$$R = \frac{\sqrt{N-1}S^{(R)}}{\sqrt{\hat{\chi}_{\alpha,2(N-1)}}}, \tag{3.11}$$

where $S^{(R)} = [\frac{1}{N-1}\sum_{n=1}^{N}(R_n - \overline{R})^2]^{1/2}$, $\overline{R} = \frac{1}{N}\sum_{n=1}^{N} R_n$, and $\hat{\chi}_{\alpha,2(N-1)}$ is the $100(1-\alpha)\%$ of χ-distribution with $2(N-1)$ degrees of freedom.

Proof: As stated before, the distributions of the glider's location in the orthogonal x,y direction of $\mathcal{U}_{ij}$ are i.i.d. normal distributions with σ_R^2. As a result, we have

$$R/\sigma_R \sim \chi_2, \tag{3.12}$$

where χ_2 is the χ-distribution with 2 degrees of freedom. Let

$$\begin{cases} \overline{R} = \frac{1}{N}\sum_{n=1}^{N} R_n \\ S^{(R)} = [\frac{1}{N-1}\sum_{n=1}^{N}(R_n - \overline{R})^2]^{1/2} \end{cases}, \tag{3.13}$$

we can obtain (3.11) from [58]. ■

As shown in Fig. 3.7, the greater Δ is, the bigger the estimated cylinder is. Glider i receives j's update messages less frequently when j moves farther away since it may need other gliders to relay these messages, therefore, $\mathcal{U}_{ij}$ will become larger. As the number of position samples becomes smaller, f will become larger and j's position estimation will become less accurate, according to Fig. 3.6. For this uncertainty, deterministic geographic routing may be a problem, while our solution offers higher success rate to forward packets to the destined glider by using a statistical region. As shown in Fig. 3.4(b), j's internal-uncertainty region becomes smaller over time (from T_0 to T_2), i.e., as more position estimates are acquired.

External-uncertainty estimation at i: After receiving j's trajectory and internal-uncertainty region parameters ($\bar{W}, \bar{t}, \overrightarrow{v}, H_U, H_L, R$), glider i can update the estimate of j's external-uncertainty region. Note that, because AUVs involved in missions show predictable trajectories, information about the sawtooth segment can be used to derive the entire glider trajectory through extrapolation assuming symmetry between glider ascent and descent. Due to packet delays and losses in the network, j's external-uncertainty regions as seen by single- and multi-hop neighbors are *delayed versions* of j's own internal uncertainty (Fig. 3.4(b)). Hence, when using *multi-hop neighbor discovery schemes*, the internal uncertainty of a generic node j, $\mathcal{U}_{jj}$, provides a *lower bound* for all the external uncertainties associated with that node, $\mathcal{U}_{ij}$, $\forall i \in \mathcal{N}$. When there is an unexpected significant change in j's trajectory, j will inform its neighbors immediately so that the other gliders will not continue to estimate the external-uncertainty region along the 'old' trajectory, i.e., before the change. In our solution, a higher queueing priority is assigned to broadcast packets containing this change of course information.

3.3 External Uncertainty Estimation for General AUVs

The previous section depicts how the internal and external uncertainties associated with the underwater glider can be estimated. In this section, we extend the position uncertainty estimation to a general type of AUV whose trajectory is predictable. As mentioned above, to obtain $\mathcal{U}_{ij}$, j needs to estimate its internal-uncertainty region $\mathcal{U}_{jj}$ first, then $\mathcal{U}_{jj}$ is broadcast to its neighbors. Upon receiving this information, i will derive $\mathcal{U}_{ij}$ based on $\mathcal{U}_{jj}$. Therefore, it is necessary to estimate $\mathcal{U}_{jj}$ first before $\mathcal{U}_{ij}$ can be derived. In this section, we first provide a statistical model for internal uncertainty estimation. Then in the next section, we propose a solution to predict the external uncertainty.

Our estimation of the internal uncertainty relies on a statistical approach to estimate the confidence region using the position estimates of an AUV itself. We assume each AUV follows a predictable trajectory. This assumption is reasonable since AUVs generally need to stay on the pre-planned path(s) to take measurements. The advance in mechanical, electrical and computer engineering technologies has made it possible for the AUVs to autonomously steer close to (or on) the pre-planned path(s) under disturbance such as drifting. Moreover, AUVs are designed to follow some regular movement pattern (e.g., saw-tooth pattern of underwater glider). Assume this predictable trajectory of an AUV can be described by a function $(x, y, z) = \mathbf{f}(t, \boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is the list of p parameters that needs to be specified by the AUV's trajectory. Furthermore, we assume $\mathbf{f}$ is differentiable except at a countable set of discrete points. This assumption generally holds since AUVs rely on forces such as mechanical propulsion and/or buoyancy for acceleration, which results in a differentiable trajectory except at points where abrupt change is made. For example, an underwater glider generally follows a saw-tooth trajectory, whose piece-wise trajectory can be described as a line (differentiable) segment $(x, y, z) = (at + x_0, bt + y_0, ct + z_0)$. In this case, $\boldsymbol{\theta} = (a, b, c, x_0, y_0, z_0)$. Assume AUV j estimates its own coordinates, $P_n = (x_n, y_n, z_n)$, at sampling times t_n ($n = 1 \dots N$), and its trajectory segment is of the form $P(t) = \mathbf{f}(t, \boldsymbol{\theta})$, we need to estimate $\boldsymbol{\theta}$ so that the trajectory can be determined. Note that the position P_n can be estimated using existing localization techniques such as dead reckoning, or long baseline navigation [45]. Based on the derived trajectory, j's internal uncertainty (i.e., confidence region) can further be estimated. From nonlinear regression theory in statistics, we estimate its trajectory using the Gauss-Newton Algorithm [61], which relies on the linear approximation using the Taylor expansion $\mathbf{f}(t, \boldsymbol{\theta}) \approx \mathbf{f}(t, \boldsymbol{\theta}^{(0)}) + \mathbf{Z}^0(\boldsymbol{\theta} - \boldsymbol{\theta}^{(0)})$, where $\mathbf{Z}^0 = \boldsymbol{Z}_{|\boldsymbol{\theta}=\boldsymbol{\theta}^{(0)}} = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\theta}}_{|\boldsymbol{\theta}=\boldsymbol{\theta}^{(0)}}$ and $\boldsymbol{\theta}^{(0)}$ is the vector to provide initial values of these parameters to start the algorithm. The objective of our estimation is to find the best $\hat{\boldsymbol{\theta}}$ such that $S(\boldsymbol{\theta}) = \sum_{n=1}^{N} \|P_n - \mathbf{f}(t_n, \boldsymbol{\theta})\|^2$ is minimized. The idea of the Gauss-Newton Algorithm is to find $\boldsymbol{\theta}$ through iteration, i.e., to update $\boldsymbol{\theta}^{(q)}$ iteratively until convergence. From nonlinear regression theory, it has been shown that $\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^{(0)} = (\mathbf{Z}^T\mathbf{Z})^{-1}\mathbf{Z}^T\boldsymbol{\epsilon}^{(0)}$, where $\boldsymbol{\epsilon}^{(0)} = P(t) - \mathbf{f}(t, \boldsymbol{\theta}^{(0)})$. Starting from the initial position, we have the following iterative formula to estimate $\boldsymbol{\theta}^{(q)}$, i.e., $\boldsymbol{\theta}^{(q+1)} = \boldsymbol{\theta}^{(q)} + \boldsymbol{\delta}^{(q)}$, where $\boldsymbol{\delta}^{(q)} = [(\mathbf{Z}^q)^T\mathbf{Z}^q]^{-1}\mathbf{Z}^q\boldsymbol{\epsilon}^{(q)}$ and $\boldsymbol{\epsilon}^{(q)} = P(t) - \mathbf{f}(t, \boldsymbol{\theta}^{(q)})$.

From statistical inference theory, we have the asymptotic formula $\hat{\boldsymbol{\theta}} \sim \mathcal{N}(\boldsymbol{\theta}, \sigma^2\mathbf{C}^{-1})$, where $\mathcal{N}()$ is the normal distribution, $\mathbf{C} = \mathbf{Z}^T\mathbf{Z}$, and σ^2 is the variance. From this equation, we have $\frac{\mathbf{a}^T\hat{\boldsymbol{\theta}} - \mathbf{a}^T\boldsymbol{\theta}}{\hat{\sigma}(\mathbf{a}^T\mathbf{C}^{-1}\mathbf{a})^{1/2}} \sim t_{N-p}$, where t_{N-p} is the t-distribution with $N - p$ degrees of freedom [61], $\mathbf{a}^T = [0, 0, \dots, 0, 1, 0, \dots, 0]$ is a vector with p elements, $\hat{\sigma} = S(\hat{\boldsymbol{\theta}})/(N - p)$. Note that this actually gives the confidence interval for each element in $\boldsymbol{\theta}$. It is also proved in [61] that asymptotically

$$\frac{\mathbf{f}(t_N, \boldsymbol{\theta}) - \mathbf{f}(t_N, \hat{\boldsymbol{\theta}})}{\hat{\sigma}\left[1 + (\mathbf{Z}_N)^T\mathbf{Z}^\mathbf{T}\mathbf{Z}^{-1}\mathbf{Z}_N\right]^{1/2}} \sim t_{N-p}. \tag{3.14}$$

An approximate $100(1 - \alpha)\%$ confidence interval at t_N is

$$\mathbf{f}(t_N, \hat{\boldsymbol{\theta}}) \pm \hat{t}_{N-p}^{\alpha/2}\hat{\sigma}\left[1 + (\mathbf{Z}_N)^T\mathbf{Z}^\mathbf{T}\mathbf{Z}^{-1}\mathbf{Z}_N\right]^{1/2}, \tag{3.15}$$

which gives the internal-uncertainty region. Here $\mathbf{Z}_N = \frac{\partial \mathbf{f}(t_N, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}}$ is the Jacobian matrix of $\mathbf{f}$ at t_N and $\hat{t}_{N-p}^{\alpha/2}$ is the $100(1-\alpha/2)\%$ of t-distribution with $N-p$ degrees of freedom. Note that the estimation of internal uncertainty does not assume a particular localization technique for AUVs, e.g., dead reckoning, GPS, particle filtering [62], among others, though different internal uncertainty regions and distributions may result.

3.3.1 Modeling External Uncertainty for General AUVs

In this part, we present our model on external uncertainty for general AUVs. We start from the estimation of one-hop external uncertainty. Then we show how this estimate can be used to dynamically adjust the update interval. Last, we extend the one-hop estimate to that of a node that is multiple hops away.

One-hop External Uncertainty Estimation

After receiving the trajectory and internal-uncertainty region parameters from AUV j, i.e., ($\hat{\boldsymbol{\theta}}$, $\mathbf{f}(t_N, \hat{\boldsymbol{\theta}}), \hat{t}_{N-p}^{\alpha/2} \hat{\sigma} \left[1 + (\mathbf{Z}_N)^T \mathbf{Z}^{\mathbf{T}} \mathbf{Z}^{-1} \mathbf{Z}_N\right]^{1/2}$) (estimated as in Sect. 3.3), AUV i can update the estimate of j's external-uncertainty region. Due to packet delays and losses in the network, j's external-uncertainty regions as seen by single- and multi-hop neighbors are *delayed versions* of j's own internal uncertainty. Hence, when using *multi-hop neighbor discovery schemes*, the internal uncertainty $\mathcal{U}_{jj}$ provides a *lower bound* for all the external uncertainties associated with that node, $\mathcal{U}_{ij}$, $\forall i \in \mathcal{N}$. In the rest of this section, we derive $\mathcal{U}_{ij}$ based on the received $\mathcal{U}_{jj}$.

We use UKF to predict how the internal uncertainty 'propagates' through the network. This is done in two steps detailed below: *(1) Region Prediction* – this is to predict the current position of an AUV assuming that its previous location is at a point in the internal-uncertainty region; then, the external-uncertainty region is obtained by taking the set containing these predicted positions; and *(2) Distribution Estimation* – this is to calculate the probability density function (pdf) of the current position by integrating the internal-uncertainty pdf over points with the same predicted position.

(1) Region Prediction: AUV i first needs to predict j's position assuming j is located at a point in $\mathcal{U}_{jj}$ and then considers the union of these predicted points. The movement model of j can be described using the following nonlinear dynamical system. The equivalent discrete-time dynamic equation can be derived as in [63] by means of the state-space method using iterations. AUV i

estimates the state from step $q = 1$ whenever $\mathcal{U}_{jj}$ is received and q is incremented until a new $\mathcal{U}_{jj}$ is received (q is reset to 1 upon receiving this information). Hence,

$$\mathbf{s_j^q} = \mathbf{F_j s_j^{q-1}} + \mathbf{o}(\mathbf{s_j^{q-1}}) + \mathbf{G u_j^{q-1}} + \mathbf{B w_j^{q-1}} \tag{3.16}$$

represents the state-transition equation for the system describing the motion of AUV j between steps $q-1$ and q. In this equation, $\mathbf{s_j^q} = [x_j^q, y_j^q, z_j^q, \dot{x}_j^q, \dot{y}_j^q, \dot{z}_j^q, v_{j,x}^{oc}, v_{j,y}^{oc}, v_{j,z}^{oc}]^T$ represents 3D position, velocity, and ocean-current velocity of AUV j at step q, $\mathbf{o}(\mathbf{s_j^{q-1}})$ is the ocean-current prediction function (which is generally nonlinear), $\mathbf{u_j^{q-1}} = [u_j^{q-1,x}, u_j^{q-1,y}, u_j^{q-1,z}]^T$ is the control input for $t \in [(q-1)T, qT)$, and $\mathbf{w_j^{q-1}} = [w_j^{q-1,x}, w_j^{q-1,y}, w_j^{q-1,z}, w_{oc,j}^{q-1,x}, w_{oc,j}^{q-1,y}, w_{oc,j}^{q-1,z}]^T$ represents discrete random acceleration caused by non-ideal noise in the control input and/or the variation in ocean current speed. Note that $\mathbf{o}(\mathbf{s_j^{q-1}})$ can be predicted using ocean-current models or data from real-time onshore ocean observing systems; also, AUVs are spaced apart so currents affecting different AUVs are generally different.

In (3.16), $\mathbf{F_j}$, $\mathbf{G}$, and $\mathbf{B}$ are matrices to adjust the state $\mathbf{s_j^q}$ according to the previous state, control input, and random acceleration noise, respectively, and are defined as

$$\mathbf{F_j} = \begin{bmatrix} \mathbf{I}_3 & T_j'\mathbf{I}_3 & T_j'\mathbf{I}_3 \\ 0 & \mathbf{I}_3 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \mathbf{G} = \begin{bmatrix} 0 \\ \mathbf{I}_3 \\ 0 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 0 & 0 \\ \mathbf{I}_3 & 0 \\ 0 & \mathbf{I}_3 \end{bmatrix},$$

where $\mathbf{I}_3$ is the 3×3 identity matrix, T_j' is the difference between the current time and the last time when $\mathcal{U}_{jj}$ was estimated or the last update time that UKF was run, i.e., $T_j' = t_{now} - t_{\mathcal{U}_{jj}}$ if i receives j's updated internal uncertainty after the last UKF update, whereas $T_j' = T$ if i does not receive j's update message, where $t_{\mathcal{U}_{jj}}$ is the time when $\mathcal{U}_{jj}$ is estimated by j and T is the UKF update interval. Note that, when used as superscript, T indicates matrix transpose; otherwise, it represents the time interval.

The variable $\mathbf{w_j^{q-1}}$ represents 3D samples of discrete time white Gaussian noise; hence, $\mathbf{w_j^{q-1}} \sim \mathcal{N}(0, \mathbf{Q})$, where $\mathbf{Q} \geq 0$ is the covariance matrix of the process. The random acceleration is also assumed to be independent on the three axes. Here we assume that an AUV can measure the ocean-current velocity using sensors such as Acoustic Doppler Current Profiler (ADCP), which are, however, expensive; for AUVs without ADCP, we can force the state for ocean current to be zero, where the model reduces to a linear KF and the effect of ocean current should be treated as noise in $\mathbf{w_j^q}$.

Note that (3.16) includes delays due to transmission, propagation, reception, and packet loss. As the ocean-current velocity is generally nonlinear, (3.16) is a nonlinear relationship between $\mathbf{s_j^q}$ and $\mathbf{s_j^{q-1}}$. Therefore, a nonlinear Kalman filter should be used. Here we use UKF because it can provide more accurate prediction than the extended KF (another type for nonlinear prediction) while having the same computation complexity $\mathcal{O}(L^3)$ for state estimation, where L is the dimension of the state variable, as proven in [64]. Note that in our case $L = 9$.

The position observed by the AUV at step q is related to the state by the *measurement equation*, $\mathbf{P_j^q} = \mathbf{H}\mathbf{s_j^q} + T_j'\tilde{\mathbf{C}}\mathbf{v_j^{q-1}}$, where $\mathbf{P_j^q} = [P_j^{q,x}, P_j^{q,y}, P_j^{q,z}]$ represents the *observed position* of the AUV at step q; here, $\mathbf{H} = \begin{bmatrix} \mathbf{I}_3 & 0 & 0 \end{bmatrix}$ is the matrix that extracts the position, whereas $\tilde{\mathbf{C}} = \begin{bmatrix} 0 & \mathbf{I}_3 & 0 \end{bmatrix}^T$ adds the noise. The variable $\mathbf{v_j^{q-1}} = [v_j^{q-1,x}, v_j^{q-1,y},\ v_j^{q-1,z}]^T$ represents the *measurement noise* in *velocity*, expressed as 3D samples of discrete-time white Gaussian noise. Hence, $\mathbf{v_j^q} \sim \mathcal{N}(0, \mathbf{R})$, where $\mathbf{R} \geq 0$ is the covariance matrix of the process. The observed position of the AUV $\mathbf{P_j^q}$ is therefore the actual position of the AUV affected by a measurement noise, which we represent as a Gaussian variable.

The UKF algorithm provides a computationally efficient set of recursive equations to estimate the state of such process, and can be proven to be the optimal filter in the minimum square sense [65]. To implement the UKF algorithm, we need to extend the state vector to the augmented vector $\mathbf{s_j^{q,+}} = [(\mathbf{s_j^q})^T, (\mathbf{w_j^{q-1}})^T, (\mathbf{v_j^{q-1}})^T]^T$ and use the corresponding covariance matrix $\mathbf{V_j^{q,+}} = \mathbb{E}[\mathbf{s_j^{q,+}}(\mathbf{s_j^{q,+}})^T]$. The use of UKF at AUV i reduces the number of necessary location updates. In fact, the filter is used to *estimate the position* at the AUV based on measurements, which is a common practice in robotics, and to *predict* the position of the AUVs thus limitig message exchange (i.e., reducing the need for frequent position updates). The position of j can be estimated and predicted at i based on past estimates $\mathbf{P_j^q}$. To update the state vector, i needs to calculate a so-called $L \times (2L+1)$ sigma point matrix $\boldsymbol{\chi}_{j,q-1}$ with the following column vectors $\boldsymbol{\chi}_{j,q-1}^m$ $(m = 0, \ldots, 2L)$, i.e., for $m = 0$, $\boldsymbol{\chi}_{j,q-1}^0 = \mathbf{s_j^{q,+}}$; for $m = 1, \ldots, L$, $\boldsymbol{\chi}_{j,q-1}^m = \mathbf{s_j^{q,+}} + \left[(L+\lambda)\mathbf{V_j^{q,+}}\right]_m^{1/2}$; for $m = L+1, \ldots, 2L$, $\boldsymbol{\chi}_{j,q-1}^m = \mathbf{s_j^{q,+}} - \left[(L+\lambda)\mathbf{V_j^{q,+}}\right]_m^{1/2}$.

Here $[(L+\lambda)\mathbf{V}_{\mathbf{j}}^{\mathbf{q},+}]_m^{1/2}$ is the m-th column of the matrix square root of $(L+\lambda)\mathbf{V}_{\mathbf{j}}^{\mathbf{q},+}$, where $\lambda = \varsigma^2(L+\kappa) - L$ is a scaling factor depending on ς and κ that controls the spread of the sigma points. These sigma vectors $\boldsymbol{\chi}_{j,q-1}^m$ $(m = 0, \dots, 2L)$ are propagated through the nonlinear state estimate, i.e., (3.16), denoted by $\mathcal{T}$ here, $\mathcal{Y}_{j,q-1}^m = \mathcal{T}(\boldsymbol{\chi}_{j,q-1}^m)$. The state and covariance are predicted by recombining these weighted sigma points, i.e.,

$$\hat{\mathbf{s}}_{\mathbf{j}}^{\mathbf{q}-} = \sum\nolimits_{m=0}^{2L} W_s^m \boldsymbol{\chi}_{j,q-1}^m, \tag{3.17}$$

$$\hat{\mathcal{Y}}_j^{q-1} = \sum\nolimits_{m=0}^{2L} W_s^m \mathcal{Y}_{j,q-1}^m, \tag{3.18}$$

$$\mathbf{V}_{\mathbf{j}}^{\mathbf{q}-} = \sum\nolimits_{m=0}^{2L} W_c^m [\mathcal{Y}_{j,q-1}^m - \hat{\mathbf{s}}_{\mathbf{j}}^{\mathbf{q}}][\mathcal{Y}_{j,q-1}^m - \hat{\mathbf{s}}_{\mathbf{j}}^{\mathbf{q}}]^T, \tag{3.19}$$

where $W_s^0 = \lambda/(L+\lambda)$, $W_c^0 = \lambda/(L+\lambda) - (1 - \varsigma^2 + \beta)$, and $W_s^m = W_c^m = 1/[2(L+\lambda)]$ for $m = 1, \dots, 2L$; β is related to the distribution of $\mathbf{s}$. Normal values are $\varsigma = 10^{-3}$, $\kappa = 1$, and $\beta = 2$. If the distribution of $\mathbf{s}$ is Gaussian, $\beta = 2$ is optimal. Here the superscript $q-$ means that the state $\hat{\mathbf{s}}_{\mathbf{j}}^{\mathbf{q}-}$ or $\mathbf{V}_{\mathbf{j}}^{\mathbf{q}-}$ covariance estimate is *a priori* estimate. Eqs. (3.18) and (3.19) describe how i predicts the state of AUV j before receiving the measurement (*a priori estimate*). Then, i projects the covariance matrix ahead. Once received the measurement $\mathbf{P}_{\mathbf{j}}^{\mathbf{q}}$, i updates the Kalman gain $\mathbf{K}_{\mathbf{j}}^{\mathbf{q}}$ and corrects the state estimate and covariance matrix according to the measurement, i.e.,

$$\mathbf{V}_{j,\tilde{\mathcal{Y}}_q\tilde{\mathcal{Y}}_q} = \sum\nolimits_{m=0}^{2L} W_s^m [\mathcal{Y}_{j,q}^m - \hat{\mathcal{Y}}_{j,q-1}^m][\mathcal{Y}_{j,q}^m - \hat{\mathcal{Y}}_{j,q-1}^m]^T,$$

$$\mathbf{V}_{j,\tilde{\mathbf{s}}_q\tilde{\mathcal{Y}}_q} = \sum\nolimits_{m=0}^{2L} W_s^m [\mathbf{s}_j^{q,m} - \hat{\mathbf{s}}_j^{q-1,m}][\mathcal{Y}_{j,q}^m - \hat{\mathcal{Y}}_{j,q-1}^m]^T,$$

$$\mathbf{K}_{\mathbf{j}}^{\mathbf{q}} = \mathbf{V}_{j,\tilde{\mathbf{s}}_q\tilde{\mathcal{Y}}_q}(\mathbf{V}_{j,\tilde{\mathcal{Y}}_q\tilde{\mathcal{Y}}_q})^{-1}, \tag{3.20}$$

$$\hat{\mathbf{s}}_{\mathbf{j}}^{\mathbf{q}} = \hat{\mathbf{s}}_{\mathbf{j}}^{\mathbf{q}-} + \mathbf{K}_{\mathbf{j}}^{\mathbf{q}}(\mathcal{Y}_j^q - \mathcal{Y}_j^{q-1}), \tag{3.21}$$

$$\mathbf{V}_{\mathbf{j}}^{\mathbf{q}} = \mathbf{V}_{\mathbf{j}}^{\mathbf{q}-} - \mathbf{K}_{\mathbf{j}}^{\mathbf{q}}\mathbf{V}_{j,\tilde{\mathcal{Y}}_q\tilde{\mathcal{Y}}_q}(\mathbf{K}_{\mathbf{j}}^{\mathbf{q}})^T, \tag{3.22}$$

where (3.20) updates the Kalman gain, (3.21) calculates the new state (*a posteriori estimate*), and (3.22) updates the covariance matrix. Note that the complexity of the above computations is the same as the extended KF [64] and that the processing cost at i is much lower than the communication cost.

Let us denote the UKF filtering at time t_q from position $\mathbf{p}$ at time t_{q-1} by $h_{UKF}(t_q, \mathbf{p}, t_{q-1})$, then the predicted external-uncertainty region at step q is $\mathcal{U}_{ij}^q = \{h_{UKF}(qT, \mathbf{p}, (q-1)T) | \mathbf{p} \in$

$\mathcal{U}_{ij}^{q-1}\}$, which, for simplicity, we furher simplify in $\mathcal{U}_{ij}^{q} = h_{UKF}(qT, \mathcal{U}_{ij}^{q-1}, (q-1)T)$.

(2) Distribution Estimation: Let $\mathbf{p} \in \mathcal{U}_{ij}^{q}$, assume $\mathbf{p}$ is predicted from point $\mathbf{p}'$ at step $q-1$, i.e., $\mathbf{p} = h_{UKF}(qT, \mathbf{p}', (q-1)T)$, $\mathbf{p}' \in \mathcal{U}_{ij}^{q-1}$. The pdf $g_{ij}^{q}(\mathbf{p})$ of the external uncertainty $\mathcal{U}_{ij}^{q}$ at step q can be derived from the pdf $g_{ij}^{q-1}(\mathbf{p})$ of $\mathcal{U}_{ij}^{q-1}$ as

$$g_{ij}^{q}(\mathbf{p}) = \int_{\mathbf{p}=h_{UKF}(qT,\mathbf{p}',(q-1)T),\mathbf{p}'\in\mathcal{U}_{ij}^{q-1}} g_{ij}^{q-1}(\mathbf{p}')d\mathbf{p}'.$$

With the help of UKF and the probability theory, we can derive the external uncertainty and its pdf. Note that the initial pdf $g_{ij}^{0}(\mathbf{p})$ is the t-distribution on $\mathcal{U}_{jj}$ (i.e., $\mathcal{U}_{jj}^{0}$) received from j. To reduce the complexity, we convert an uncertainty region (internal or external) into its discrete counterparts, i.e., we divide an uncertainty region into a finite number of equal-size small regions. When the number of small regions is sufficiently large, the UKF filtering on each small region can be approximated by the UKF filtering on a point – e.g., the *centroid* – in this small region. Hence, the predicted external-uncertainty region can be approximated as the region contained in the hull of these predicted points. The pdf functions are also approximated by the probability mass functions on discrete points, which simplifies the pdf estimation after UKF filtering.

Adjustment of the UKF Update Interval: So far, we assumed the update interval T for the UKF algorithm to be *fixed*. A small T determines frequent updates, i.e., the estimation error is corrected in a timely manner; however, frequent external-uncertainty estimations lead to waste of computation resource and energy, causing large network overhead. On the other hand, a large T would save such resources; yet, it may lead to large estimation errors (due to slow update and correction) and thus worse overall performance. To capture this tradeoff, we propose an algorithm to maximize T (i.e., to minimize the update overhead) while keeping the estimation error within an acceptable range: AUV j selects the optimal value T^* such that the prediction errors of all its neighbors (denoted by $\mathcal{N}_j$) be below a specified threshold $e_{\max}$. To do this, j needs to estimate the prediction errors of its neighbors. Say, j estimates the prediction error for i. At each step q, each AUV j emulates the prediction procedure performed at i, calculates its actual new position by filtering the new measurement. Then, it checks to see if the probability of i's prediction error being greater than a maximum error $e_{\max}$ is within a probability threshold, i.e., if $\Pr\{\|\mathbf{P_j^q} - \mathbf{H}\hat{\mathbf{s}}\mathbf{_j^q}\| > e_{\max}\} < \gamma$. To make the formulation clear, we denote $\mathbf{P_j^q}$, $\hat{\mathbf{s}}\mathbf{_j^q}$, and $\mathbf{v_j^q}$ by $\mathbf{P_{ij}^q}$, $\hat{\mathbf{s}}\mathbf{_{ij}^q}$ and $\mathbf{v_{ij}^q}$, respectively. Letting $\mathbf{\Xi_{ij}^q} = \mathbf{P_j^q} - \mathbf{H}\hat{\mathbf{s}}\mathbf{_j^q}$, this condition is actually $\Pr\{\mathbf{\Xi_{ij}^q}(\mathbf{\Xi_{ij}^q})^T > e_{\max}^2\} < \gamma$. From the measurement equation $\mathbf{P^q_{ij}} = \mathbf{H}\mathbf{s^q_{ij}} + T\tilde{\mathbf{C}}\mathbf{v^{q-1}_{ij}}$, assuming $\mathbf{v^{q-1}_{ij}} \sim \mathcal{N}(\mathbf{0}, \xi^2_{ij}\mathbf{I}_3)$, we can see that $\boldsymbol{\Xi}^{\mathbf{q}}_{\mathbf{ij}}(\boldsymbol{\Xi}^{\mathbf{q}}_{\mathbf{ij}})^T/(\xi^2_{ij}T^2)$ has the χ^2-distribution χ^2_3 (note that $\boldsymbol{\Xi}^{\mathbf{q}}_{\mathbf{ij}}$ has 3 elements). From the probability constraint, for each i, the maximal T is $T = \frac{e_{\max}}{\xi_{ij}\sqrt{\hat{\chi}_{\gamma,3}}}$, where $\hat{\chi}_{\gamma,3}$ is the $(1-\gamma)\%$ of χ^2_3-distribution; therefore, $T^* = \min_{i\in\mathcal{N}_j} \frac{e_{\max}}{\xi_{ij}\sqrt{\hat{\chi}_{\gamma,3}}}$.

External-uncertainty Estimation Across Multiple Links: For a multi-hop neighbor AUV j, depending on the selection of the path to j, the estimated uncertainty region may be different. This is because for different paths the j's uncertainty region estimated by intermediate vehicles is generally different. This depends on factors such as availability of ocean-current information, packet loss, communication delays (which introduces asynchronous updates of the external uncertainty for different nodes). Our objective for the multi-hop estimation is to select the estimate that gives minimum uncertainty. To compare the degree of uncertainty, we use *information entropy* as the metric,

$$\mathcal{H}_{\mathcal{U}_{ij}} = -\int_{\mathbf{p}\in\mathcal{U}_{ij}} g_{ij}(\mathbf{p})\log(g_{ij}(\mathbf{p}))d\mathbf{p}; \tag{3.23}$$

here, the bigger $\mathcal{H}_{\mathcal{U}_{ij}}$, the more uncertain $\mathcal{U}_{ij}$. The reason to use this metric instead of simply using the size of the uncertainty region is that the entropy characterizes better the uncertainty. To estimate the j's uncertainty region as it propagates along a path r_{ij}, i estimates the uncertainty region broadcast by $k = prev(i, r_{ij})$, which is i's previous hop along r_{ij}. The estimated uncertainty region $\mathcal{U}_{ij,k\in r_{ij}}$ at time t_{now} is denoted as,

$$\mathcal{U}_{ij,r_{ij}} = h_{UKF}(t_{now}, \mathcal{U}_{kj,prev(k,r_{ij})}, t_{kj,prev(k,r_{ij})}), \tag{3.24}$$

where $\mathcal{U}_{kj,prev(k,r_{ij})}$ is the most recently received estimate of $\mathcal{U}_{kj}$ by k that is sent at time $t_{kj,prev(k,r_{ij})}$. If we denote the set of all paths from i to j as $\mathcal{P}_{ij}$, then the external-uncertainty region of j estimated by i is $\mathcal{U}_{ij} = \arg\min_{\mathcal{U}_{ij,r_{ij}}, r_{ij}\in\mathcal{P}_{ij}} \mathcal{H}_{\mathcal{U}_{ij,r_{ij}}}$. Note that this multi-hop estimation will incur low overhead as, from (3.24), we see that this estimation is performed recursively, i.e., i can use its neighbor's external-uncertainty estimation for multi-hop node j to estimate $\mathcal{U}_{ij}$.

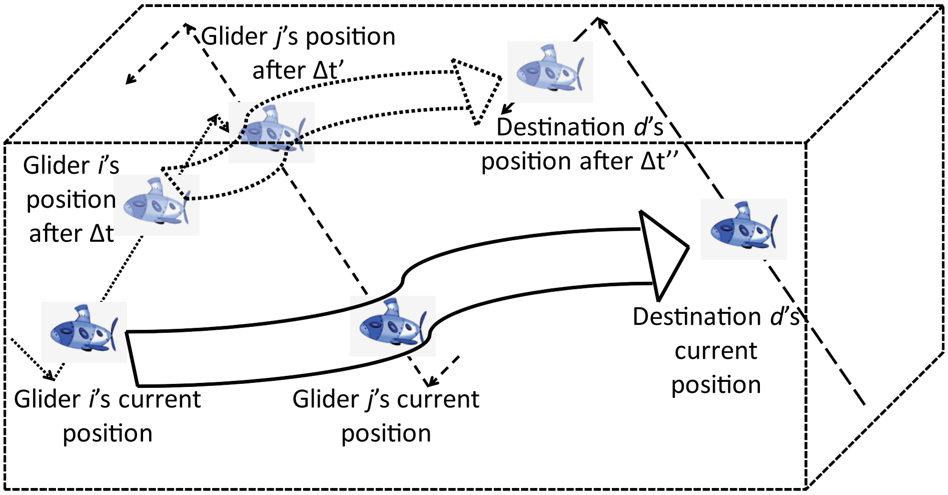

Glider *j*'s position after Δt'
Destination *d*'s position after Δt''
Glider *i*'s position after Δt
Destination *d*'s current position
Glider *i*'s current position
Glider *j*'s current position

Figure 3.8: Glider i delays its transmission by Δt waiting for a better topology so to improve e2e energy and/or throughput to destination d. Wide arrows represent the packet forwarding routes and dashed/dotted simple arrows represent glider trajectories.

3.4 QoS-aware Underwater Optimization Framework for Inter-vehicle Communications

The internal and external uncertainties estimated above can be used to predict the AUV position. The predictability of AUV's trajectory can be further exploited to improve the communication performance among AUVs. In this section, we propose an optimization framework to minimize the communication energy consumption by delaying packet transmissions in order to wait for a favorable network topology, thus trading end-to-end (e2e) delay for energy and/or throughput[3]. For instance, Fig. 3.8 depicts a scenario where glider i waits for a certain time period Δt [s] to save transmission energy and to achieve higher throughput. Based on j's and d's trajectory, glider i predicts a "better" topology with relatively shorter links after Δt and postpones transmission in favor of lower transmission energy and higher data rate. This approach differs from that proposed for Delay Tolerant Networks (DTNs), where delaying transmission becomes necessary to overcome the temporary lack of network connectivity [23, 24].

To estimate an AUV's position, in [66] we proposed a statistical approach to estimate a glider's

[3] Due to the peculiar 'V' shape of the underwater acoustic ambient noise and the high medium power absorption exponentially increasing with distance [4], a shorter distance between AUVs translates into a lower transmission loss and a higher available bandwidth.

trajectory. The estimates were used to minimize e2e energy consumption for networks where packets in the queue need to be forwarded right away (delay-sensitive traffic). In this work, we focus on delay-tolerant traffic and propose an optimization framework that uses acoustic directional transducers to reduce the computation and communication overhead for inter-vehicle data transmission.

Based on the estimated external uncertainty, we propose QUO VADIS, a **Q**oS-aware[4] **u**nderwater **o**ptimization framework for inter-**v**ehicle communication using **a**coustic **di**rectional transducer**s**. QUO VADIS is a cross-layer optimization framework for delay-tolerant UW-ASNs that jointly considers the e2e delay requirements and constraints of underwater acoustic communication modems, including transducer directivity, power control, packet length, modulation, and coding schemes. Specifically, the proposed framework uses the external-uncertainty region estimates of the gliders and forwards delay-tolerant traffic where the maximum e2e delay is large: *Class I (delay-tolerant, loss-tolerant)* and *Class II (delay-tolerant, loss-sensitive)* [10]. Moreover, our cross-layer communication framework exploits the frequency-dependent radiation pattern of underwater acoustic transducers. By decreasing the frequency band, transducers can change their "directivity" turning from being almost omnidirectional (with a gain of ≈ 0 dBi) – which is a desirable feature to support neighbor discovery and multicasting, geocasting, anycasting, and broadcasting) – to directional (with gains up to 10 dBi) – which is useful for long-haul unicast transmissions.

3.4.1 Proposed Optimization Framework

In this part, we first introduce to motivation of adopting a cross-layer optimization design for inter-vehicle communications. We then outline the proposed optimization framework for different classes of traffic and present the detail formulation of the framework. And the solution for this framework is given in the end of this section.

Underwater channel is characterized by high and variable propagation delay, limited bandwidth capacity, frequency dependent attenuation, noise, fading, and Doppler spread. Due to these unique characteristics, we adopt a cross-layer design approach, which has been shown to be necessary [10]. Our proposed inter-vehicle solution is based on the WHOI Micro-Modem. To exploit its functionalities, we first test the Signal-to-Noise Ratio (SNR) performance of the Micro-Modem

[4] The goal of QoS (Quality of Service) is to guarantee a network to deliver predictable results. It involves prioritization of network traffic. Elements of network performance within the scope of QoS often include bandwidth (throughput), latency (delay), and error rate.

Table 3.1: Four Types of Packets used by WHOI Acoustic Micro-Modem (Type 1 and 4 unimplemented yet)

Type	Modulation	Coding Scheme	bps	Max. Frames	Frame Bytes
0	FH-FSK		80	1	32
2	PSK	1/15 spreading	500	3	64
3	PSK	1/7 spreading	1200	2	256
5	PSK	9/17 Rate Block Code	5300	8	256

with our testbed (See 4.1). As Bit Error Rate (BER) can not be directly measured, we use the measured Packet Error Rate (PER) versus SNR figure to characterize the modem's communication performance.

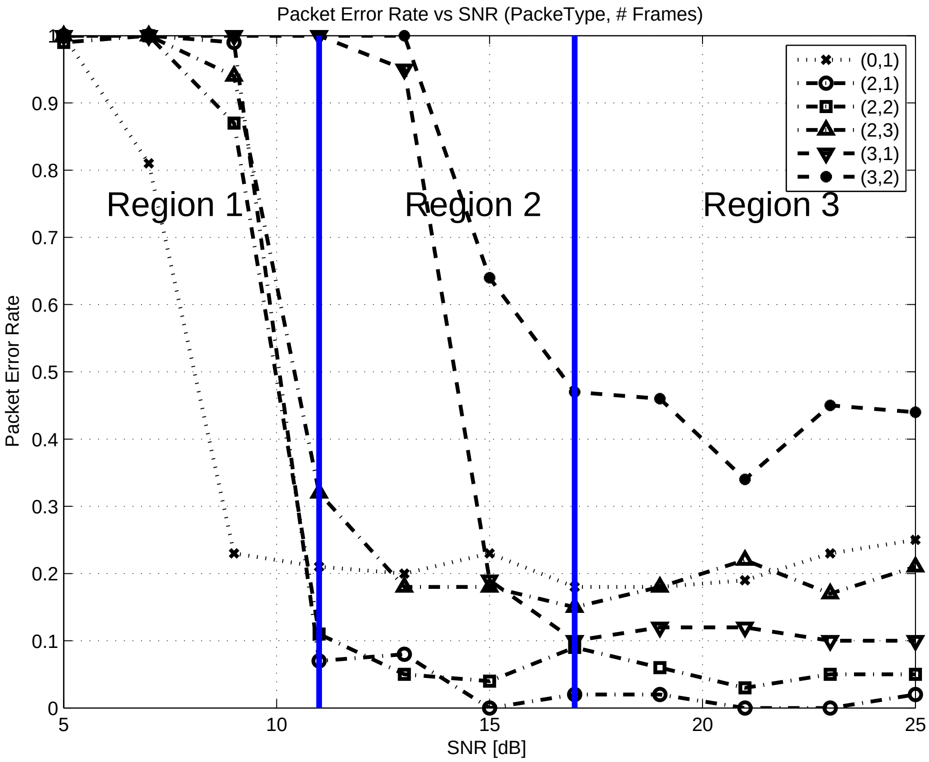


Figure 3.9: Packet Error Rate (PER) for Type 0, 2, 3 packet.

As shown in Table 3.1, there are 4 types of packets using different modulation and coding schemes. Each packet is divided into a number of frames N_F depending on its packet type. The PER of each packet type at each frame length is shown in Figs. 3.9 and 3.10. As we can see in Fig. 3.9, in the low SNR region (Region 1: $<$ 11dB), the PER relationship between different types are: Type 0 $<$ Type 2 $<$ Type 3. Note that: i) as SNR $>$ 11dB (Region 2 and 3), Type 2 packet has lower

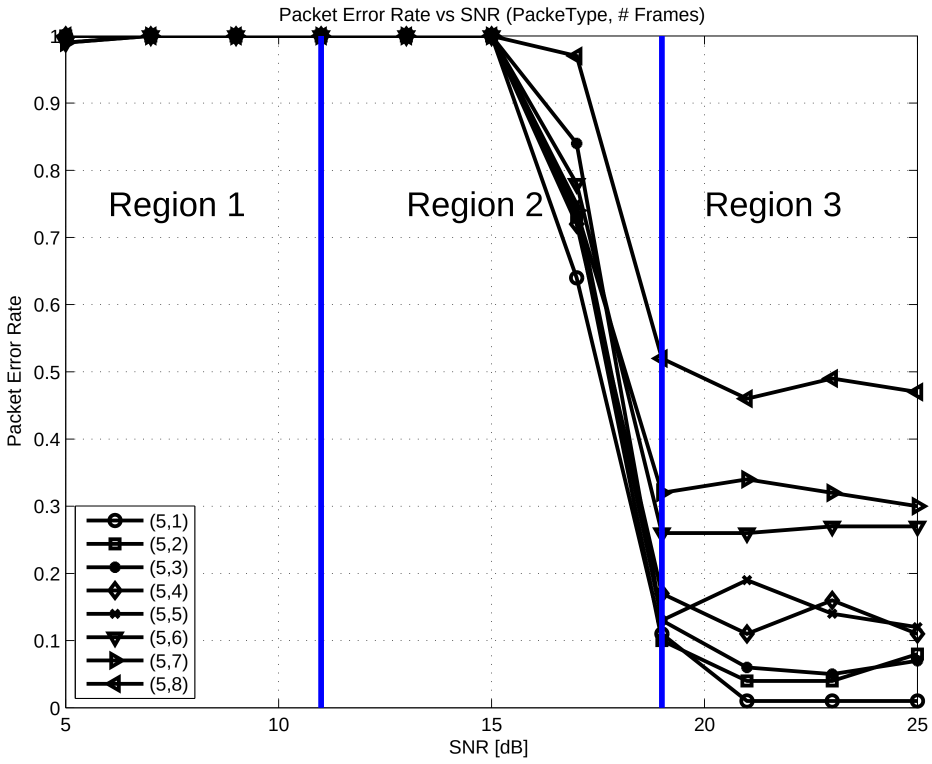


Figure 3.10: Packet Error Rate (PER) for Type 5 packet.

PER than Type 0; Type 2 packet with 3 frames has about the same PER as Type 0, but its bit rate is much higher than Type 0; ii) Type 3 packet with 1 frame has approximately the same PER as Type 0, but the bit rate is much higher. As for Type 5 packet (Fig. 3.10), when SNR $<$ 17dB (Region 1 and 2), its PER is higher than the other packet types. For SNR $>$17dB (Region 3), it has very good PER performance. It has the highest bit rate of 5300 bps.

As shown in Fig. 3.11, as N_F increases, measured througput increases along with the PER. Different types of packets have different PERs for a specific SNR. Therefore the protocol must balance the tradeoff between PER and throughput with a joint consideration of packet type and N_F.

With the external-uncertainty regions in Sect. 3.2, a glider needs to select an appropriate neighbor to forward each packet to its final destination. Because the major part of available energy in battery-powered gliders should be devoted to propulsion [67], acoustic communications should not take a large portion of the available energy. Our proposed protocol minimizes the energy spent to send a message to its destination and considers the functionalities of a real acoustic modem for a practical solution. Specifically, we provide support and differentiated service to delay-tolerant applications with different QoS requirements, from loss sensitive to loss tolerant. Hence, we consider

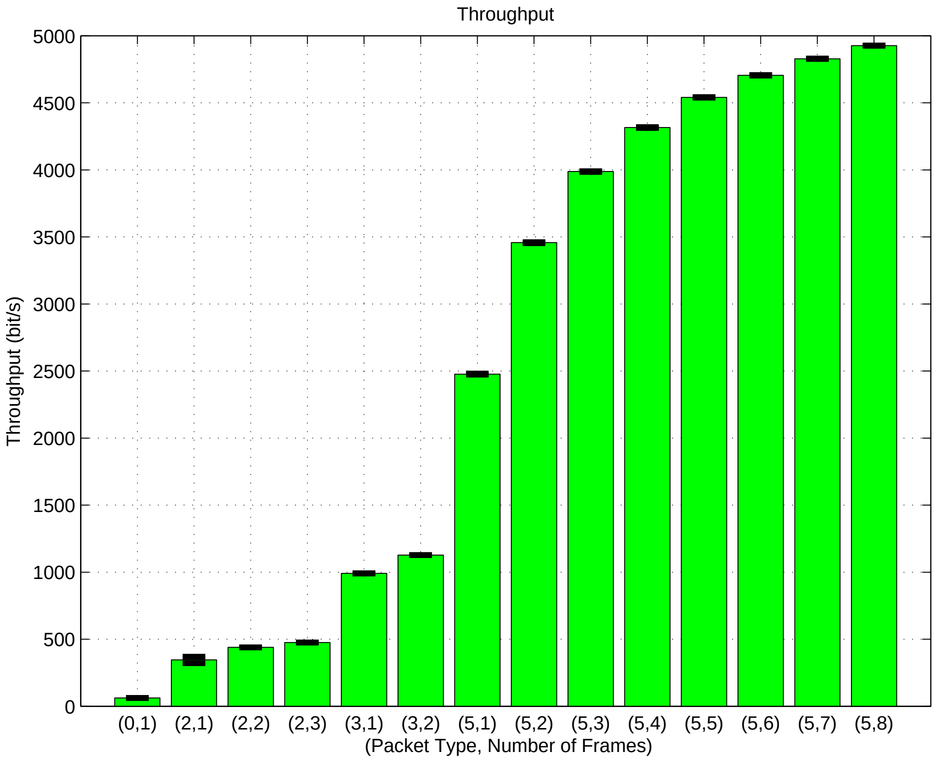


Figure 3.11: Measured throughput.

the following two classes of traffic:

Class I (delay-tolerant, loss-tolerant). It may include multimedia streams that, being intended for storage or subsequent offline processing, do not need to be delivered within strict delay bounds. This class may also include scalar environmental data or non time-critical multimedia content such as snapshots. In this case, the loss of a packet is tolerable at the current hop, but its e2e PER should still be below a specified threshold.

Class II (delay-tolerant, loss-sensitive). It may include data from critical monitoring processes that require some form of offline post processing. In this case, a packet must be re-transmitted if it is not received correctly.

Note that our solution also inherently supports the other two classes of traffic as we can always decrease the maximum e2e delay from the source node to the destination to a small value so that packets are sent right away (or we can modify the remaining time-to-live to 0 in the optimization framework, as seen later in the framework formulation). These two classes of traffic are:

Class III (delay-sensitive, loss-tolerant). It may include video/audio streams as well as meta-data associated with the stream with delay bounds and that are relatively loss tolerant (e.g., video streams can be within a certain level of distortion). This class may also include monitoring data from densely deployed scalar sensors whose monitored phenomenon is characterized by high temporal/spatial correlation, or loss-tolerant snapshots of a phenomenon taken from different viewpoints.

Class IV (delay-sensitive, loss-sensitive). This class may include data from time-critical monitoring processes such as distributed control applications.

In the following two subsections, we first formulate the optimization framework and then the solution for this framework is presented.

3.4.2 Formulation of Optimization Framework

Our protocol employs only local information to make routing decisions, resulting in a scalable distributed solution. The external-uncertainty regions obtained as described in Sect. 3.2 are used to select the neighbor with minimum packet routing energy consumption. Here, a framework using the WHOI Micro-Modem [35] is presented. This framework can be extended and generalized in such a way as to incorporate the constraints of other underwater communication modems.

To be more specific, given the current time t_{now} [s] and a message m generated at time t_0 [s], glider i jointly optimizes the time Δt [s] to wait for the best topology configuration, a neighbor j^*, a frequency band f_{ij}, transmission power $P_{TX}^{(i,j)}(t)$ [W], packet type ξ, and number of frames[5] N_F, so that the estimated energy $E_{id}(t)$ [J] to route m to destined glider d's region $\mathcal{U}_{id}$ is minimized and message m reaches it within $B_{\max}$ [s], the maximum e2e delay from the source to the destination. We assume power control is possible in the range $[P_{min}, P_{max}]$ although transmission power is currently fixed for the WHOI Micro-Modem. We anticipate more advanced amplifier hardware will make this power optimization possible.

Here, $E_{id}(t)$ is estimated by the energy to transmit the packet to neighbor j in one transmission, the average number of transmissions $\hat{N}_{TX}^{(i,j)}(t)$ to send m to j, and the estimated number of hops $\hat{N}_{hop}^{(j,d)}(t)$ to reach region $\mathcal{U}_{id}$ via j. We need to estimate the transmission power and the number of hops to destination. The external-uncertainty region is used to estimate the number of hops $\hat{N}_{hop}^{(j,d)}(t)$ to d via neighbor j and the lower bound of the transmission power as follows (Fig. 3.12).

[5] Each packet sent by WHOI Micro-Modem consists of a number of frames.

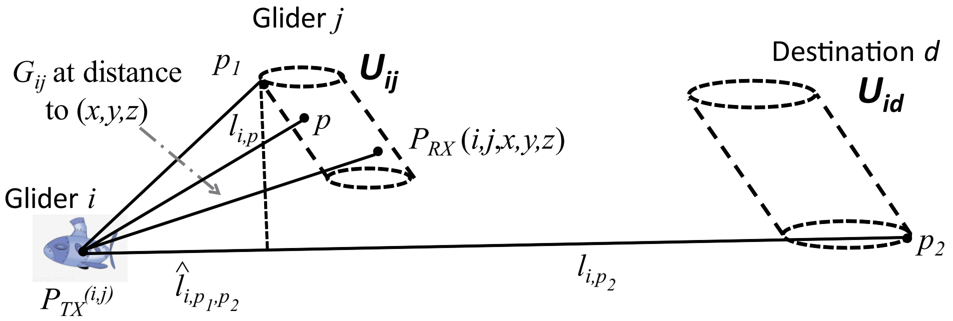


Figure 3.12: Use of external-uncertainty region in the optimization framework.

Let $\hat{l}_{i,p_1,p_2}(t)$ $[\mathrm{m}]$ be the projected distance of line segment from i to position p_1 on the line from i to position p_2, and $l_{i,p}(t)$ be the distance from i to position p. $\hat{N}_{hop}^{(j,d)}(t)$ is estimated by the worst case of $l_{i,p}(t)/\hat{l}_{i,p_1,p_2}(t)$, i.e., (3.29). The lower bound for transmission power is estimated by the average transmission power so that the received power at every point in $\mathcal{U}_{ij}$ is above the specified threshold P_{TH}. The transmission power lower bound is the integral of the product of the transmission power to obtain P_{TH} at a point in $\mathcal{U}_{ij}$ and the probability density function (pdf) of j to be at this point.

To estimate the received power, it is necessary to estimate the transducer gains at the transmitter and receiver. To estimate the transmitter's gain $G_{TX}(\theta_{ij}, \phi_{ij}, f_{ij})$, i needs to compute the radiation angles – the horizontal angle $\theta_{ij} \in [-180°, 180°]$ and the vertical angle $\phi_{ij} \in [-90°, 90°]$ with respect to j. Assume the initial position of the transducer is as shown in the top left corner of Fig. 3.13, then i's normalized transducer direction vector is $\overrightarrow{\mathbf{n}_i} = (0, 0, -1)$ with the *horizontal plane* $z = z_0^{(i)}$ (defined as the plane perpendicular to $\overrightarrow{\mathbf{n}_i}$). While the glider is moving, its pitch, yaw, and roll angles are denoted by ε_i, ζ_i, and η_i, respectively. From geometry, the direction vector after rotation is $\overrightarrow{\mathbf{n}_i'} = \mathbf{Q}_x(\eta_i)\mathbf{Q}_y(\zeta_i)\mathbf{Q}_z(\varepsilon_i)\overrightarrow{\mathbf{n}_i}^T$, while the transducer's horizontal plane is $\mathbf{Q}_x(-\eta_i)\mathbf{Q}_y(-\zeta_i)\mathbf{Q}_z(-\varepsilon_i)[x, y, z]^T = z_0^{(i)}$, where $z_0^{(i)}$ is a constant, and $\mathbf{Q}_x(\eta_i)$, $\mathbf{Q}_y(\zeta_i)$ and $\mathbf{Q}_z(\varepsilon_i)$ are

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\eta_i & -\sin\eta_i \\ 0 & \sin\eta_i & \cos\eta_i \end{bmatrix}, \begin{bmatrix} \cos\zeta_i & 0 & -\sin\zeta_i \\ 0 & 1 & 0 \\ \sin\zeta_i & 0 & \cos\zeta_i \end{bmatrix}, \begin{bmatrix} \cos\varepsilon_i & -\sin\varepsilon_i & 0 \\ \sin\varepsilon_i & \cos\varepsilon_i & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

respectively.

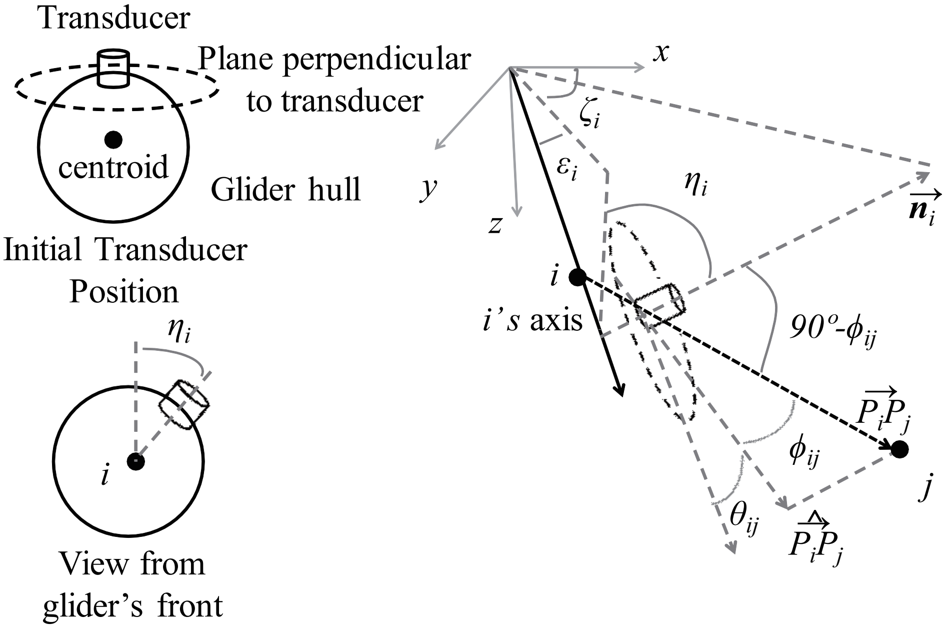


Figure 3.13: Derivation of transducer angles from glider i to j.

With the position vector $\overrightarrow{P_iP_j}$ from i to j, we can derive $\cos\phi_{ij} = \frac{\widehat{\overrightarrow{P_iP_j}} \circ \overrightarrow{P_iP_j}}{\|\widehat{\overrightarrow{P_iP_j}}\| \cdot \|\overrightarrow{P_iP_j}\|}$ and $\cos\theta_{ij} = \frac{\widehat{\overrightarrow{P_iP_j}} \circ \overrightarrow{\mathbf{v}}_i}{\|\widehat{\overrightarrow{P_iP_j}}\| \cdot \|\overrightarrow{\mathbf{v}}_i\|}$, where $\widehat{\overrightarrow{P_iP_j}}$ is the projection of $\overrightarrow{P_iP_j}$ on the transducer's horizontal plane, $\circ$ is the inner product, and $\overrightarrow{\mathbf{v}_i'} = \|\overrightarrow{\mathbf{v}_i'}\| \cdot [\cos\varepsilon_i \cos\zeta_i, \cos\varepsilon_i \sin\zeta_i, \sin\varepsilon_i] = (a_i^*, b_i^*, c_i^*)$ is the velocity vector of glider i as estimated in Sect. 3.2. As $\overrightarrow{\mathbf{n}_i'}$ is perpendicular to the transducer's horizontal plane, we have $\sin\phi_{ij} = \cos(90 - \phi_{ij}) = \frac{\overrightarrow{\mathbf{n}_i'} \circ \overrightarrow{P_iP_j}}{\|\overrightarrow{P_iP_j}\|}$ and $\widehat{\overrightarrow{P_iP_j}} = \overrightarrow{P_iP_j} - (\overrightarrow{P_iP_j} \circ \overrightarrow{\mathbf{n}_i'}) \cdot \overrightarrow{\mathbf{n}_i'}$. The transducer's gain at receiver j, $G_{RX}(\theta_{ji}, \phi_{ji}, f_{ij})$, can be estimated in a similar way.

Let $L_m(\xi)$ be m's length in bits depending on packet type ξ and $B(\xi)$ be the corresponding bit rate. The energy to transmit the packet to neighbor j in one transmission can therefore be approximated by $P_{TX}^{(i,j)}(t) \cdot \frac{L_m(\xi)}{B(\xi)}$.

Overall, the optimization problem can be formulated as

$\mathbf{P(i, d, t_{now}, \Delta t_p)}$: **Cross-layer Optimization Problem**

Given: $P_{min}, P_{max}, \Xi, \Omega_\xi, G_{TX}(), G_{RX}(), \eta, B_{\max}, PER_{\max}^{e2e}$

Computed: $\varepsilon_i, \zeta_i, \varepsilon_j, \zeta_j, \mathcal{U}_{ij}, \forall j \in \mathcal{N}_i \cup \{d\}$ (i.e., $R_j^{(i)}, H_L^{(i,j)}, H_H^{(i,j)}$)

Find: $j^* \in \mathcal{N}_i, P_{TX}^{(i,j)*}(t) \in [P_{min}, P_{max}], \xi^* \in \Xi, N_F^* \in \Omega_\xi, \Delta t^*, f_{ij}^* \in [f_L, f_U]$

$$\textbf{Minimize: } E_{id}(t) = P_{TX}^{(i,j)}(t) \cdot \frac{L_m(\xi)}{B(\xi)} \cdot \hat{N}_{TX}^{(i,j)}(t) \cdot \hat{N}_{hop}^{(j,d)}(t) \qquad (3.25)$$

Subject to:

(class-independent relationships)

$$t = t_{now} + \Delta t; \tag{3.26}$$

$$t_{TTL} = B_{\max} - (t_{now} - t_0); \tag{3.27}$$

$$L_m(\xi) = L_F(\xi) \cdot N_F + L_H; \tag{3.28}$$

$$\hat{N}_{hop}^{(j,d)}(t) = \frac{\max_{p \in \mathcal{U}_{id}} l_{i,p}(t)}{\min_{p_1 \in \mathcal{U}_{ij}, p_2 \in \mathcal{U}_{id}} \hat{l}_{i,p_1,p_2}(t)}; \tag{3.29}$$

$$SINR_{ij}(t) = \frac{P_{TX}^{(i,j)}(t) \cdot 10^{G_{ij}(l_{ij}(t), f_{ij})/10}}{\sum_{k \in \mathcal{A} \setminus \{i\}} P_{TX}^{(k,j)}(t) \cdot 10^{G_{ij}(l_{kj}(t), f_{ij})/10} + N_0}; \tag{3.30}$$

$$G_{ij}(l_{ij}, f_{ij}) = G_{TX}(\theta_{ij}, \phi_{ij}, f_{ij}) + G_{RX}(\theta_{ji}, \phi_{ji}, f_{ij}) - L_{AMP}(f_{ij}) - TL(l_{ij}, f_{ij}); \tag{3.31}$$

$$\theta_{ij} = \arcsin \frac{\overrightarrow{\mathbf{n}_i'} \circ \overrightarrow{P_i P_j}}{\|\overrightarrow{P_i P_j}\|}; \tag{3.32}$$

$$\phi_{ij} = \arccos \frac{\widehat{\overrightarrow{P_i P_j}} \circ \overrightarrow{\mathbf{v}}_i}{\|\widehat{\overrightarrow{P_i P_j}}\| \cdot \|\overrightarrow{\mathbf{v}}_i\|}. \tag{3.33}$$

In this formulation, $\mathcal{N}_i$, Ξ, and Ω_ξ denote the set of i's neighbors, the set of packet types, and the set of number of type ξ frames respectively; $L_F(\xi)$ [bit] is the length of a frame of type ξ, L_H [bit] is the length of message m's header; $PER(SINR_{ij}(t), \xi)$ is the PER of type ξ at the Signal to Interference-plus-Noise Ratio $SINR_{ij}(t)$, $TL(l_{ij}(t), f_{ij})$ is the transmission loss for distance $l_{ij}(t)$ and carrier frequency f_{ij} [kHz] – which is calculated using (1.1) – $\mathcal{A} \setminus \{i\}$ is the set of active transmitters excluding i, and $P_{TX}^{(i,j)}(t)$ is the transmission power used by i to reach j.

Note that $N_0 = \int_{f_L}^{f_U} psd_{N_0}(f, w)df$ is the ambient noise, where $psd_{N_0}(f, w)$ is the empirical noise power spectral density (psd) for frequency band $[f_L, f_U]$ and w [m/s] is the surface wind speed as in [4]. t_{TTL} is the remaining Time-To-Live (TTL) for the packet, $L_{AMP}(f_{ij})$ [dB] is the power loss of the power amplifier at f_{ij} and $PER_{\max}^{e2e}$ is the maximum e2e error rate for packet m.

The objective function (3.25) estimates the energy required to send message m to the destination region $\mathcal{U}_{id}$; (3.26) is the time after waiting Δt; (3.27) calculates the remaining TTL for message m; (3.28) calculates the total message's length; (3.29) estimates the number of hops $\hat{N}_{hop}^{(i,j)}(t)$ to reach destination d; (3.30) estimates the SINR at j while (3.31) estimates the total transmission gain in dB from i to j, including the transducer gain at the transmitter and receiver, loss at the power amplifier, and transmission loss; (3.32) and (3.33) estimate the transducer's radiation angles of j with respect to i. The constraints for $\mathbf{P}(\mathbf{i}, \mathbf{d}, \mathbf{t_{now}}, \mathbf{\Delta t_p})$ are,

(class-independent constraints)

$$P_{TX}^{(i,j)}(t) \geq \int_{(x,y,z) \in \mathcal{U}_{ij}} P_{RX}(i, j, x, y, z) \cdot 10^{-G_{ij}(l_{ij}(t), f_{ij})/10} \cdot pdf_R(x, y) \cdot pdf_H(z) dxdydz; \tag{3.34}$$

$$P_{RX}(i, j, x, y, z) \geq P_{TH}; \tag{3.35}$$

$$0 \leq \Delta t \leq \frac{t_{TTL}}{\hat{N}_{TX}^{(i,j)}(t) \cdot \hat{N}_{hop}^{(j,d)}(t)}. \tag{3.36}$$

In these constraints, $P_{RX}(i, j, x, y, z)$ is the received signal power at the generic 3D location (x, y, z) when i transmits to j. Last, $pdf_R(x, y)$ and $pdf_H(z)$ are the pdfs of the glider's position on the horizontal plane (i.e., χ-distribution with degree of $2N - 2$) and on the vertical direction (i.e., Student's t-distribution with $N - 1$ degrees of freedom), respectively [66], P_{TH} is the received power threshold so that the packet can be received with a certain predefined probability. (3.34) estimates the lower bound of the transmission power to cover the external-uncertainty region so that the received power is above a pre-specified threshold, as accounted for in (3.35); (3.36) estimates the bounds of Δt, which must be less than the maximum tolerable delay at the current hop. To support the two classes of delay-tolerant traffic, we have the following additional constraints,

(additional class-dependent constraints)

$$\textbf{Class I} = \begin{cases} \hat{N}_{TX}^{(i,j)}(t) = 1 \\ 1 - \left[1 - PER(SINR_{ij}(t), \xi)\right]^{\hat{N}_{hop}^{(j,d)}(t)} \leq PER_{\max}^{e2e} \end{cases}$$

$$\textbf{Class II} = \left\{ \hat{N}_{TX}^{(i,j)}(t) = \left[1 - PER(SINR_{ij}(t), \xi)\right]^{-1} \right. .$$

The first constraint for Class I traffic forces packet m to be transmitted only once, while the second constraint guarantees the e2e PER of m should be less than a specified threshold $PER_{\max}^{e2e}$. The constraint for Class II traffic guarantees message m will be transmitted for the average number of times for successful reception at j. By solving this local optimization problem every time the inputs change significantly (and not every time a packet needs to be sent), i is able to select the optimal next hop j^* so that message m is routed (using minimum network energy) to the external-uncertainty region $\mathcal{U}_{id}$ where destination d should be. Obviously different objective functions (e2e delay, delivery ratio, throughput) could be used depending on the traffic class and mission QoS requirements.

3.4.3 Solution to the Optimization Framework

To reduce the complexity, we can convert $\mathbf{P}(\mathbf{i}, \mathbf{d}, \mathbf{t_{now}}, \mathbf{\Delta t_p})$ into a discrete optimization problem by considering finite sets of $P_{TX}^{(i,j)}$ and Δt, which can be taken to be a number of equally spaced values within their respective ranges. The problem then can be solved by comparing the e2e energy

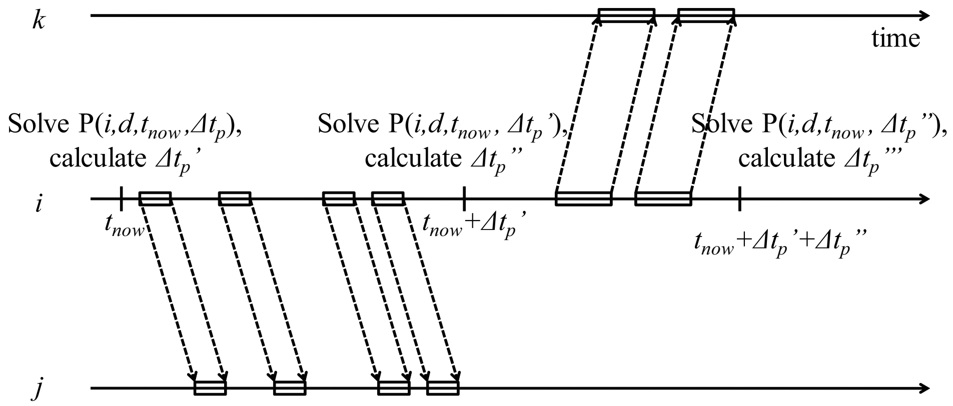

k
time
Solve P(i,d,t_now,Δt_p), calculate Δt_p'
Solve P(i,d,t_now, Δt_p'), calculate Δt_p"
Solve P(i,d,t_now, Δt_p"), calculate Δt_p'"
i
t_now
t_now+Δt_p'
t_now+Δt_p'+Δt_p"
j

Figure 3.14: Solving $\mathbf{P(i, d, t_{now}, \Delta t_p)}$ every Δt_p at i.

consumption estimates of different combination of these discrete values. Assuming that transmission power and time are discretized into N_P and N_{time} values, respectively, for the case of WHOI modem (3 frequencies and 14 combinations of packet type and number of frames [66]), the processor in node i needs to calculate the objective value $42N_P \cdot N_{time} \cdot |\mathcal{N}_i|$ times in each round. The embedded Gumstix motherboard (400 MHz processor and 64 MB RAM) attached to the Micro-Modem is adequate to solve such a problem. To further reduce the computation, instead of running the solution for every packet, it will be rerun only at $t_{now} + \Delta t_p$ for the same class of traffic flow that is sent from i to the same destination d. Here, Δt_p is taken as the minimum of the Δt values of the packets belonging to the same class of traffic and the same destination, estimated from the previous run. Figure 3.14 depicts an example of how $\mathbf{P(i, d, t_{now}, \Delta t_p)}$ is solved at i. At time t_{now}, the problem is solved with j found to be the next hop to d. The minimum of the Δt values of these packets belonging to the same class of traffic and the same destination observed before t_{now} is $\Delta t'_p$. Packets for d will then be forwarded to j with the calculated transmission power at the selected frequency band until $t_{now} + \Delta t'_p$. Then, the problem is solved again and k is found to be the next hop. The minimum Δt observed so far is $\Delta t''_p$ and, hence, the problem will be solved at $t_{now} + \Delta t'_p + \Delta t''_p$. As the Gumstix board in use consumes 2 W of power at maximum and the WHOI Micro-Modem consumes 8-48 W, and the time to solve the optimization problem in this way is generally less than 2 or 3 times of the time taken to transmit a packet. The energy spent in computation is less than that spent in transmission. Therefore it makes sense to optimize the communications using our proposed approach. Moreover, the rapid advance of the processor technology may further reduce the computation energy to solve this optimization problem.

Once the optimal frequency band is selected, i needs to notify j to switch to the selected band. A simple protocol can be used as follows. All AUVs use the same frequency band as the Common Control Channel (CCC) to tell the receiver which band is selected. A short packet or preamble with the selected band number is first sent by the transmitter using the CCC, followed by the data packet using selected frequency band after the time for the transmitter and receiver to finish frequency band switching. The receiver will first listen on the CCC, switch to the selected band embedded in the short control packet or preamble, receive the data packet, and then send back a short ACK packet to acknowledge the reception. Finally, both sides switch back to the CCC if the transmission succeeds or the transmission times out. More sophisticated frequency-band switching protocols, which are out of the scope of this work, can be designed to improve network performance. We rely on the Medium Access Control (MAC) scheme with the WHOI modem to send the data. Since the speed of acoustic wave underwater is very slow when compared with radio waves, the propagation delay has to be considered in order to avoid packet collisions. However, it is difficult to estimate the propagation delay since the positions are uncertain. It may not improve the performance much as the actual propagation delay may be different from the estimation. Moreover, the inter-vehicle traffic underwater is generally low. So the problem of packet collisions is not severe and hence we can just use the MAC scheme provided by the WHOI modem.

3.5 Reliable Geocasting Solution for AUVs

To coordinate the sampling task using multiple AUVs, besides the optimization of the communications among AUVs, there is general a need to send query or commands to vehicles that are located within a certain region. Reliable data delivery to these AUVs is a major concern in applications such as surveillance, data collection, navigation, and ocean monitoring. Geocasting – which is the transmission of data packet(s) to nodes located in a certain geographic region – is becoming a crucial communication primitive. In UW-ASNs, geocasting may be required to assign surveillance tasks to AUVs or to query sensor nodes in a region. It can also be used to notify the nodes within an area of a tactical event (e.g., for detection of enemy vessels). Furthermore, geocasting can be used to facilitate location-based services by announcing a service in a certain region or by sending an emergency warning to a subset of network elements.

Existing geocasting solutions such as [36, 37, 43, 44] are designed for terrestrial wireless networks and hence are not tailed for UW-ASNs that are characterized by large access delay, low bit rate, and high packet loss ratio. Many of these solutions (e.g., [43, 44]) are based on ideal network models derived from graph theory, e.g., the Unit Disk Graph (UDG) model. As a result, these solutions do not perform well in UW-ASNs. Moreover, location uncertainty makes geocasting underwater more difficult compared with that in terrestrial wireless networks. To support geocasting, in fact, location information is required at each node.

Due to these challenges, it is crucial to ensure the communication end-to-end (e2e) reliability between nodes with inaccurate position information. Using e2e error recovery mechanism to ensure the reliability incurs high delay and energy consumption as the recovery is initiated from the end of a route. On the other hand, ensuring link reliability usually leads to reduced delay and energy consumption as a link in error is recovered right away. However, this may not guarantee e2e reliability as a node may deplete its energy, or move and become disconnected from the network. Since e2e error recovery mechanisms generally incur high delay and energy consumption, we choose an approach to guarantee e2e reliability by maximizing link reliability although this may not guarantee e2e reliability (as a node may become disconnected due to energy depletion or movement). Given the 3D geocasting region, under the condition of node position uncertainty, the geocasting protocol needs to: (i) select a path that can forward packets to the maximal number of nodes along the specified direction in a given time, and (ii) maximize the link reliability so that minimal number of retransmissions is required.

We aim at providing a solution to geocast packets to nodes that are located within a directional 3D region. As shown in Fig. 3.15, the geocasting region is a cylinder specified by a tuple $(\mathbf{c}, \overrightarrow{\mathbf{v}}, r)$, where $\mathbf{c} = (x_c, y_c, z_c)$ is the center coordinates, $\overrightarrow{\mathbf{v}} = (v_x, v_y, v_z)$ is the vector specifying geocasting distance and direction, and r is the radius of the region in the plane perpendicular to the specified direction. These seven parameters are the minimum number to characterize a prolonged 3D region. The reason for not assuming a (simpler) spherical region is that the three dimensions of a region in the ocean are generally very different (especially in shallow water). Hence, a sphere would not represent accurately such a region.

We further assume that all the nodes have the same *statistical transmission range* R, which is defined as the average distance to receive a specified percentage of the transmitted packets (e.g., 50%). The case where nodes have different transmission ranges is left as future work.

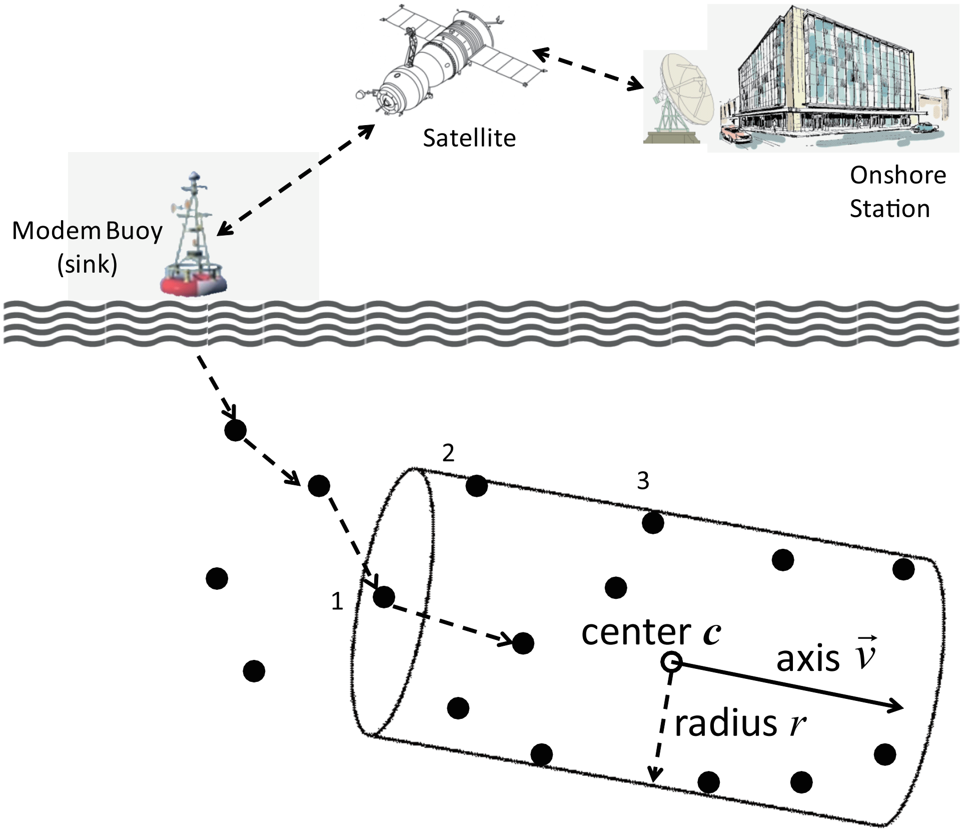
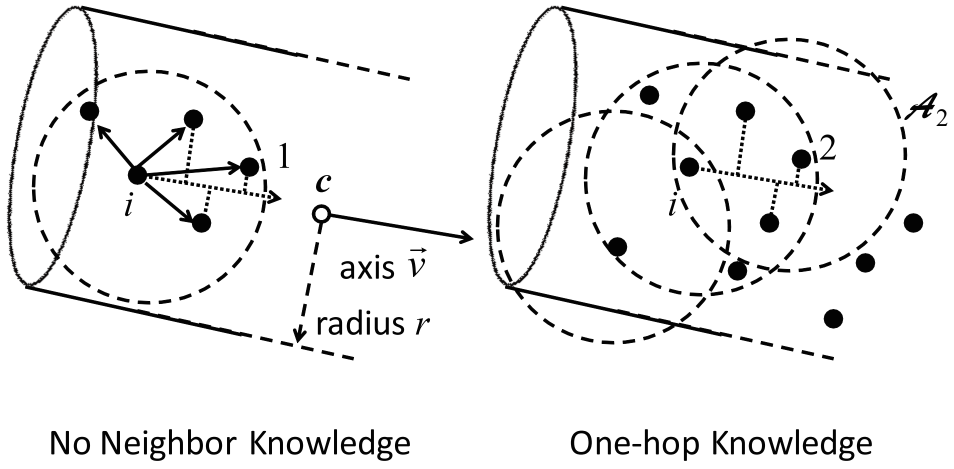


Figure 3.15: Geocasting scenario.

To perform geocasting, a node (such as a sink) issues a geocasting packet, which contains the geocasting region information, i.e., $(\mathbf{c}, \vec{\mathbf{v}}, r)$. If this node is in the geocasting region, the packet will then be forwarded using the geocasting algorithms. Otherwise, the packet will be unicast to a destination node on the boundary of the destination region and then be forwarded using the geocasting algorithms. In the rest of this section, we focus on the problem of geocasting a packet from a node in the geocasting region.

3.5.1 Proposed Geocasting Solution

As shown in Fig. 3.16, based on different degrees of neighbor information, two different versions of the geocasting algorithm are designed for the following cases: 1) *No neighbor knowledge*, which means that each node in the geocasting region has only its own location information (i.e., internal uncertainty) but not those of other nodes; 2) *One-hop neighbor knowledge*, where each node has



No Neighbor Knowledge

One-hop Knowledge

Figure 3.16: Two versions of our proposed solution.

the location information of itself (internal uncertainty) and of its neighbors (external uncertainty). Although different, the overall idea of both versions is to give priority to nodes that are close to the central axis of the region and that are farther along the $\vec{v}$ direction. This is intuitive as generally nodes that are close to the central axis have more neighbors; and by forwarding packets to nodes that are farther along the $\vec{v}$ direction, packets can quickly penetrate the geocast region in this direction around the central axis. To prioritize transmissions, we choose to use timers due to their wide availability on existing underwater modems. Different times are used to hold off the transmissions until the time expires. These times are carefully chosen to avoid packet collisions while trying to maximize the coverage of the transmissions. Moreover, to reduce the number of transmissions, a mechanism is proposed to select a subset of neighbors for packet forwarding.

In case 1, each node estimates its own internal uncertainty and decides when to forward the packet itself. As nodes do not know the external uncertainty of their neighbors, an opportunistic approach is adopted. Furthermore, in order to improve the geocasting reliability, an advertising mechanism is adopted to notify the receiver before the transmission of geocasting packets, i.e., a short packet with higher packet success rate is used to notify the receivers of an incoming packet. In this way, neighbors that did not receive the geocasting packet – but that did receive the short packet – will be able to know that the geocasting packet is lost. An acknowledgement mechanism is devised to allow neighbors of these nodes to forward the geocasting packet to them without the need for retransmissions from the original sender.

In case 2, instead of forwarding packets opportunistically, priority of packet forwarding is decided by the positions of neighbors. A scheduling scheme is designed to prioritize packet transmissions among neighbors. Moreover, a subset of neighbors is selected to maximize the coverage region without introducing packet collisions at the original sender. In case 1, obviously, no overhead is incurred for the exchange of location information. On the other hand, in case 2 (which relies on one-hop neighbor knowledge), nodes need to periodically broadcast information on their uncertainty region. This could be done in different ways, e.g., by periodically embedding this information in the packet that needs to be geocast. In the Sects. 3.5.2 and 3.5.3, we present the details of our solution for both cases. Then in Sect. 3.5.4, we will show that only slight modification is necessary to handle void regions in the area; furthermore, we will analyze the communication overhead associated with both versions and discuss the cases where the no-hop or one-hop version is appropriate in Sect. 3.5.5.

3.5.2 No Neighbor Knowledge

As introduced above, to improve the geocasting reliability for this version, an advertising mechanism is adopted to notify the receiver before the transmission of geocasting packets. This mechanism employs a short packet with higher packet success rate to notify the receivers of an incoming packet. In this way, neighbors that do not receive the geocasting packet but the short packet will be able to know that packets are lost. Upon receiving the packet, the neighbors start their hold-off timers and once this timer expires, the node will broadcast the packet it receives if the channel is idle. An acknowledgement mechanism is further devised to allow neighbors of these nodes to forward the geocasting packet to them without the need to retransmit from the original sender.

To geocast a packet, immediately before broadcasting the packet, i first transmits a short packet, called NOTICE packet, which is sent to cater for the nodes that may have received it but did not receive the geocasting packet. The reason to send the NOTICE packet is that short packets have lower packet error rates than normal geocasting packets. Moreover, this NOTICE packet may be sent using a more reliable modulation and coding scheme. For example, as shown in Figs. 3.9 and 3.10, Packet Error Rates (PERs) of WHOI Micro-Modems for type 0 (using FSK modulation) packet with 32-byte payload is much lower than that of type 5 (using PSK modulation and 9/17 rate block code) packet with 2048-byte payload.

On receiving the geocasting packet for the first time, each neighbor of i, say node j, starts a hold-off timer, T_{hold}. T_{hold} is a uniformly distributed random variable in $[0, 2T_{hold}^{mean}]$, where

$$T_{hold}^{mean} = \left(1 - \frac{d_{ij}^{(\overrightarrow{\mathbf{v}})}}{R}\right)\tau + \frac{d_j}{R}\tau + \frac{\phi_{ij}}{\psi}, \tag{3.37}$$

where $d_{ij}^{(\overrightarrow{\mathbf{v}})}$ is the expected projection distance of the vector $\overrightarrow{\mathbf{p}_i\mathbf{p}_j}$ (position vector from i to j when i, j is at position $\mathbf{p}_i$, $\mathbf{p}_j$, respectively) along the vector $\overrightarrow{\mathbf{v}}$, R is the transmission radius, τ is the estimated transmission time for the current packet, d_j is the expected distance of j to the central vector $\overrightarrow{\mathbf{v}}$, $\psi = 1500$ m/s is the propagation speed of acoustic waves, and $\phi_{ij} = \max\{0, R - \mathbb{E}[\|\overrightarrow{\mathbf{p}_i\mathbf{p}_j}\|]\}$. Here

$$d_{ij}^{(\overrightarrow{\mathbf{v}})} = \int_{\mathbf{p}_j \in \mathcal{U}_{jj}} \left(\overrightarrow{\mathbf{p}_i\mathbf{p}_j} \odot \frac{\overrightarrow{\mathbf{v}}}{\|\overrightarrow{\mathbf{v}}\|}\right) f_j(\mathbf{p}_j) d\mathbf{p}_j, \tag{3.38}$$

$$d_j = \int_{\mathbf{p}_j \in \mathcal{U}_{jj}} \|\overrightarrow{\mathbf{c}\mathbf{p}_j} \otimes \frac{\overrightarrow{\mathbf{v}}}{\|\overrightarrow{\mathbf{v}}\|}\| \cdot f_j(\mathbf{p}_j) d\mathbf{p}_j, \tag{3.39}$$

$$\mathbb{E}[\|\overrightarrow{\mathbf{p}_i\mathbf{p}_j}\|] = \int_{\mathbf{p}_j \in \mathcal{U}_{jj}} \|\overrightarrow{\mathbf{p}_i\mathbf{p}_j}\| f_j(\mathbf{p}_j) d\mathbf{p}_j, \tag{3.40}$$

where $f_j(\mathbf{p}_j)$ is j's pdf at position $\mathbf{p}_j$ in the internal-uncertain region $\mathcal{U}_{jj}$, $\overrightarrow{\mathbf{c}\mathbf{p}_j}$ is the position vector from geocasting region center $\mathbf{c}$ to $\mathbf{p}_j$, and $\odot$ and $\otimes$ are the inner and cross product operator, respectively.

The first and second term in (3.37) give less time to the neighbor that goes farther in the $\overrightarrow{\mathbf{v}}$ direction and that is closer to the central axis respectively, while the third term offsets the propagation delay so that all the nodes receive the packet. This provides fairness by guaranteeing synchronization in starting the hold-off timers of the nodes receiving the data packet. Once the hold-off timer expires, the node broadcasts the packet if the channel is not busy. Otherwise, it just backs off. For the example in Fig. 3.16, on average, node 1 is the first node to forward packets as it has the greatest $d_{ij}^{(\overrightarrow{\mathbf{v}})}$ and smallest d_j.

A node that does not receive the geocasting packet – but that receives the NOTICE packet – will inform the neighboring nodes by sending a NACK packet. Before transmitting a NACK, the node waits for a duration of $T_{NACK-hold-off} = T_{hold} + \frac{R}{\psi} + T_{TX}^{Q}$, where T_{TX}^{Q} is the transmission time of the geocasting packet. This ensures that a node waits long enough to overhear the transmission from a forwarding node in the neighborhood, if any. A node receiving the NACK will respond with probability $\Pr(n)$, where n is the number of NACKs received and $\Pr(n)$ is an increasing function with respect to (w.r.t.) n. A node that receives higher number of NACKs will have a higher probability to respond. If a node does not get the packet during the NACK timeout period, it will retransmit the NACK.

An example is given in Fig. 3.17 to further illustrate the protocol for this no-hop version. The subfigures depict the steps of this protocol. In Fig. 3.17(a), node i broadcasts a NOTICE packet, followed by the geocasting data packet (Fig. 3.17(b)). Upon receiving the packets, neighbor nodes will start their timers according to (3.37). As a result, node 1 will be the first node to do the geocasting. As shown in Fig. Fig. 3.17(e), node 2 will broadcast a NACK message, informing its neighbors that it misses the geocasting packets from node i (Fig. 3.17(b)) and from node 2 (Fig. 3.17(d)). Upon receiving this NACK, node 1 and node 3 schedule a transmission of the data packet with certain probability. Node 1 will then retransmit the data packet as it has higher transmission probability, as shown in Fig. 3.17(e).

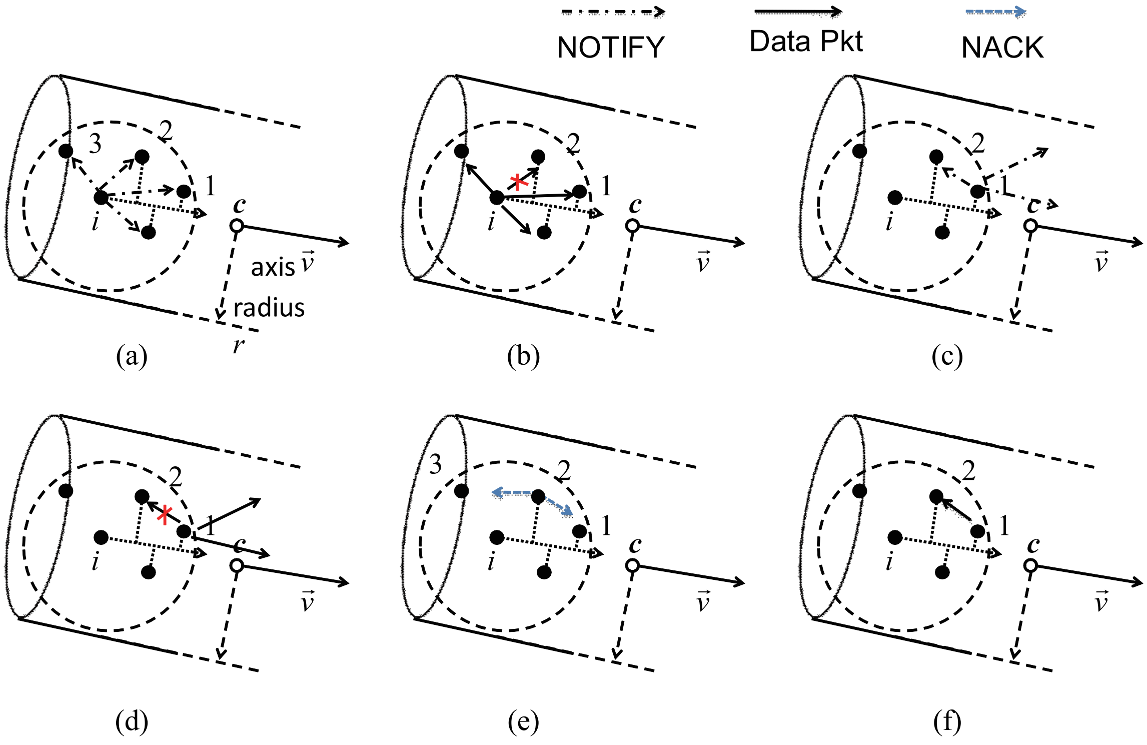


Figure 3.17: Example for no-hop version.

In (3.37), we need to find an appropriate τ to avoid packet collisions. A small τ cannot space out consecutive transmissions to avoid packet collisions. On the other hand, large τ may not only introduce big e2e delay but also impair the priority of transmissions. If the time difference between i's receiving the geocasting packet from j and that from k is greater than T_{TX}^{Q}, collision at i can be avoided. That is, if the probability of reception time (the hold-off time plus the propagation delay) difference being less than T_{TX}^{Q} is kept very low, collisions can be reduced to a great extent at i. Assuming no significant change in ψ spatially, we have the following constraint:

$$\Pr\left(|T_{hold}^{j} + \mathbb{E}[\|\overrightarrow{\mathbf{p}_i\mathbf{p}_j}\|]/\psi - T_{hold}^{k} - \mathbb{E}[\|\overrightarrow{\mathbf{p}_i\mathbf{p}_k}\|]/\psi| \leq T_{TX}^{Q}\right) < \gamma,$$

where T_{hold}^{j} and T_{hold}^{k} are the hold-off times of j and k, respectively, and γ is the threshold collision probability. Since there is no information for neighbors, we assume that j and k are uniformly distributed in the transmission region of i, the pdf function of $\mathbb{E}[\|\overrightarrow{\mathbf{p}_i\mathbf{p}_j}\|]$ is $f_{\mathbb{E}[\|\overrightarrow{\mathbf{p}_i\mathbf{p}_j}\|]}(r) = \frac{2r}{R^2}$ (the same for $\mathbb{E}[\|\overrightarrow{\mathbf{p}_i\mathbf{p}_k}\|]$). Since T_{hold}^{j} and T_{hold}^{k} are uniformly distributed in their respective intervals, let $\Delta T_{j,k}^{i} = \left(T_{hold}^{j} + \mathbb{E}[\|\overrightarrow{\mathbf{p}_i\mathbf{p}_j}\|]/\psi\right) - \left(T_{hold}^{k} + \mathbb{E}[\|\overrightarrow{\mathbf{p}_i\mathbf{p}_k}\|]/\psi\right)$, the pdf of $\Delta T_{j,k}^{i}$ can then be derived as

$$f_{\Delta T_{j,k}^{i}}(s) = \int\int\int f_{\mathbb{E}[\|\overrightarrow{\mathbf{p}_i\mathbf{p}_j}\|]}(r_j) f_{\mathbb{E}[\|\overrightarrow{\mathbf{p}_i\mathbf{p}_k}\|]}(r_k) f_{T_{hold}^{k}}(s_k) \cdot f_{T_{hold}^{j}}(s + s_k + r_k/\psi - r_j/\psi) dr_j dr_k ds_k; \tag{3.41}$$

$$= \int_0^R \int_0^R \int_{s=s_j-s_k+r_j/\psi-r_k/\psi} \frac{1}{2T_{hold}^{k}} \frac{1}{2T_{hold}^{j}} \frac{2r_j}{R^2} \frac{2r_k}{R^2} ds_k dr_j dr_k. \tag{3.42}$$

Therefore $\Pr(|T_{hold}^{j} + \mathbb{E}[\|\overrightarrow{\mathbf{p}_i\mathbf{p}_j}\|]/\psi - T_{hold}^{k} - \mathbb{E}[\|\overrightarrow{\mathbf{p}_i\mathbf{p}_k}\|]/\psi| \leq T_{TX}^{Q}) = \int_{-T_{TX}^{Q}}^{T_{TX}^{Q}} f_{\Delta T_{j,k}^{i}}(s)ds$. Note that $T_{hold}^{j} = (1 - \frac{r_j \cos\theta_j}{R})\tau + \frac{r_j \sin\theta_j}{R}\tau + \frac{R-r_j}{R}$ and $T_{hold}^{k} = (1 - \frac{r_k \cos\theta_k}{R})\tau + \frac{r_k \sin\theta_k}{R}\tau + \frac{R-r_k}{R}$ where θ_j and θ_k are the angles of $\overrightarrow{\mathbf{p}_i\mathbf{p}_j}$ and $\overrightarrow{\mathbf{p}_i\mathbf{p}_k}$ to $\overrightarrow{\mathbf{v}}$ respectively.

The optimal τ is found by solving an optimization problem.

$\mathbf{P}_{\mathbf{desync}}^{\mathbf{nohop}}$: **No Hop Desyncrhonization Optimization Problem**

Given: $R, \gamma, f_j(\mathbf{p}_j), f_k(\mathbf{p}_k)$; **Find:** τ^*; **Minimize:** τ;

Subject to:

$$\Pr\left(|T_{hold}^{j} + \mathbb{E}[\|\overrightarrow{\mathbf{p}_i\mathbf{p}_j}\|]/\psi - T_{hold}^{k} - \mathbb{E}[\|\overrightarrow{\mathbf{p}_i\mathbf{p}_k}\|]/\psi| \leq T_{TX}^{Q}\right) < \gamma, \;\; j \in \mathcal{N}_i.$$

This problem can be solved by find the minimal $\tau^* = \arg\min_{j \in \mathcal{N}_i} \min\{\ \tau \mid \int_{-T_{TX}^{Q}}^{T_{TX}^{Q}} f_{\Delta T_{j,k}^{i}}(s)ds < \gamma\}$ using numerical methods such as the well-known Gauss-Newton algorithm. That is, we can first find the minimal τ for a specific node j and then find the minimal value for all the neighbors.

3.5.3 One-Hop Neighbor Knowledge

For this version, with one-hope information, instead of using the opportunistic forwarding mechanism, we can prioritize the transmissions using an optimal scheduling algorithm. To do this, we can first rank the nodes by jointly considering the relative position along $\overrightarrow{\mathbf{v}}$ and the number of two-hop neighbors. We then divide the neighbors into two sets – the set of forwarding nodes and the set of non-forwarding nodes, which are the nodes that will forward the geocasting packet and the nodes that will not forward the packet – they instead only acknowledge the reception of the packet. To further accelerate the geocasting speed, we partition the set of forwarding nodes into ordered subsets $\mathcal{OS} = \{\mathcal{S}_0, \ldots, \mathcal{S}_M\}$, where nodes in each set $\mathcal{S}_i$ are scheduled to transmit at almost the same time without introducing collisions. Nodes in these sets are further scheduled to either forward the packets or acknowledge the packet reception sequentially.

First of all, transmitter i can prioritize the transmissions of its neighbors by calculating the hold-off timer – similarly to (3.37) – as,

$$T_{hold} = \left(1 - \frac{d_{ij}^{(\overrightarrow{\mathbf{v}})}}{R} + \frac{d_j}{R}\right) \cdot \tau \cdot \frac{1}{\overline{N}_{\mathcal{A}_j}(j)}. \tag{3.43}$$

Here, $\overline{N}_{\mathcal{A}_j}(j)$ represents the expected number of nodes within the 3D region $\mathcal{A}_j$ near j, which is the region inside the sphere of radius R centered at j. That is, $\overline{N}_{\mathcal{A}_j}(j) = \sum_{k\in\mathcal{N}_i} \int_{\mathcal{U}_{ik}\cap\mathcal{A}_j} f_k(\mathbf{p}_k) d\mathbf{p}_k$, where $\mathcal{N}_i$ is the set of i's neighbors. We use external-uncertainty region $\mathcal{U}_{ik}$ to take into account neighbors with predictable trajectories such as underwater gliders [11].

Note that different from (3.37), which is used as timer, (3.43) only serves as a metric to prioritize the transmissions of node, i.e., it is not used in the node as a timer to start transmissions, as shown in the rest of this section. With respect to (3.37), the third term is now removed as the calculation at i does not need to offset for the propagation delay. In addition, $\overline{N}_{\mathcal{A}_j}(j)$, the number of nodes near j, is used as a factor to prioritize transmissions: the more neighbors a node has, the earlier it should transmit in order to reduce the e2e delay. The pdf of $d_{ij}^{(\overrightarrow{\mathbf{v}})}$ is $f_{d_{ij}^{(\overrightarrow{\mathbf{v}})}}(d) = \int_{\|\overrightarrow{\mathbf{p}_i\mathbf{p}_j}\|=d, \mathbf{p}_i\in\mathcal{U}_{ii}, \mathbf{p}_j\in\mathcal{U}_{ij}} f_{\mathcal{U}_{ii}}(\mathbf{p}_i) f_{\mathcal{U}_{ij}}(\mathbf{p}_j)$, where $f_{\mathcal{U}_{ii}}()$ and $f_{\mathcal{U}_{ij}}()$ are the pdfs of i in internal-uncertainty region $\mathcal{U}_{ii}$, and j in $\mathcal{U}_{ij}$, respectively. The pdf of d_j can also be obtained similarly. The node with the smallest T_{hold} is selected as the neighbor with the highest priority and is denoted as j^*.

In addition to giving j^* the highest priority, we want to allow for more simultaneous transmissions so that more area can be covered. Moreover, we can allow transmissions at opposite sides of the region to take place as this can reduce the possibility of collisions. For example, for the scenario shown in Fig. 3.18(a), to geocast the data to the specified region as fast as possible, the packet should be forwarded with high priority by the neighbor that can go far in the desired direction and that has high number of neighbors, i.e., according to (3.43). Moreover, to accelerate packet forwarding, we also design forwarding schemes to allow simultaneous transmissions. For example, as shown in Fig. 3.18(a), node 3 can also forward the packet at almost the same time as node 2. Intuitively, as node 3 is at the opposite side of node 2, forwarding packets at node 3 introduces less number of collisions than the nodes between it and node 2. For example, if node 4 is allowed to transmit at almost the same time as node 2, packet collisions may happen at node 6. To further ensure the least number of collisions, we design an algorithm to pick up the best nodes that can transmit at the same time, as detailed in the rest of this section.

Before going into the details of our algorithm, we illustrate its idea through the example as shown in the subfigures in Fig. 3.18. In subfigure (b), the data packet is first broacast to node i's neighbors. According to (3.43), node 2 will be the node that geocasts first since it has the smallest T_{hold} ($\overline{N}_{\mathcal{A}_j}(j) = 5$ and it is the farthest along $\overrightarrow{\mathbf{v}}$). Now $\mathcal{S}_0 = \{2\}$ and Algorithm 1 is run. According to subfigure (b), node 3 is the only node that does not share with node 2 a common neighbor (except i) within i's neighborhood. The other node, say node 4, has node 6 as common neighbor with node 2. Note that nodes in $\mathcal{S}_0$, i.e., node 2 and 3 ($\mathcal{S}_0 = \{2,\ 3\}$), are located in the opposite directions of i's neighborhood. This will allow geocasting of packets occur in opposite directions. Moreover, intuitionally, the collision probability of the geocasting in these two directions will be small. Therefore the data packet can be forwarded in both directions in minimum time. According the the scheduling rules, node 3 will forward the packet after node 2. Next, node 5 will be selected to be the first node in $\mathcal{S}_1$, and node 4 can also be put in this set. As a result, node 5 will geocast after node 2, followed by node 4. At last, there is no other node that is not covered by the geocasting of these nodes. So all the remaining nodes, (i.e., node 6 and 7), will be selected as non-forwarding nodes. According to the corresponding rules, node 6 will send out the ACK first, followed by node 7.

To achieve this goal, starting from j^*, i partitions its neighbors into sets $\mathcal{S}_m$ ($m = 0, 1, 2, \ldots, M$).

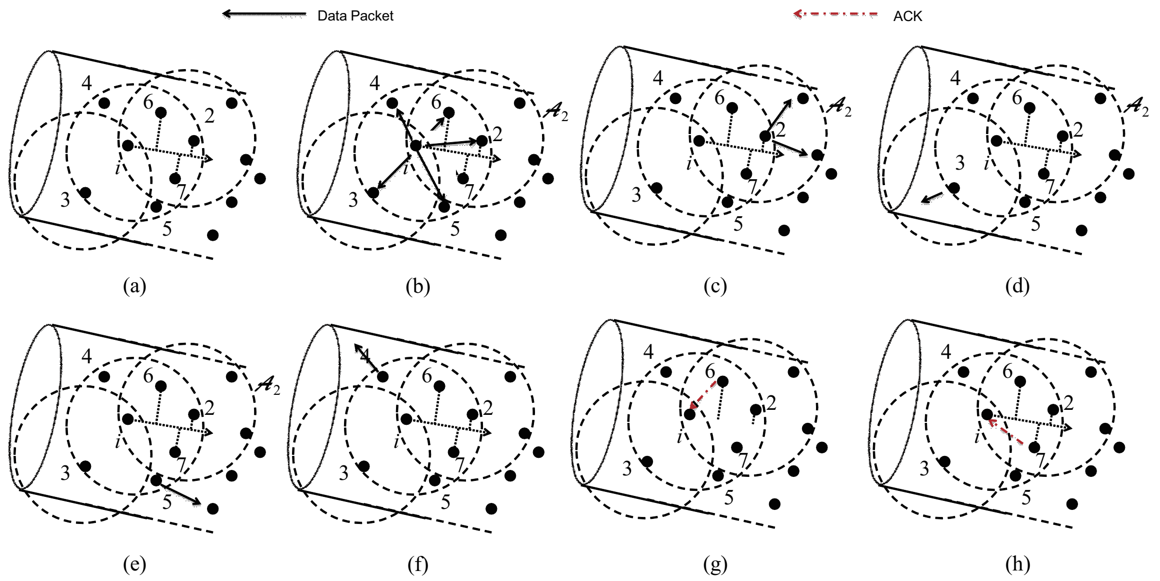


Figure 3.18: Illustration of the one-hop algorithm.

Nodes within $\mathcal{S}_m$ can forward packets without colliding at i's neighbors. Moreover, nodes in $\mathcal{S}_m$ are scheduled to transmit earlier than nodes in $\mathcal{S}_{m+1}$. To avoid collisions, we require that there be no node that is within the statistical transmission range R of nodes in $\mathcal{S}_m$. To calculate these sets, i starts from $\mathcal{S}_0$, which includes j^*, and then calculates $\mathcal{S}_{m+1}$ using $\mathcal{S}_0, \ldots, \mathcal{S}_m$ recursively, as illustrated in Algorithm 1. Node j^* is first put in $\mathcal{S}_0$ and then i searches for a node k in $\mathcal{N}_i$ such that there is no node in the transmission ranges of both k and any node in $\mathcal{S}_0$ (except i itself). The remaining set of nodes not covered by the transmission of nodes in $\mathcal{S}_0$ can be calculated by $\mathcal{N}_{remain} = \mathcal{N}_i - \{k \mid \mathbb{E}[\|\overrightarrow{\mathbf{p}_j\mathbf{p}_k}\|] \leq R, j \in \mathcal{S}_0\}$. Similarly, we can find the set $\mathcal{S}_1 \subset \mathcal{N}_{remain}$ such that nodes in $\mathcal{S}_1$ can transmit at the same time without causing collisions except at i. This process can be repeated to find $\mathcal{S}_M$ ($M \in \mathbb{N}$) such that $\mathcal{S}_{M+1} = \emptyset$ (no further sets can be found). $\mathcal{S}_m$ ($m = 0, 1, 2, \ldots, M$) are put sequentially into an ordered set $\mathcal{OS}$ of sets and the transmissions are scheduled accordingly; nodes in $\mathcal{S}_0$ transmit first, followed by nodes in $\mathcal{S}_1, \mathcal{S}_2, \ldots$.

Nodes in $\mathcal{S}_m$ ($m = 0, 1, 2, \ldots, M$) are set as forwarding nodes that will forward the geocasting packets to their neighbors. As transmitting a packet (short or not) takes a relative long time for existing underwater modems, it is better for i to use the overheard transmissions of the forwarding nodes as acknowledgements for these nodes in order to save time. Hence, it is necessary to avoid the collision at i. So, i needs to schedule the transmissions of these nodes by putting the scheduling

Algorithm 1: Compute Ordered Set $\mathcal{OS}$ using $\mathcal{N}_i$ and $\mathcal{U}_{ij}$'s

```
$\mathcal{N}_{remain} = \mathcal{N}_i$; Calculate $T_{hold}$'s, and $\mathbb{E}[||\overrightarrow{\mathbf{p}_j\mathbf{p}_k}||]$'s;
while $\mathcal{N}_{remain} \neq \emptyset$ do
    $j^* = \arg\min_{j \in \mathcal{N}_{remain}} T_{hold}$; $\mathcal{S} = \{j^*\}$;
    for $k \in \mathcal{N}_{remain} - \mathcal{S}$ do
        $\mathcal{S} = \mathcal{S} \cup \{k\}$ where $k$ have no common neighbor except $i$ with nodes in $S$;
    end
    Add $\mathcal{S}$ to the end of $\mathcal{OS}$ if $\mathcal{S} \neq \emptyset$; break if $\mathcal{S} = \emptyset$;
    $\mathcal{N}_{remain} = \mathcal{N}_{remain} - \{l \mid \mathbb{E}[||\overrightarrow{\mathbf{p}_q\mathbf{p}_l}||] \leq R, q \in \mathcal{S}\}$;
end
```

information in the geocasting packet. On the other hand, nodes in $\mathcal{N}_i - \sum_{m=0}^{M} \mathcal{S}_m$ will be set as non-forwarding nodes, which will only acknowledge the received geocasting packets but not forward. To geocast quickly the packets to the whole region, transmission of these acknowledgement packets is scheduled after the transmission of the forwarding nodes. As collisions may still happen at two-hop neighbors, we randomize the transmissions of the neighbors for collision avoidance.

Scheduling of forwarding nodes: As the transmission time is T_{TX}^{Q}, collision between packets can be avoided if the time difference between reception of two packets at i is greater than T_{TX}^{Q}. Packets will arrive sequentially if the transmission time is delayed for some integer multiple of T_{TX}^{Q}. First, i does not delay the transmission of the node with the highest priority. It then chooses a random permutation of the numbers from 1 to $|\mathcal{S}_m|$ and uses this permutation as the transmission order of the rest of the nodes in $\mathcal{S}_m$ so that their transmissions arrive at i one by one. The timeout for forwarding nodes should be set to $2T_P^{j^*} + |\mathcal{OS}| \cdot T_{TX}^{Q}$, where $T_P^{j^*}$ is the propagation delay required to reach j^* and $|\mathcal{OS}|$ denotes the number of forwarding nodes (i.e., $|\mathcal{OS}| = \sum_{m=0}^{M} |\mathcal{S}_m|$).

Scheduling of non-forwarding nodes: An explicit ACK is sent by a non-forwarding node to the sender after waiting for an ACK-hold-off period, $T_{hold-off}^{ACK}$. To avoid collisions with the geocasting packet, $T_{hold-off}^{ACK}$ should be greater than the timeout for forwarding nodes. We require it be uniformly distributed in $[2T_P^{j^*} + |\mathcal{OS}| \cdot T_{TX}^{Q}, 2T_P^{j^*} + |\mathcal{OS}| \cdot T_{TX}^{Q} + (|\mathcal{N}_i| - |\mathcal{OS}|) \cdot T_{TX}^{ACK}]$. The sender will keep track of all the ACKs and overhearings it receives, and will retransmit the packet if there is even a single neighbor that does not reply implicitly or explicitly. The retransmission timeout is chosen to be $R/\psi + 2T_P^{j^*} + |\mathcal{OS}| \cdot T_{TX}^{Q} + (|\mathcal{N}_i| - |\mathcal{OS}| + 1) \cdot T_{TX}^{ACK}$ so that it is long enough to hear from all its neighbors before it retransmits. Note that R/ψ and the extra T_{TX}^{ACK} are to offset the propagation delay and the transmission delay, respectively.

To de-synchronize the transmissions, an appropriate τ needs to be selected. We can formulate an optimization problem similarly to case 1. However, the pdfs of T^j_{hold} and T^k_{hold} are now derived from $\mathcal{U}_{ij}$ and $\mathcal{U}_{ik}$, respectively. For example, T^j_{hold} is distributed in $[d^{i,j}_{\min}/\psi, d^{i,j}_{\max}/\psi]$ with pdf $f_{T^j_{hold}}(t) = \int_{\|\overrightarrow{\mathbf{p}_i\mathbf{p}_j}\|=\psi t, \mathbf{p}_i\in\mathcal{U}_{ii}, \mathbf{p}_j\in\mathcal{U}_{ij}} f_{\mathcal{U}_{ii}}(\mathbf{p}_i) f_{\mathcal{U}_{ij}}(\mathbf{p}_j)$, where $d^{i,j}_{\min}$ and $d^{i,j}_{\max}$ are the minimal and maximal distances between i (in $\mathcal{U}_{ii}$) and j (in $\mathcal{U}_{ij}$), respectively. Rather than pre-computing τ as in case 1, calculation of τ can now be done *online* so to adjust dynamically as the network topology changes.

3.5.4 Handling Void Region

Our solution ensures all the neighbors will receive the geocasting packet. Therefore, even if there exists void regions, it will not pose a big problem for packet forwarding. Only slight modification needs to be made to the geocasting protocol to avoid replicating the packets. That is, each node keeps track of the packet identifier and it drops the packet that is just received if it has *forwarded* it before. Otherwise, the node will forward the packet to its neighbors. For example, suppose the one-hop version is in use; there is a void region as illustrated in Fig. 3.19; node 6 has geocast a packet to node 1 and node 2; and now node 1 is forwarding this packet along $\overrightarrow{v}$ (it is not aware of the void). Node 2 is selected by node 1 as the first node to continue the forwarding. If node 2 has been selected as a forwarding node during node 6's forwarding, it should have forwarded the packet to node 3 already and hence it will not forward the packet from node 1. Otherwise, if node 2 was a non-forwarding node during node 6's forwarding, it will forward the packet from node 1 so that node 3 receives the packet. Such forwarding will take place until node 4 will forward the packet to node 5. This is similar for the no-hop version.

3.5.5 Application Scenarios of Both Versions

In this part, we analyze the communication overhead of these two versions and discuss the scenarios to apply the no-hop version or one-hop version. The cost for the no-hop version is the communication overhead (NOTICE packet) and the transmission delay incurred. For the one-hop version, the communication overhead is the information about internal uncertainty of itself. This information includes the pdf of a node's internal uncertainty if it follows well-known distribution function, or the probability mass function (pmf) if it does not (where the pdf is properly discretized). This

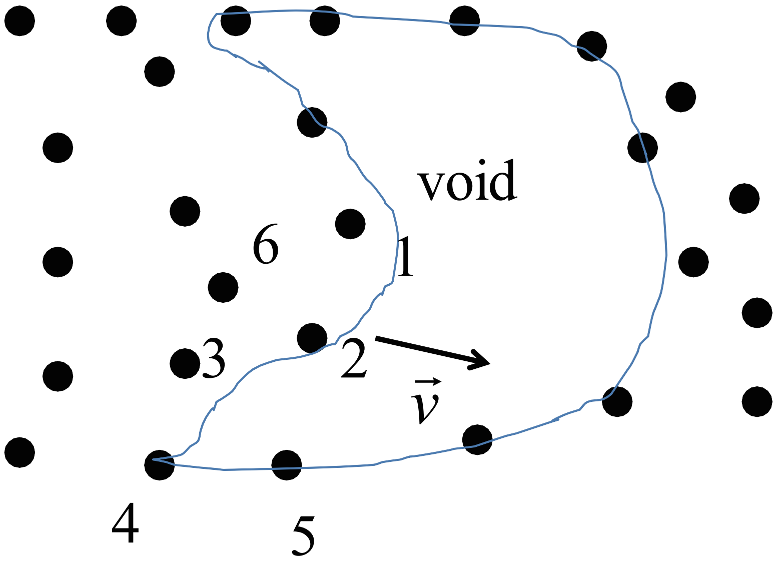


Figure 3.19: Example with a void region.

requires the number of values to determine the well-known distribution, or N_{pmf} values and the associated positions if the internal uncertainty region is discretized into N_{pmf} values (which are $4N_{pmf}$ values). In general, the internal uncertainty region is one connected region, in which case we only need the positions of the boundary points. Suppose the number of boundary points is N_{bd} (clearly $N_{bd} \leq N_{pmf}$), the number of values to be sent is $3N_{bd} + N_{pmf}$. This number can be further reduced if the shape of the internal uncertainty region is a regular 3D shape. For example, if it is a cylinder, we only need the position and radius of the cylinder circular disk center, the cylinder height and the discretization size. Moreover, the associated transmission delay for the one-hop case is larger than the no-hop version. Comparing both versions, obviously the no-hop version has less networking cost in terms of overhead and transmission delay than the one-hop. On the other hand, the one-hop version has better geocasting performance in terms of reliability and delay.

3.6 Under-ice Localization Solution for AUVs

Our model of internal and external uncertainties can be further used to minimize the localization uncertainty underwater. Such localization uncertainty minimization solution can be used in environments where deployment of localization infrastructure is difficult, e.g., in the under-ice environment or open ocean environment. For example, AUVs have been used for continuous measurement of fresh water exiting the Arctic through the Canadian Arctic Archipelago and Davis Strait in order to study the impact of climate change to the circulation of the world's oceans. The ability to minimize localization uncertainty under ice is important so that, for example, scientists can accurately measure how much fresh water flows through the strait – and at what times of year – so they have a baseline for comparison in coming years.

Existing localization schemes underwater generally rely on the deployment of transponders or nodes with underwater communication capabilities as reference points, which requires either much deployment effort or much communication overhead. Moreover, these schemes are not able to estimate the uncertainty associated with the calculated position, which is high in under-ice environments, and thus are not able to minimize position uncertainty.

In this section, we first show how external uncertainty can be used to estimate the uncertainty with the standard distance-based localization technique (i.e., DISLU). Then we propose a novel Doppler-based localization technique DOPLU that jointly estimates localization uncertainty. DISLU requires ranging packets to measure the distances for position calculation, which introduces communication overhead. This weakness in DISLU can be offset by DOPLU, which exploits ongoing inter-vehicle communications to avoid the need for ranging packets. Such an 'opportunistic' approach (i.e., DOPLU) does not guarantee correct absolute locations (as Doppler shifts only characterize *relative* position change) so the team of AUVs needs to go back to DISLU to correct the locations when the error is too large. Based on this idea, we propose algorithms to solve two optimization problems, one for minimization of localization uncertainty and the other for minimization of communication overhead.

The communication protocol for our solution is presented in Fig. 3.20. Each AUV first runs DISLU using the distances measured from the round-trip time. Then, DOPLU is run using Doppler-shift information extracted from inter-vehicle packets. By overhearing the ongoing packets from the reference nodes, AUV i estimates the Doppler shifts and then extracts the relative velocity, from which the AUVs calculate their absolute velocities. DISLU is run to fix the localization error introduced by DOPLU after T_p, which is the time after the last DISLU is started (T_s is the duration for which enough Doppler shifts are collected to estimate the position).

Both DISLU and DOPLU use the external uncertainty and corresponding probability distribution function (pdf) to estimate the uncertainty resulted from the localization technique, i.e., the

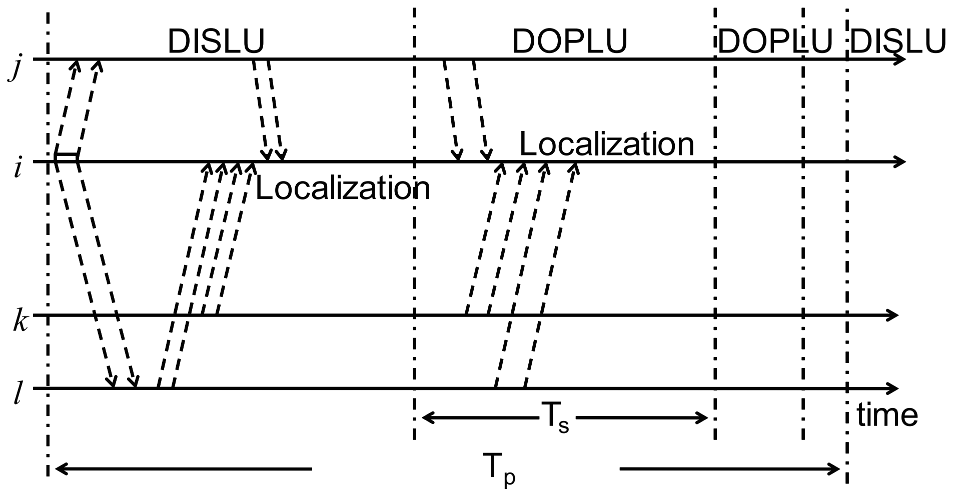


Figure 3.20: Overview of the proposed approach (paired arrows represent the start and the end of one transmission).

internal uncertainty and pdf of the AUV running the localization algorithm. Then this internal uncertainty information is broadcast for other AUVs to estimate external uncertainties.

3.6.1 Distance-based Localization with Uncertainty Estimate (DISLU)

We present here the DISLU technique, which is based on the following idea: to estimate its own position, vehicle i needs 1) to *estimate the distances* between itself and its reference vehicles, and 2) to *estimate its own position* based on these distances.

DISLU relies on the round-trip time T_{RTT} to measure the inter-vehicle distance. By extracting the one-way propagation time, i is able to calculate the inter-vehicle distance. That is, the distance between transmitter i and receiver j is $d_{ij} = c \cdot (T_{RTT} - T_i^{(TX)} - T_j^{(TX)} - T_j^{(hold)})/2$, where $T_i^{(TX)}$ and $T_j^{(TX)}$ are the duration to transmit the packet at i and the duration to transmit acknowledgement at j (i.e., transmission delays), $T_j^{(hold)}$ is the holdoff time of j to avoid collisions. To reduce the transmission time, we can use the short ping packets (e.g., provided by WHOI modem). Once j receives the ping packet, it starts a hold-off timer, $T_j^{(hold)}$, which is a uniformly distributed random variable in $[0, 2T_{hold}^{mean}]$ where T_{hold}^{mean} is given by,

$$T_{hold}^{mean} = (1 - \frac{d_{ij}}{R})\tau + \frac{\phi_{ij}}{c}, \tag{3.44}$$

where d_{ij} is the distance from i to j, τ is the estimated transmission time for the current packet, $c = 1500$ m/s is the propagation speed of acoustic waves, R is the transmission radius of the underwater modem, and $\phi_{ij} = \max\{0, R - d_{ij}\}$. The first term in (3.44) gives less time to the neighbor that is closer to i, and the second term is the extra delay that a node should wait so that all the nodes receive the packet. This gives fairness by providing synchronization in starting the hold-off timers of all the nodes receiving the data packet. $T_j^{(hold)}$ is then embedded in the acknowledge packet for i's information.

After the calculation of d_{ij}'s, i estimates its own position as the point with the least mean squared error to the reference nodes. Then, i estimates its internal uncertainty region using *conditional probability* and the distribution of the reference nodes within their external-uncertainty regions.

Given the set of i's neighbors $\mathcal{N}_i$, the external uncertainty regions $\mathcal{U}_{ij}$, the distances d_{ij}, and the pdf of j within region $\mathcal{U}_{ij}$, $\forall j \in \mathcal{N}_i$, i can estimate the pdf of being at generic point p as

$$\begin{aligned} g(P_i = p) &= \int_{p_j \in \mathcal{U}_{ij}, j \in \mathcal{N}_i} g(P_i = p, \bigcap_{j \in \mathcal{N}_i} P_j = p_j) \\ &= \int_{p_j \in \mathcal{U}_{ij}, j \in \mathcal{N}_i} \big[g(P_i = p | \bigcap_{j \in \mathcal{N}_i} P_j = p_j) \cdot g(\bigcap_{j \in \mathcal{N}_i} P_j = p_j) \big]. \end{aligned} \tag{3.45}$$

Here $g(P_i = p)$ is the pdf of the position of i at point p, $g(|)$ denotes conditional probability function. In our solution, p is calculated as the point that has the minimum squared error, i.e., $p \in S_i$, where $S_i \equiv \{q = \arg\min \sum_{j \in \mathcal{N}_i} \|d(p, p_j) - d_{ij}\|^2\}$ (i.e., $\mathcal{U}_{ii} = S_i$). Here, $d(p, p_j)$ is the distance between point p and p_j. Note that S_i may have more than one element due to the Euclidean norm (e.g., there are two possible positions for the case with three reference nodes and corresponding distances to them known) and p may not be uniformly distributed in S_i if more position constraints are given. Here we assume p to be uniformly distributed in S_i. In other words, we have

$$g(P_i = p | \bigcap_{j \in \mathcal{N}_i} P_j = p_j) = \begin{cases} 1/|S_i|, & p \in S_i \\ 0 \quad\ , & p \notin S_i \end{cases}, \tag{3.46}$$

where $|S_i|$ is the number of elements in S_i if S_i is a discrete set, or the area (or volume) of S_i if S_i is a non-empty non-discrete set (e.g., the case with two references).

The joint pdf, $g(\bigcap_{j \in \mathcal{N}_i} P_j = p_j)$, can be approximated as,

$$g(\bigcap_{j \in \mathcal{N}_i} P_j = p_j) \approx \prod_{j \in \mathcal{N}_j} g(P_j = p_j), \tag{3.47}$$

as the distributions of these AUVs is approximately independent. Since T_p and T_s are generally large (see Sect.3.6.3), the positions of AUVs can be treated as independent after drifting for a long time (while accuracy derivation of the joint pdf is rather difficult). Therefore, (3.45) can be expanded as,

$$g(P_i = p) \approx \int_{p_j \in \mathcal{U}_{ij}, j \in \mathcal{N}_i} \Big[g(P_i = p | \bigcap_{j \in \mathcal{N}_i} P_j = p_j) \cdot \prod_{j \in \mathcal{N}_j} g(P_j = p_j) \Big]. \tag{3.48}$$

Hence i's internal uncertainty $\mathcal{U}_{ii}$ with $g()$ being the pdf is estimated, which is then broadcast to other AUVs. AUVs receiving this information then use $\mathcal{U}_{ii}$ to estimate i's external uncertainty.

3.6.2 Doppler-based Localization with Uncertainty Estimate (DOPLU)

DOPLU runs between two consecutive run of DISLU. Obviously, whenever the Doppler shifts from more than 3 nodes are extracted, DOPLU can be run. The time between two consecutive runs of the DISLU is divided into sub-slots with appropriate duration T_s (Fig. 3.20) so that the DOPLU will be run at an appropriate frequency. Within each sub-slot, the vehicle that runs DOPLU extracts Doppler shifts from the packet it overhears (even if the packet is not intended to be received by it) from the reference vehicles. With the additional information it obtains from the packet header (such as velocity of the reference node), it computes its own absolute velocity, which is then used to estimate its own position and internal uncertainty. This reduces the communication overhead for sending packets to estimate inter-vehicle distance.

An algorithm is designed so that T_s can be adjusted dynamically according to the frequency of ongoing communication activities. Within T_s, a AUV is expected to collect enough Doppler shifts from its reference neighbors so that the DOPLU algorithm runs efficiently. Note that if T_s is too small, it is very likely that the velocity calculated by DOPLU is close to that obtained from the last calculation, which means waste of computation resources. On the other hand, T_s should not be too large as it would lead to too much localization error. After all, the less frequent a AUV calculates its position, the more position error accumulates.

In the rest of this section, we focus on the main problem, i.e., how to estimate the position and internal uncertainty when Doppler shifts are available, and leave the optimization of T_s in Sect. 3.6.3. Using the Doppler shifts regarding to the reference nodes, i can estimate its own absolute velocity using the projected positions (i.e., by adding history position with history velocity times the time passed) and velocities. Using this relationship for all reference nodes, i obtains an equation group to solve, where absolute velocity $\overrightarrow{\mathbf{v}_i}$ can be estimated.

To see how to calculate the absolute velocity, assume that at the end of one sub-slot, AUV i has collected the Doppler shift Δf_{ij} from reference node j. From the definition of Doppler shift, we have $\Delta f_{ij} = -\frac{\overrightarrow{\mathbf{v}_{ij}} \circ \overrightarrow{P_i P_j}}{\|\overrightarrow{P_i P_j}\|} \frac{f_0}{c}$, where $\mathbf{v}_{ij}$ is the relative velocity of i to j, $\overrightarrow{P_i P_j}$ is the position vector from i to j, f_0 is the carrier frequency, $c = 1500 \text{ m/s}$ is the speed of sound, and $\circ$ is the inner product operation. From this equation, we have

$$\frac{\overrightarrow{\mathbf{v}_{ij}} \circ \overrightarrow{P_i P_j}}{\|\overrightarrow{P_i P_j}\|} = -\Delta f_{ij} \frac{c}{f_0}. \tag{3.49}$$

Note that we assume the Doppler shift is estimated accurately. In reality, the frequency-dependent Doppler frequency spread is usually significant due to the inherently wideband nature of the underwater acoustic channel with low Q-factor. Moreover, the temporary variations in factors such as temperature, salinity, depth and ocean surface affect the acoustic speed and propagation path, while drifting due to ocean currents affects the motion of the transmitter and the receiver. All these lead to randomness in the Doppler measurements. Therefore, estimation of Doppler shifts is non-trivial and some solutions such as [68] and [69] have been proposed. To apply DOPLU, special design such as OFDM communication [70] can be applied in physical layer to deal with the generated inter-symbol interference. In this work, we focus on the localization solution itself and assume the Doppler shift reading from acoustic modem - where appropriate Doppler estimation techniques have been applied - is accurate. Consideration of the randomness in Doppler reading in DOPLU is left as future work.

From (3.49), assume that i has collected the Doppler shifts of $N_i^{(ref)}$ reference nodes, we then have an equation group with $N_i^{(ref)}$ equations. We then can derive i's velocity $\overrightarrow{\mathbf{v}_i}$. Assume $\overrightarrow{\mathbf{v}_i} = (v_x^{(i)}, v_y^{(i)}, v_z^{(i)})$ and $\frac{\overrightarrow{P_i P_j}}{\|\overrightarrow{P_i P_j}\|} = (\alpha_x^{(ij)}, \alpha_y^{(ij)}, \alpha_z^{(ij)})$, (3.49) is then $\overrightarrow{\mathbf{v}_{ij}} \circ \frac{\overrightarrow{P_i P_j}}{\|\overrightarrow{P_i P_j}\|} = (\overrightarrow{\mathbf{v}_i} - \overrightarrow{\mathbf{v}_j}) \circ \frac{\overrightarrow{P_i P_j}}{\|\overrightarrow{P_i P_j}\|} = (v_x^{(i)} - v_x^{(j)})\alpha_x^{(ij)} + (v_y^{(i)} - v_y^{(j)})\alpha_y^{(ij)} + (v_z^{(i)} - v_z^{(j)})\alpha_z^{(ij)} = -\Delta f_{ij} \frac{c}{f_0}$. By manipulating this equation, we have

$$v_x^{(i)}\alpha_x^{(ij)} + v_y^{(i)}\alpha_y^{(ij)} = -\Delta f_{ij}\frac{c}{f_0} - v_z^{(i)}\alpha_z^{(ij)} + v_x^{(j)}\alpha_x^{(ij)} + v_y^{(j)}\alpha_y^{(ij)} + v_z^{(j)}\alpha_z^{(ij)}. \tag{3.50}$$

In this equation, $v_x^{(i)}$ and $v_y^{(i)}$ in the left-hand side are variables to be solved, whereas $v_z^{(i)}$ in the right-hand side can be derived from pressure sensor reading, $(\alpha_x^{(ij)}, \alpha_y^{(ij)}, \alpha_z^{(ij)})$ is the normalized vector of $\overrightarrow{P_i P_j}$, and $(v_x^{(j)}, v_y^{(j)}, v_z^{(j)})$ is obtained from the velocity information embedded in the overheard packet header of j.

Considering all the $N_i^{(ref)}$ reference nodes, we can obtain a linear equation group, which can be expressed in a matrix form as $\mathbf{Ax} = \mathbf{b}$, where

$$\mathbf{A} = \begin{bmatrix} \alpha_x^{i1} & \alpha_y^{i1} \\ \alpha_x^{i2} & \alpha_y^{i2} \\ \cdot & \cdot \\ \alpha_x^{iN_i^{(ref)}} & \alpha_y^{iN_i^{(ref)}} \end{bmatrix}, \mathbf{x} = \begin{bmatrix} v_x \\ v_y \end{bmatrix}, \mathbf{b} = \begin{bmatrix} b_{i1} \\ b_{i2} \\ \cdot \\ b_{iN_i^{(ref)}} \end{bmatrix}. \tag{3.51}$$

Here $b_{ij} = -\Delta f_{ij}\frac{c}{f_0} - v_z^{(i)}\alpha_z^{(ij)} + v_x^{(j)}\alpha_x^{(ij)} + v_y^{(j)}\alpha_y^{(ij)} + v_z^{(j)}\alpha_z^{(ij)}$. We want to find the optimal $\mathbf{x}^*$ such that the sum of squared errors is minimized. That is,

$$\mathbf{x}^* = \arg\min \|\mathbf{b} - \mathbf{Ax}\|^2. \tag{3.52}$$

From *matrix theory*, $\mathbf{x}^*$ can be solved as $\mathbf{x}^* = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}$. Once the velocity is calculated, the position of i is updated as $p_i = p_i' + \overrightarrow{\mathbf{v}} \cdot T_s$, where $\overrightarrow{\mathbf{v}} = [v_x^{(i)}, v_y^{(i)}, v_z^{(i)}]^T$.

Assume that the uncertainty regions $\mathcal{U}_{ij}$ and the distribution pdf of j within region $\mathcal{U}_{ij}$ are known (by embedding these parameters in the header of the packet), $\forall j \in \mathcal{N}_i$, i can estimate the pdf of being at point p as $g(P_i = p) \approx \int_{p_j\in\mathcal{U}_{ij},\forall j\in\mathcal{N}_i} \left[g(P_i = p|\bigcap_{j\in\mathcal{N}_i} P_j = p_j) \cdot g(\bigcap_{j\in\mathcal{N}_i} P_j = p_j)\right]$. Similar to the case of DISLU, i can calculate the distribution of its own location and, hence, its internal uncertainty region.

Minimization of Location Uncertainty: Obviously, localization using different references leads to different estimation of internal uncertainty and corresponding pdf. Our objective is to minimize the estimated internal uncertainty. Using our notions of internal and external uncertainty, this can be achieved by solving an optimization problem. To measure the degree of uncertainty, we use *information entropy* as the metric, i.e.,

$$H(\mathcal{U}_{ij}, g_{ij}) = -\int_{p\in\mathcal{U}_{ij}} g_{ij}(p)\log(g_{ij}(p))dp. \tag{3.53}$$

The bigger $H(\mathcal{U}_{ij}, g_{ij})$ is, the more uncertain $\mathcal{U}_{ij}$ is. The reason to use information entropy instead of simply the size of uncertainty region is that it can better characterize uncertainty. **Example:** Assume that an AUV's position is distributed (in 1D) in [0,10] along x-axis with pdf being 9.9 in [0,0.1] and 0.1/99 in [0.1, 10] (Case 1). Then its entropy is -3.17 bits, which is less than the entropy 3.32 bits when the AUV is uniformly distributed in [0,10] (Case 2) or the entropy 3 bits when the AUV is uniformly distributed in [0,8] (Case 3). Obviously Case 1 is the most certain in these 3 cases even though Case 2 has the same size and Case 3 has the smallest size of the region. Note that the information flow between AUVs can occur in loops; this may not amplify errors of the positioning algorithm, as our problem selects the neighbors that can minimize the uncertainty.

With this metric, the problem to minimize localization uncertainty can be formulated as,

$$\begin{aligned}
\textbf{Given: } & \mathcal{N}_i, \mathcal{U}_{ij}, g_{ij}(); \\
\textbf{Find: } & \mathcal{A}_i^*; \quad \textbf{Minimize: } H(\mathcal{U}_{ii}, g_{ii}); \\
\textbf{S.t.: } & \mathcal{U}_{ii} \equiv \{q = \arg\min \sum_{j \in \mathcal{A}_i} \|d(p, p_j) - d_{ij}\|^2\}; && (3.54) \\
& g(P_i = p) = \int_{p_j \in \mathcal{U}_{ij}, j \in \mathcal{A}_i} \Big[g(P_i = p | \bigcap_{j \in \mathcal{N}_i} P_j = p_j) \cdot \prod_{j \in \mathcal{A}_i} g(P_j = p_j) \Big]; && (3.55) \\
& |\mathcal{A}_i| \geq 3; \quad \mathcal{A}_i \subset \mathcal{N}_i. && (3.56)
\end{aligned}$$

Here $\mathcal{A}_i$ represents a subset of i's reference nodes, (3.54) and (3.55) estimate the internal uncertainty and corresponding pdf when nodes in $\mathcal{A}_i$ are used as references; and (3.56) are the constraints for $\mathcal{A}_i$ so that enough reference nodes are selected for localization.

To reduce the complexity, we can convert an uncertainty region (internal or external) into discrete counterparts. That is, we divide an uncertainty region into a finite number of equal-size small regions. When the number K_i of small regions is sufficiently large, the pdf of the AUV's position on a point – such as the centroid – in this small region can therefore be approximated by the probability on a small region. Hence the estimated external-uncertainty region can be approximated as the region contained in the hull of these estimated points. The pdf functions are also be approximated by the probability mass functions on discrete points, which simplifies the pdf estimation. The above optimization can then be solved using exhaustive search algorithm after the discretization. The computation complexity of the exhaustive search is $\mathcal{O}(2^{|\mathcal{A}_i|} K_i^{|\mathcal{A}_i|})$. Since the number of AUVs is generally small, this complexity is mainly decided by K_i. Depending on the computation capability of the onboard processor, appropriate K_i can be used. Further improvement of the solution can be done after converting it into appropriate optimization that can be solved efficiently and we leave this as future work.

3.6.3 Minimization of Communication Overhead

In this part, we discuss how to optimize T_s and T_p so that localization overhead can be minimized while keeping the localization uncertainty low. We first propose an algorithm to dynamically adjust T_s in order to maintain the performance of DOPLU. Then, T_p is optimized to minimize the localization overhead.

As for T_s, it should be large enough so that packets from enough reference nodes are overheard. Suppose $K_{\min}$ is the minimum number of reference nodes (or $|\mathcal{A}_i|$ if the optimization algorithm in Sect. 3.6.2 is used) so that $\mathbf{x}^*$ can be calculated using DOPLU. In the beginning, T_s is initialized as $T_s = \frac{R}{c} + T_{TX} \cdot K_{\min}$, i.e., the minimum time to overhear packets from $K_{\min}$ reference nodes. Suppose that during the last T_s' period, Doppler shifts from N' reference nodes with smaller degree of uncertainty (seen by i) than i's are received. On average, it takes T_s'/N' to receive a useful Doppler shift. Then, the expected time to receive $K_{\min}$ useful Doppler shifts is $T_s' \cdot K_{\min}/N'$. We update T_s using a weighted average. That is, $T_s = \beta \cdot T_s' + (1-\beta) \cdot T_s' \cdot K_{\min}/N'$, where $\beta \in (0,1)$ is a weight factor.

Using internal and external uncertainty, we can also optimize the interval T_p running DISLU. By optimizing T_p, we minimize the overhead to use DISLU and hence the overall overhead. DISLU is run when the localization error is large. The localization error can be estimated by calculating the distance from the position estimated by DISLU to that estimated by DOPLU. When the localization error is greater than a threshold d_{th}, DISLU is run to correct the error. Since the position is not deterministic, this requirement is expressed in a probabilistic way. That is, DISLU should be run when the probability of the localization error being over d_{th} is above a threshold probability γ. Therefore, to minimize the overhead of running DISLU, T_p should be maximal under the constraint that the probability of the localization error being over d_{th} is below γ. This can be formulated into the following optimization problem,

$$\begin{aligned} &\textbf{Given: } \mathcal{U}_{ij}, g_{ij}(), \gamma; \\ &\textbf{Find: } T_p^*; \quad \textbf{Maximize: } T_p; \\ &\textbf{S.t.: } \Pr\{\|\overrightarrow{p_i(T_p)\tilde{p}_i(T_p)}\| > d_{th}\} < \gamma, \end{aligned}$$

where $p_i(T_p)$ and $\tilde{p}_i(T_p)$ are the predicted positions using the DOPLU and DISLU after T_p from the last DISLU run time, respectively. This prediction of future internal uncertainty is based on the current estimated internal uncertainty and AUV's trajectory, as in Sect. 3.3. As the previous optimization problem, we can also convert it into discrete variable optimization problem and solve it in a similar way. Depending on the prediction method and the type of AUVs, the computation complexity varies. For example, using the prediction method in Sect. 3.2, the computation complexity is $\mathcal{O}(K_i N_{smp})$ for underwater gliders with N_{smp} of position samples. Note that T_s and T_p can be jointly optimized, which is more complicated and hence is left as future work.

3.7 Team Formation and Steering Algorithms

Based on our proposed communication solutions, we further proposed a solution for the coordination of a team of underwater gliders. Our solution is based on the functionalities of WHOI acoustic Micro-Modem. We focus on how to form the team according to the given formation geometry when randomly scattered gliders are selected and on how to steer them along the trajectory while maintaining their formation. We assume that the gliders in the team have been selected from a pool of vehicles using a task allocation algorithm (e.g., [57]). As in Fig. 3.21, given i) the number of scattered gliders, ii) the corresponding geometry formation, and iii) the target trajectory, two phases of operations are required to perform the monitoring mission: 1) the selected gliders need to be mapped into a specified geometric formation making sure that no collisions occur (Phase I); 2) after the first phase, the team needs to steer through the 3D region of interest along the predefined trajectory while maintaining its formation (Phase II). Note that swarming using a specified geometric formation is necessary not only in coordinated monitoring missions but also in many applications such as surveillance/tracking and collision avoidance in critical navigation missions.

In our solution, a glider is selected to play the role of *leader* in order to guide the other gliders, which will then act as *followers*. These are *logical roles* that do not depend on the physical position within the formation, i.e., the leader is not necessary ahead of the followers at all times. As GPS does not work underwater, gliders can only receive GPS signals when at the surface; therefore, to calculate their positions while underwater they can only rely on localization algorithms. Moreover, accuracy of the location information decreases as the time in the water increases due to the accumulation of localization errors. Consequently, in order to take advantage of the GPS information, the last surfaced glider is chosen to be the *leader*. The aim of the leader is to let the team be on track along the target trajectory, while the aim of the other gliders, the so-called *followers*, is to maintain

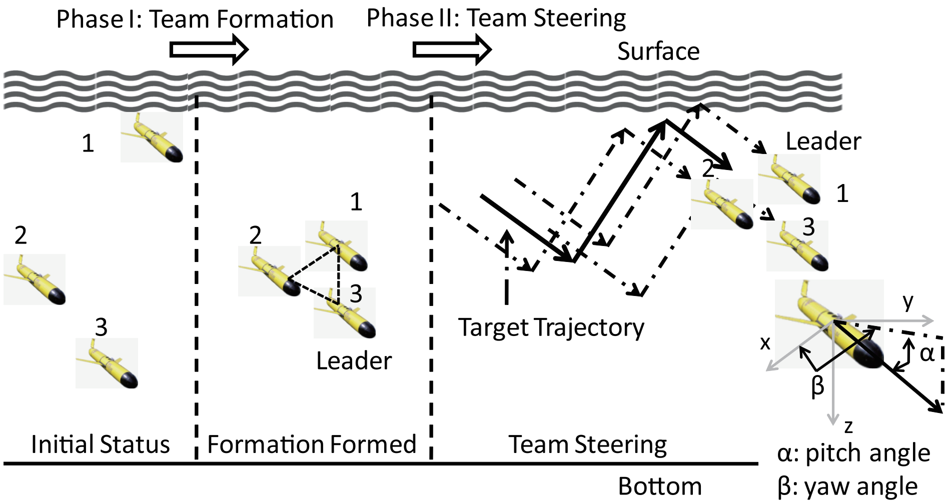


Figure 3.21: Overview of the proposed solution for team formation and steering.

the formation according to the predefined geometry. When surfaced, a glider advertises itself as the *'new'* leader by broadcasting a message. Upon receiving this message, the *'old'* leader sends a confirmation to the new leader using an acknowledgement packet (ACK), whereas the other gliders just update their leader information without sending any ACK to avoid message implosion. The ACK packet from the 'old' leader serves also as communication redundancy to echo the message from the new leader; by leveraging spatial diversity, the probability of reaching all the gliders is thus increased.

Our solution is based on the SLOCUM glider. SLOCUM glider control involves monitoring navigation performance, adjusting glide angle by controlling pitch and/or buoyancy, and adjusting heading by controlling roll or rudder position. The gliders can use Precision Navigation TCM2 attitude sensors to sense heading, pitch and roll, and pressure sensors to measure depth and, from pressure rate, vertical velocity. Altitude is measured using an acoustic altimeter. A movable rudder gives the tightest turning radius (approximately 7 m) and allows turning without significant roll so that the acoustic altimeter, critical in shallow-water operations, remains accurate.

Our solution controls the *pitch angle* α and *yaw angle* β (see Fig. 3.21) to steer each glider and keep the team formation. The pitch α for a glider ranges in $[\alpha_{min}, \alpha_{max}]$ and determines the velocity of the vehicle (in fact, the horizontal velocity can be considered constant in the absence of ocean currents).

3.7.1 Team Formation

To enable the communications between the scattered gliders, we propose a communication technique that emulates the vocalizations used by killer whales. These whales use low frequency whistles ranging from 0.5 to 40 kHz (with peak energy in $6 - 12$ kHz) to communicate with each other. These low frequencies make long-range communication possible, as explained by the underwater communication theory: low-frequency tones undergo a lower medium attenuation and achieve a higher SNR [4] at the receiver. Moreover, the whistles are usually short, which is advantageous as they are less affected by multipath. This effect is similar to what happens in wireless communications: shorter packets experience a lower Packet Error Rate (PER).

We incorporated these characteristics in our acoustic communication framework, to understand what we show in Figs. 3.9 and 3.10 the WHOI Micro-Modem's PER performance as measured in our testbed emulation [66]. As presented in Table 3.1, there are four types of packets used by these underwater acoustic modems, each adopting a different combination of modulation and coding scheme, and a specific number of frames. By comparing Figs. 3.9 and 3.10, it is clear that type 0 packets have the lowest PER when the SNR is low, which means that they perform the best for long-range communications. This is because type 0 is the shortest packet and the modulation it uses (FSK) is very robust; this comes, however, at the price of low bit rates (80 bps). For these reasons, this packet type is a proper choice for long-range control message exchange to resemble long-haul whale vocalizations.

The communication protocol for team formation is depicted in Fig. 3.22. To calculate the 'best' formation position for each glider - with the objective of minimizing the formation time while avoiding collisions -, the leader broadcasts a packet to collect the positions from the followers. Upon receiving the position packet from all the followers, the leader runs the formation mapping algorithm to find the best mapping (glider $\rightarrow$ vertex in the geometry); then, the leader informs each follower about their assigned formation position. The followers then acknowledge the reception of the message from the leader, and all the gliders start moving towards their assigned positions. All these control messages use short type 0 packets as their aim is to reach far apart gliders that are scattered in a wide region.

The gliders can move in regular formations as shown in Fig. 3.23. Different formation geometries can be used depending upon the number of gliders and the type of mission. Given the number

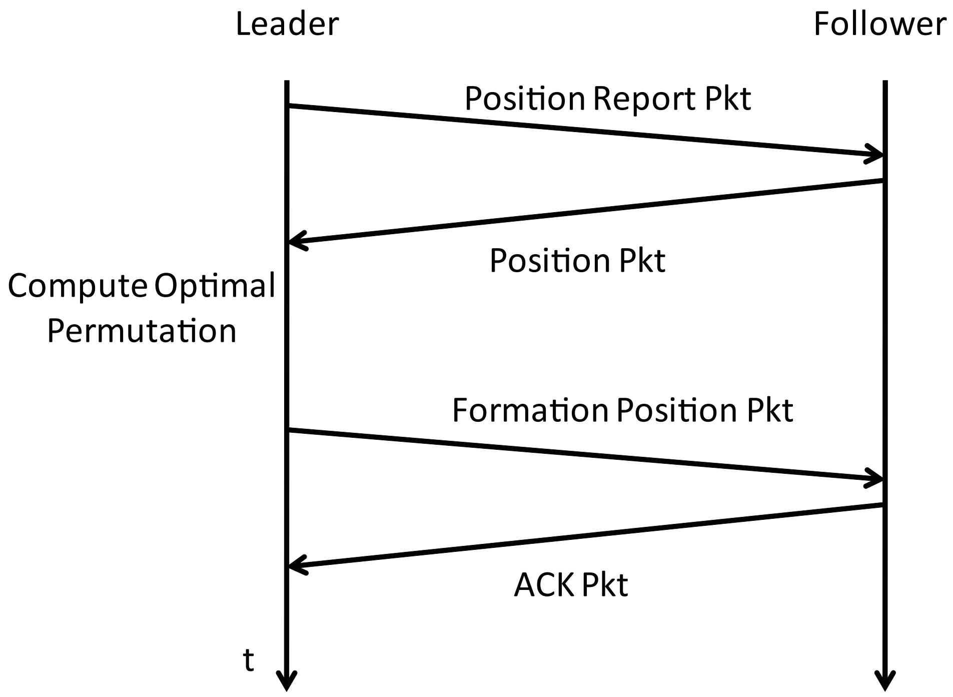


Figure 3.22: Protocol for Team Formation (Type 0 packets).

of gliders forming the team and the corresponding geometry formation, the problem that we face is to map every glider to its position in the formation. Selecting a position in the formation for a glider depends upon factors such as the time for that glider to reach this position, the possibility of collision with other gliders, and the permutation of the gliders with the formation positions. We have to determine the optimum to minimize both *the time* and *energy* spent to attain the formation. We first optimize on time to find the mapping, and then on energy consumption, while deciding on the exact trajectory for the selected mapping.

The formation optimization problem, which aims at mapping the gliders to formation positions, finds - out of all the permutations that avoid collisions (the so-called *feasible solutions*) - the best permutation that minimizes the time to form the team formation. Specifically, given $M = |\mathcal{N}|$ gliders $1, 2, \ldots, M$, and the corresponding formation points $G_1, G_2, \ldots, G_M$, we need to find a permutation $\pi \in \Pi$ such that the time spent by the gliders to form the formation is minimized while no collision occurs. Here, Π is the set of all $M!$ permutations.

For simplicity and because of the large inter-vehicle distances, in this dissertation a glider is considered to be a single mass point. Note, however, that our solutions can be straightforwardly extended to account for the real dimensions of the gliders by adding marginal spaces between the

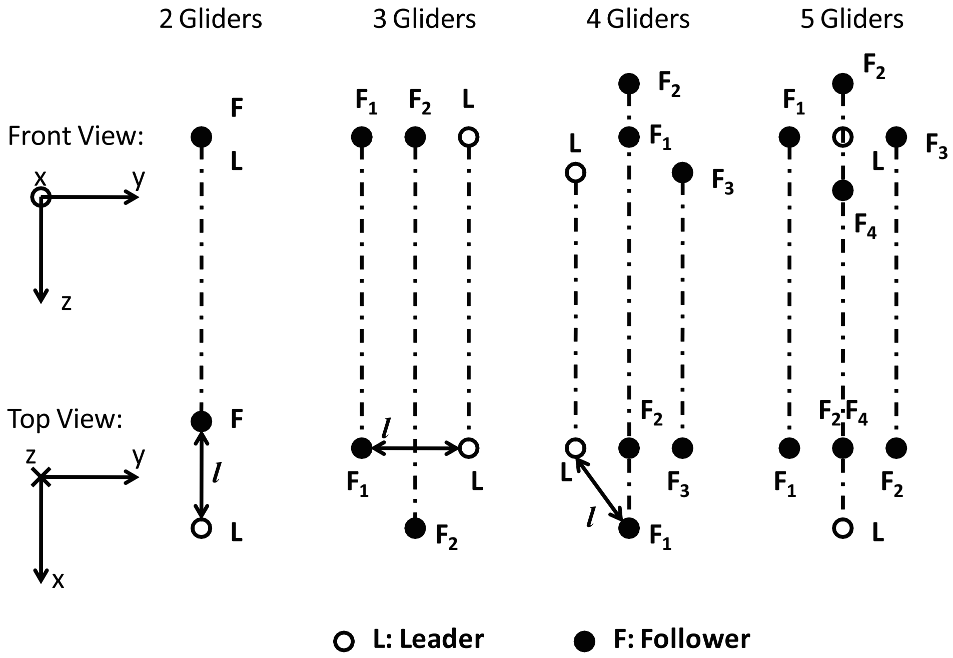


Figure 3.23: Formation geometries for 2 - 5 gliders in Front and Top Views, where the mission-specific inter-glider distance is l and the last surfaced glider is chosen as leader.

points. To ensure no collision among gliders, the *sufficient and necessary condition* is that two or more gliders of the team do not meet at the same point *and* at the same time as they move along their trajectories. However, solving this problem in the 3D space is complex; also, the solution would be affected by the uncertainty of the velocities of the gliders caused by ocean currents. Therefore, we adopt a simpler conservative approach that relies on a *sufficient condition* to avoid collisions. Note that the fastest way for a glider to move to a point is to follow the sawtooth trajectory laying in the vertical plane containing the current glider position and the destination point. Hence, a sufficient condition to ensure no collision is that the projections of the glider trajectories on the x-y plane - segments describing the horizontal advance of the gliders - do not intersect (Fig. 3.24).

If we denote the initial position of glider i and formation point G_i as $P_i^0 = (x_i^0, y_i^0, z_i^0)$ and $P_{G_i} = (x_{G_i}, y_{G_i}, z_{G_i})$, respectively, given the constant horizontal speed s_H of the gliders, the formation mapping problem can be formulated as,

$$\begin{aligned} &\textbf{Given: } P_i^0, P_{G_i}, s_H (\forall i = 1, \ldots, M) \\ &\textbf{Find: } \pi^* \in \Pi \\ &\textbf{Minimize: } \max\left(\sqrt{(x_i^0 - x_{G_j})^2 + (y_i^0 - y_{G_j})^2}/s_H\right) \\ &\textbf{Subject to: } G_j = \pi(i); \qquad (3.57) \\ &\pi \in \{\pi : \forall i, m, i \neq m, \Lambda_z(T_i^{\pi(i)}) \cap \Lambda_z(T_m^{\pi(m)}) \equiv \emptyset\}; \qquad (3.58) \end{aligned}$$

where $T_i^{\pi(i)}$ is the vertical trajectory from i to its mapped point $\pi(i)$ and $\Lambda_z()$ is the vertical projection to the x-y plane. Here, (3.57) is the mapping of glider i to G_j, and (3.58) ensures that the permutations incurring intersections of vertical trajectory projection, i.e., those unfeasible, not be considered.

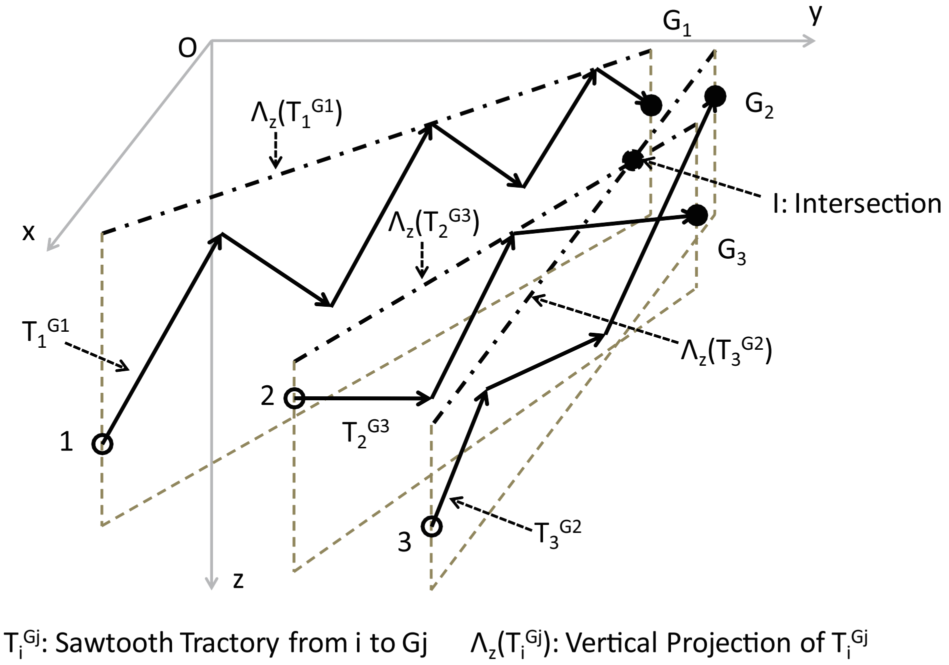


Figure 3.24: Mapping gliders 1, 2, and 3 to geometry vertexes G_1, G_3, and G_2, respectively. Note that gliders 2 and 3 may collide as $\Lambda_z(T_2^{G_3})$ and $\Lambda_z(T_3^{G_2})$ intersect at point I.

3.7.2 Team Steering

The team steering problem can be divided into two subproblems: 1) steering the team to follow the planned team trajectory and 2) maintaining the formation. As the leader (the last glider that has surfaced) has the most accurate position information, it is selected to estimate the *team dislocation*, i.e., the deviation from the target trajectory. The leader calculates the adjusted sawtooth trajectory to steer the team back to the target trajectory. Depending on the application requirements, the leader can decide to either move back to the closest point on the target trajectory, or to head towards the final destination of the target trajectory. While the former strategy is more conservative, as it minimizes the time to go back to the target trajectory, the latter is more energy efficient when the goal is to reach the final destination. The other gliders, i.e., the followers, will then focus on maintaining the geometry of the formation, which also implies following the leader's path. Due to space limitation, in the following we focus only on this second subproblem.

We use a hybrid approach to keep the team formation depending on whether the position information is *absolute* or *relative*. Specifically, *Absolute Formation Adjustment (AFA)* is used when absolute information such as gliders' position is available; whereas *Relative Formation Adjustment (RFA)* is used when relative information such as inter-vehicle velocity is available. The reason for this hybrid approach is to reduce the communication overhead for position information dissemination. Using absolute positions, in fact, requires the exchange of location information, which introduces overhead. On the other hand, relative inter-glider velocity information can be estimated by each glider by measuring the Doppler shift of ongoing inter-vehicle communications. These relative velocities can then be used to control the trajectory of each glider in such a way as to keep the inter-distance between gliders constant. While this 'opportunistic' approach does not guarantee that the absolute geometry is maintained (e.g., rotations can occur), it does not introduce additional overhead as it may exploit ongoing communications. Consequently, in order to compensate for the errors due to formation rotations, the team periodically goes back to AFA to readjust the geometry using absolute positions.

The communication protocol for hybrid steering is presented in Fig. 3.25. Periodically, each glider runs AFA using the position information obtained from the localization algorithm. Then, RFA is run using relative information extracted from inter-vehicle packets. Glider i's relative velocity is estimated by j when an inter-vehicle packet is received. This information is then embedded into the reverse direction packet and fed back to i. At this point, the gliders are able to make adjustments according to their relative velocity. Finally, if the leader (or any other follower) assesses that the geometry is seriously compromised (i.e., if the team dislocation is greater than the dislocation associated with a new permutation), the leader can rerun the formation optimization problem and find the new best permutation (often involving only a subset of the vehicles) to reconstruct the geometry.

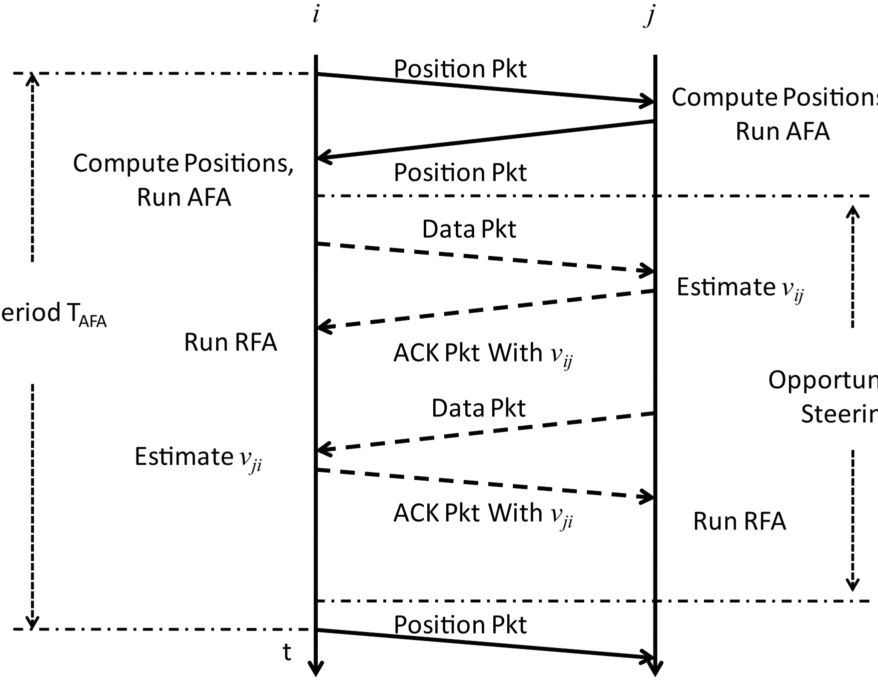


Figure 3.25: Hybrid steering using acoustic communications.

Intuitively, in order to keep the formation, two gliders need to move closer if the distance between them is larger than the initial specified distance, i.e., the equilibrium distance in the formation geometry. Conversely, they need to move farther if their distance is smaller than the equilibrium distance. In such a scenario, an *Attraction and Repulsion Model (ARM)* is appropriate to implement the swarming behavior using local controls. Bio-inspired algorithms based on the ARM have been proposed and analyzed in [71, 72]. Specifically, in [71] a class of attraction and repulsion functions for swarm formation is presented and their stability is analyzed, while in [72] a framework using artificial potentials and virtual leaders is proposed. Artificial potentials define interaction control forces between neighboring vehicles and an optimal inter-vehicle spacing is therefore enforced. Virtual leaders can be used to manipulate group geometry and direct group motion by means of additional artificial potentials. Closed-loop stability is proved and robustness to a single vehicle failure is shown.

In this dissertation, we account for the physical constraints characterizing SLOCUM gliders and their energy-efficient acoustic WHOI Micro-Modems, and propose a novel distributed attraction and repulsion swarming solution integrated with the communication mechanisms. As ARM is similar to a spring system in physics, we treat the team as such a system and we define a metric between i and j called *virtual potential energy*, $\mathcal{E}_{ij} = \frac{1}{2}k_{ij}\Delta x_{ij}^2$, where k_{ij} is the virtual spring constant and Δx_{ij} is the displacement from the expected formation equilibrium between i and j. Virtual spring constant between the leader L and a follower F, and between followers, are denoted by k_{LF} and k_{FF}, respectively. To emphasize the role of the leader, which has more recent (and therefore more accurate) location information and is in charge of steering the entire team along the target trajectory, we enforce $k_{LF} > k_{FF}$ so that the dislocation from the leader will have greater influence than that between followers. This will imply that the 'rigidity' of the edges of the team structure will not be homogeneous; rather, it will depend on the logic role of the vehicles at the vertexes, being higher when one of the two gliders of an edge is the leader.

When glider i is in its equilibrium formation position, the total virtual potential energy between i and its neighbors, $\mathcal{E}_i = \sum_{j \in \mathcal{N}_i} \mathcal{E}_{ij}$, will be zero; otherwise, it will be greater than zero, where $\mathcal{N}_i$ is the set of neighbors of i. To keep the specified formation, i should adjust its pitch (α_i) and yaw (β_i) angles so that $\mathcal{E}_i$ can be minimized. For AFA, given the team glider positions P_j and directions α_j and β_j, with $j = 1, 2, \ldots, M$, which are obtained by exchanging control packets, in a given interval δ [s] glider i will adjust its pitch and yaw by solving,

$$\begin{aligned}
\textbf{Given: } & P_i, d_{ij}, s_H, \delta, \alpha_j, \beta_j (\forall j \in \mathcal{N}_i) \\
\textbf{Find: } & \alpha_i^* \in [\alpha_{min}, \alpha_{max}], \beta_i^* \\
\textbf{Minimize: } & \mathcal{E}_i = \frac{1}{2} \sum_{j \in \mathcal{N}_i} k_{ij} \Delta x_{ij}^2 \\
\textbf{Subject to: } & \Delta x_{ij} = \|\overrightarrow{P_i P_j} + (\vec{\mathbf{v}}_j - \vec{\mathbf{v}}_i)\delta\| - d_{ij}; \qquad (3.59) \\
& \|\vec{\mathbf{v}}_i\| \cdot \cos\alpha_i = s_H; \qquad (3.60)
\end{aligned}$$

where d_{ij} is the equilibrium distance between i and j in the formation, $\overrightarrow{P_i P_j}$ is the location vector from i to j, $\vec{\mathbf{v}}_i$ is i's velocity, and $\| \cdot \|$ is the vector length. Note that the velocity of each glider $j \in \mathcal{N}_i$ can be computed at i as $\vec{\mathbf{v}}_j = (s_H \cdot \cos\beta_j, s_H \cdot \sin\beta_j, s_H \cdot \tan\alpha_j)$.

For RFA, we adopt a bio-inspired communication technique that imitates the echolocation mechanism of the bat. A bat estimates the distance to an object by shouting and then measuring the acoustic echoing time from the object. Also, a bat relies on the Doppler effect, i.e., the frequency shift caused by the relative velocity, to sense an object's direction. Specifically, if the object is moving away from the bat, the returning echo will have a lower frequency than the original sound; conversely, the echo from an object moving towards the bat will have a higher frequency. When we do not rely to absolute position information, we use a similar technique to keep the swarm formation.

The WHOI Micro-Modem can estimate the relative speed of the transmitter exploiting the frequency shift caused by the Doppler effect. Suppose that during steering glider i obtains its relative speed s_{ij} (*a scalar*) with respect to another glider j. This can be extracted from ongoing inter-vehicle communications without additional overhead: upon receiving i's packet, j can estimate the Doppler frequency shift Δf_{ij}; the relative speed s_{ij} of glider i to j along the line connecting the two gliders is then calculated from $\Delta f_{ij} = -s_{ij} \cdot f_0/c$, where f_0 is the current acoustic communication central frequency and c is the average underwater acoustic wave speed (1500 $\mathrm{m/s}$). Glider j then sends s_{ij} back to i with its own location P_j, which can be estimated using the leader's GPS position, and relative location and velocity. Both s_{ij} and P_j can be embedded in the ongoing communication packets to avoid additional overhead. In this way, i computes its relative speed vector with respect to j as $\vec{\mathbf{v}}_{ij} = s_{ij} \cdot \frac{\overrightarrow{P_iP_j}}{\|\overrightarrow{P_iP_j}\|}$.

Consequently, the expected virtual potential energy $\mathcal{E}_i$ after time δ can be estimated as $\mathcal{E}_i = \frac{1}{2}\sum_{j\in\mathcal{N}_i} k_{ij}\|\vec{\mathbf{v}}_{ij}\delta\|^2$. Hence, the problem of steering i back into formation becomes the search for the optimal pitch and yaw to obtain a correction velocity $\vec{\mathbf{v}}_i$ such that $\mathcal{E}_i$ can be minimized,

$$\begin{aligned} \textbf{Given:} &\quad \vec{\mathbf{v}}_{ij}, s_H (\forall j \in \mathcal{N}_i) \\ \textbf{Find:} &\quad \alpha_i^* \in [\alpha_{min}, \alpha_{max}], \beta_i^* \\ \textbf{Minimize:} &\quad \mathcal{E}_i = \tfrac{1}{2}\textstyle\sum_{j\in\mathcal{N}_i} k_{ij}\|(\vec{\mathbf{v}}_{ij} + \vec{\mathbf{v}}_i)\delta\|^2; \\ \textbf{Subject to:} &\quad \|\vec{\mathbf{v}}_i\| \cdot \cos\alpha_i = s_H. \end{aligned} \tag{3.61}$$

By solving this problem, glider i is able to fix its own steering so that the formation error, i.e., the virtual potential energy, can be minimized. Note that this is a distributed solution as only local information from i's neighbors is needed.

Chapter 4

Performance Evaluation

In this chapter we first present the testbed emulator in Sect. 4.1, which we proposed for evaluating the performance of our solutions. Then we evaluate the performance of our proposed solutions for inter-vehicle communication and coordination. In Sect. 4.2, we evaluate the accuracy of our internal- and external-uncertainty models; in Sect. 4.3, we evaluate the performance of the QUO VADIS solution against existing geographic routing and DTN solutions; in Sect. 4.4, we evaluate the performance of our geocasting solution; in Sect. 4.5, we evaluate the performance of our under-ice localization solution; and in the last part, we evaluate the performance of our team formation and steering solution.

4.1 Testbed Emulator

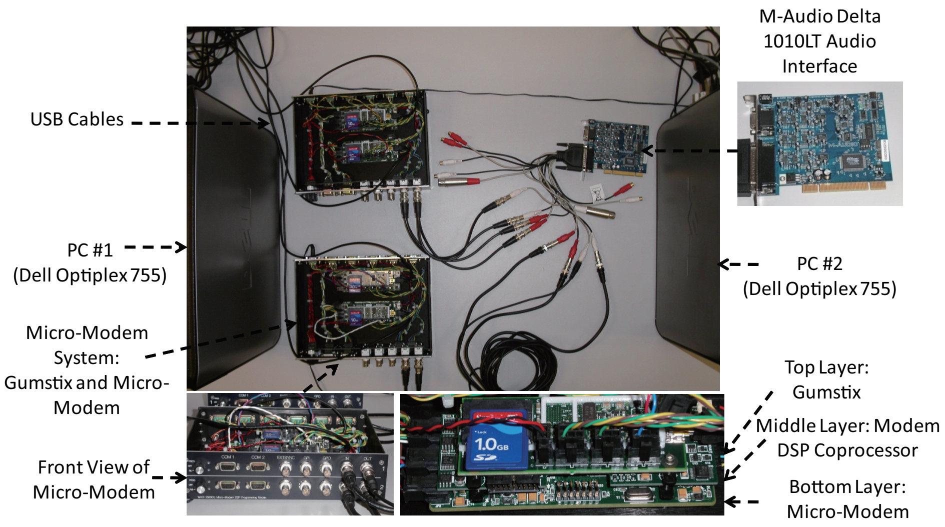

M-Audio Delta 1010LT Audio Interface
USB Cables
PC #1 (Dell Optiplex 755)
PC #2 (Dell Optiplex 755)
Micro-Modem System: Gumstix and Micro-Modem
Top Layer: Gumstix
Middle Layer: Modem DSP Coprocessor
Bottom Layer: Micro-Modem
Front View of Micro-Modem

Figure 4.1: Underwater communication emulator using WHOI Micro-Modems.

Our protocol and cross-layer design are closely coupled with the functionalities of Micro-Modem and testbed. Therefore, in this section, we present the physical and logical architecture of our underwater network testbed. Our underwater testbed relies on a multi-input multi-output audio interface installed on a Personal Computer (PC) and can process real-time signals using software. With the help of softwares such as MATLAB, we can precisely adjust the signal gains, introduce propagation delay, mix the acoustic signals, and add ambient noise and interference in real time. Consequently, underwater communication models described in [4] can be emulated.

4.1.1 Physical Architecture

As shown in Fig. 4.1, the physical architecture of our testbed includes the following components.

- **WHOI Acoustic Micro-Modem:** Low-power underwater acoustic Micro-Modem developed by WHOI. It can transmit 4 different types of packets at 4 data rates (Table 3.1) in four different bands from 3 to 30 kHz. Control of the Modem is by NMEA commands [73].

- **Audio Interface:** M-Audio Delta 1010LT PCI Audio Interface [74]. It is a 10-In 10-Out 24-bit PCI audio interface card with maximum sampling rate of 96kHz. It can process the audio signals from multiple inputs in real time and route them to corresponding outputs.

- **Gumstix Motherboard (GM):** The embedded system with Marvell PXA255 400 MHz processor, 64 MB RAM, and 1GB SD disk storage [75]. It runs OpenEmbedded Linux and controls the modem via serial port. It is connected to PCs through the USB port.

- **PC #1:** A Dell Optiplex 755 desktop with Intel 2.4 GHz Quad Core CPU and 2GB RAM. It runs the computer emulation controller software commanding the GMs. It also controls the channel emulator running at PC #2 via Ethernet and collects the emulation results from the GMs.

- **PC #2:** The same configuration as PC #1. It listens to the control information through the Ethernet from PC #1 and emulates the underwater communication channels including signal gain change and ambient noise.

4.1.2 Logical Architecture

The logical architecture of our testbed is shown in Fig. 4.2. When the emulation starts, the Emulation ConTroLler (ECTL) at PC #1 issues commands to the channel emulator at PC # 2, which will

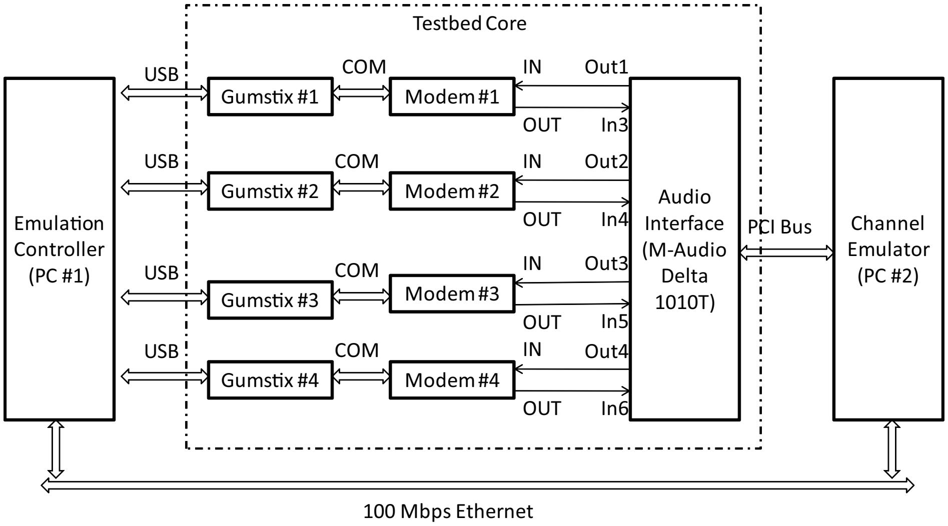


Figure 4.2: Logical architecture of proposed testbed.

start emulating the channel according to the parameters provided. ECTL then requests the GMs to run their network tasks. Whenever a packet is transmitted or received, the GMs inform the ECTL via USB connection so that ECTL can collect the emulation results and issue further commands.

Upon the command from PC #1, channel emulator at PC #2 will adjust the gains of input signals, mix them, introduce propagation delay, add ambient noise, and route the processed signals to corresponding outputs. Acoustic signals are processed in real-time with MATLAB using real-time audio processing package *Playrec* [76].

The acoustic gain for each node to node connection in simulation is determined by a Bellhop-based simulator. For each transmitter-receiver pair, a simulation is run to determine the incoherent gain given the transmitter depth, receiver depth, and receiver range. Typically the Munk sound speed profile is used for the simulation because it is a representative open ocean acoustic environment with a depth of 5000 m.

Note that due to the limited number of Micro-Modems and audio processing channels, we can only mix signals from up to 3 transmitters at the receiver modem. Therefore, we calculate, select for transmission, and mix with ambient noise, only the three most powerful signals the receiver will encounter. This is good enough in general as the inter-vehicle traffic underwater is generally low. We leave the simulation of more than three simultaneously transmitted signals as a problem for further research.

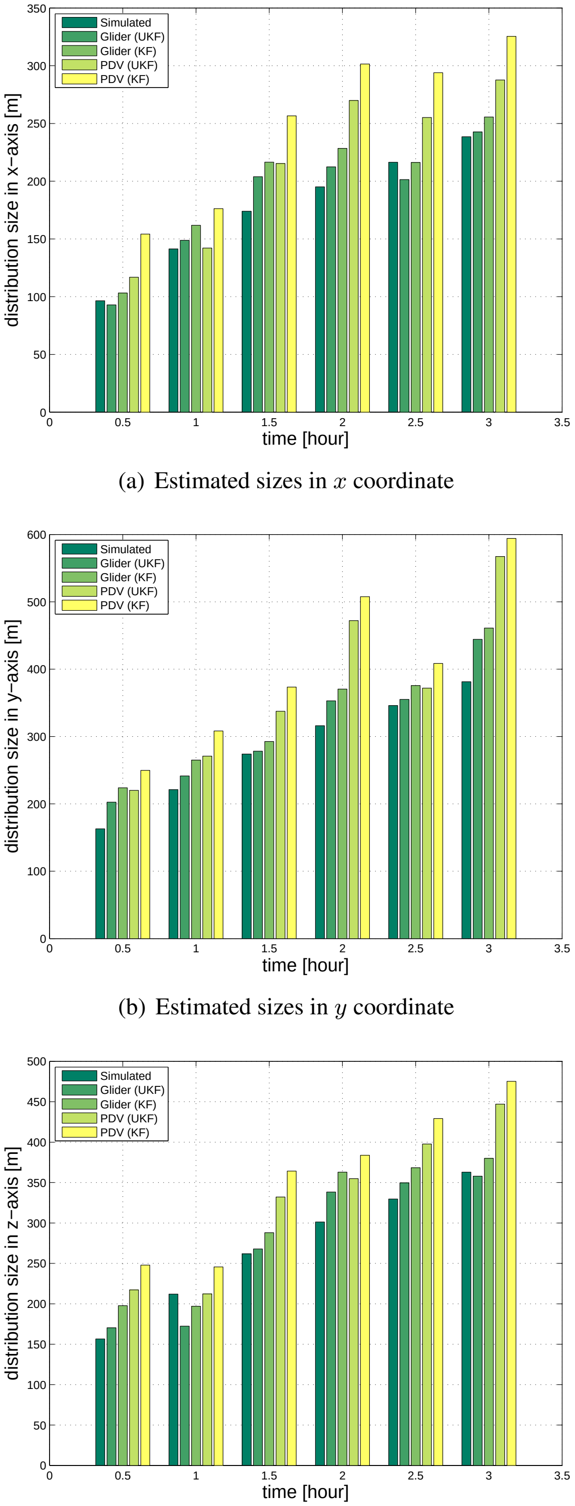
Table 4.1: Emulation Parameters for Evaluating the Position Uncertainty Model

Parameter	Value
Initial deployment region	2.5(L)×2.5(W)×1(H)Km^3
Transmission Power	[1, 10] W
Glider Horizontal Speed	0.3 $\mathrm{m/s}$
Gliding Depth Range	[0, 500] m
Pitch Angle Range	[10°, 35°]

4.2 Performance Evaluation of Position Uncertainty Model

In this section, we first list out the setup for the simulations. Then the accuracy of our prediction model is evaluated. Assume that a glider's drifting (i.e., the relative displacement from the glider's trajectory) is a 3D random process $\{X(t), t \geq 0\}$ as the following [77]: 1) In the beginning of the deployment, the drifting is 0,i.e., $X(0) = (0, 0, 0)$; 2) The drifting has independent increments, in that for all $0 \leq t_1 < t_2 < \cdots < t_n$, $X(t_n) - X(t_{n-1})$, $X(t_{n-1}) - X(t_{n-1}), \ldots, X(t_2) - X(t_1)$, $X(t_1)$ are independent; 3) The drifting has stationary increments, in that the distribution of $X(t + \varphi) - X(t)$ does not depend on t and is normally distributed with zero mean and covariance matrix $\varphi\phi^2 I_3$, where I_3 is the 3×3 identity matrix, and ϕ is a scaling factor that decides the magnitude of drifting. Note that this drifting model is ideal since the drifting in any of the x, y, z directions is Gaussian. Simulation parameters are listed in Table 4.1 and the AUVs are initially randomly deployed in the 3D underwater region. We use typical velocity for PDVs, which varies from 2 to 10 $\mathrm{Km/hr}$ [57]. The velocity of PDVs is dependent on various non-linear factors like drag force, and friction of the motor. For the underwater glider, we assume the trajectory segment can be described by the linear form as discussed in Sect. 3.3. For PDV, we assume the trajectory segment can be described by the quadratic form as $(x, y, z) = 0.5\boldsymbol{\zeta} t^2 + \boldsymbol{\eta} t + \mathbf{P}_0$, where $\boldsymbol{\zeta}$, $\boldsymbol{\eta}$, and $\mathbf{P}_0$ denote the acceleration vector, velocity vector and initial position of the vehicle, respectively. This can be used to describe the kinematic model of a PDV such as that in [78].

We evaluate the accuracy of our prediction model using statistical non-linear regression and UKF algorithm. We are also interested in comparing the external-uncertainty prediction accuracy of our proposed UKF algorithm with that predicted using Kalman Filter. To do this, we compare the 3D sizes and probability mass functions (pmfs) between those obtained in simulations and those predicted by our model. Simulations of 100 rounds are done for predictions of gliders and PDVs

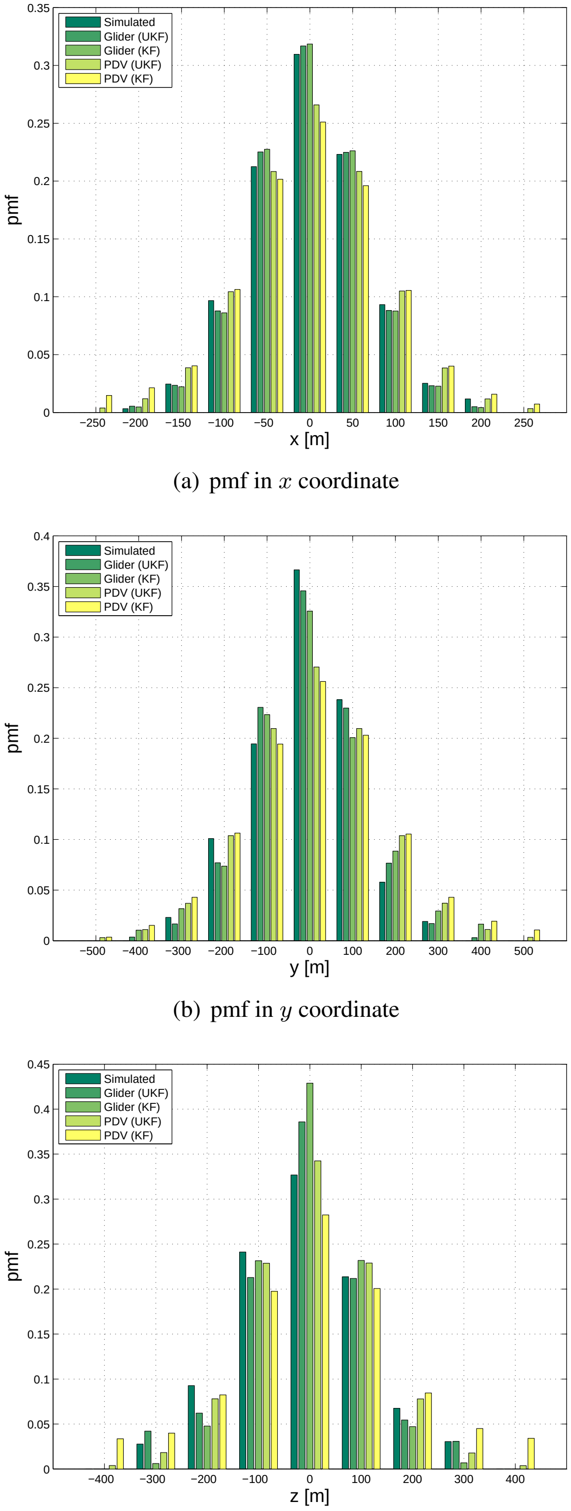


(a) Estimated sizes in x coordinate

(b) Estimated sizes in y coordinate

(c) Estimated sizes in z coordinate

Figure 4.3: Estimated region sizes.



(a) pmf in x coordinate

(b) pmf in y coordinate

(c) pmf in z coordinate

Figure 4.4: Estimated region pmfs.

and the average results are plotted in Figs. 4.3 and 4.4. Note that the ‘Glider (UKF)’ and ‘Glider (KF)’ denote the uncertainty for a glider predicted using the UKF and KF, respectively, while the notations for PDV have similar notations.

From these figures, we can see that our external uncertainty model using UKF generally gives more accurate predictions than that using KF on the region sizes and distribution functions for both types of vehicle. In any of these axis directions, the vehicle may be randomly located in a range $[\tau - \rho/2, \tau + \rho/2]$, where τ is the expected location of this vehicle in this direction. We call ρ the size of the uncertainty region as it decides the range that the AUV may be distributed. Figure 4.3 plots these sizes at different times, where the horizontal axis is the time duration that an AUV stays underwater. We assume at time 0 there is no position uncertainty (e.g., AUVs are on the ocean surface where GPS is accessible) and we assume our estimations of the external uncertainty are run at the same time. To better compare the distribution functions, we also align the pmfs (i.e., move the expected positions of the vehicle in these three cases to 0) in Fig. 4.4. Each pmf value at a discrete position, say x_0, is calculated by checking if the vehicle lies in the range $[x_0 - \psi/2, x_0 + \psi/2]$, where ψ is the interval size. Note that the prediction accuracy for glider is generally better than that for PDV. This is because gliders follow saw-tooth trajectories (piece-wise line segments), which is easier to predict than the non-linear trajectories of the PDVs. Also note that the longer the AUV stays underwater, the less accurate the prediction will be. Provided an accuracy threshold, our model can also be used for AUVs to decide when to surface for position correction (e.g., getting a GPS fix).

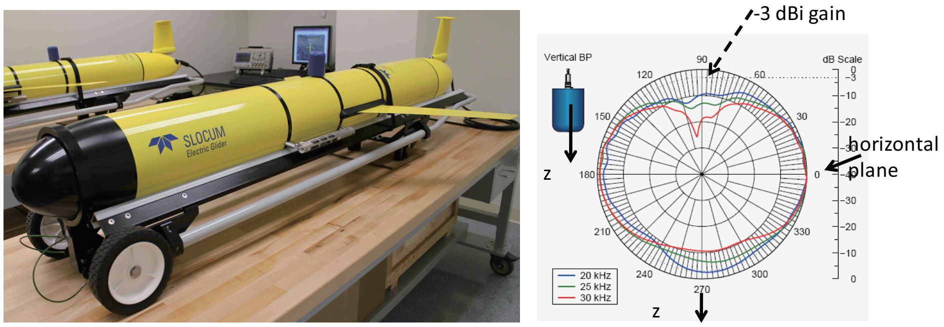


Figure 4.5: Picture of the SLOCUM underwater glider with BT-25UF mounted on top (Courtesy of Hans. C. Woithe and Dr. Ulrich Kremer) and radiation pattern of the BT-25UF transducer.

Table 4.2: Emulation Scenario Parameters for Evaluating QUO VADIS

Parameter	Value
Deployment 3D region	2500(L)$\times$2500(W)$\times$1000(H) m^3
Confidence Parameter α	0.05
$[P_{min}, P_{max}]$	[1, 10] W
Packet Types Ξ	$\{$0, 2, 3, 5$\}$
Glider Horizontal Speed	0.3 $\mathrm{m/s}$
Gliding Depth Range	[0, 100] m
Carrier Frequencies	10, 15, 25 kHz
$B_{\max}$	10 hr

4.3 QoS-aware Underwater Optimization Framework for Inter-vehicle Communications

QUO VADIS is implemented and tested on our underwater communication emulator as shown in Fig. 4.1. We are interested in evaluating the performance of the proposed solution in terms of e2e energy consumption, e2e reliability (i.e., e2e delivery ratio), average bit rate of a link, and overhead, under an environment that is described by the Bellhop model (and the Munk acoustic speed profile as input).

Assume that a glider's drifting model is the same as in Sect. 4.2. Emulation parameters are listed in Table 4.2. The radiation pattern of the BT-25UF transducer (Fig. 4.5) is used in the emulations. Every 10 seconds, a packet is generated in each node. A glider is randomly selected as the collector and half of the other gliders are randomly selected to forward their packets towards it. Note that it actually is a scenario for deep water. We will also evaluate the performance in shallow water, where acoustic waves propagate differently.

We are interested in evaluating the performance of our solution for the two classes of traffic in Sect. 3.4.2, using either the BT-25UF transducer or an ideal omni-directional transducer (with gain equal to 0 dBi). We also want to compare the performance of our solution, which delays the transmission for optimal topology configuration, with the solution without delaying the transmission. For convenience, we denote QUO VADIS for Class I traffic using the BT-25UF transducer, for Class I traffic using the ideal omni-directional transducer, for Class II traffic using the BT-25UF transducer, for Class I traffic using the ideal omni-directional transducer, the solution with no delaying of the transmission (i.e., $\Delta t = 0$ for $\mathbf{P}(\mathbf{i}, \mathbf{d}, \mathbf{t_{now}}, \mathbf{\Delta t_p})$) by 'QUO VADIS I', 'QUO VADIS I - OMNI', ‘QUO VADIS II’, ‘QUO VADIS II - OMNI’, and ‘QUO VADIS - ND’. We will also compare the performance of our solution with geographical routing solutions – MFR, GRS, CRM, and PTKF – and DTN solutions – RAPID, Spray and Wait, and MaxProp – as review in Sects. 2.1 and 2.2. To make the comparison fair, we use two variant protocols for each of these solutions by adding the constraints of the two classes of traffic to these solution. For example, we denote the MFR solution with Class I constraints in (3.37) by ‘MFR I’, and the solution with Class II constraints in (3.37) by ‘MFR II’.

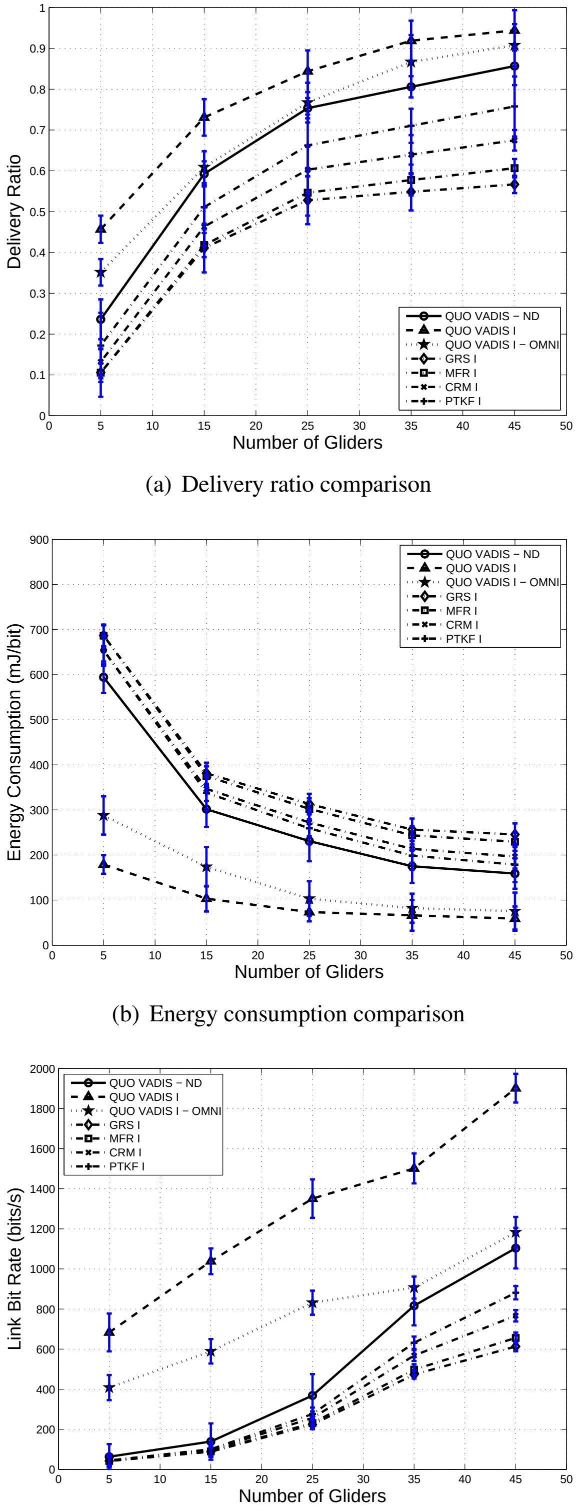
The following networking metrics are compared:

- **e2e energy consumption**: the average energy consumed to route one bit of data to the destination;
- **e2e delivery ratio**: the number of data packets received correctly over the number of data packets sent;
- **link bit rate**: the average bit rate between a transmission pair;
- **overhead**: the number of bytes used for position and control to facilitate the transmission of payload data.

For statistical relevance, emulations are run for 50 rounds and the average is plotted with 95% confidence interval. Emulations are done for different settings and the results are plotted and discussed in the following subsections.

4.3.1 Comparison With Geographic Routing Protocols

We compare the performance of our solution with geographic routing protocols in Figs. 4.6 and 4.7. As shown in these two figures, we can see that QUO VADIS has better performance than QUO VADIS - OMNI and QUO VADIS - ND for the same class of traffic in terms of these three metrics. By delaying packet transmissions to wait for the optimal network topology, the e2e energy consumption is reduced while the e2e delivery ratio and link bit rate increase (e.g., with 5 gliders, the energy consumption for QUO VADIS I is around 30% of that for QUO VADIS-ND). By exploiting the frequency-dependent radiation pattern of the transducer, received signal power may obtained a gain of up to 20 dB, which we observed in the simulations. Hence QUO VADIS using the BT-25UF

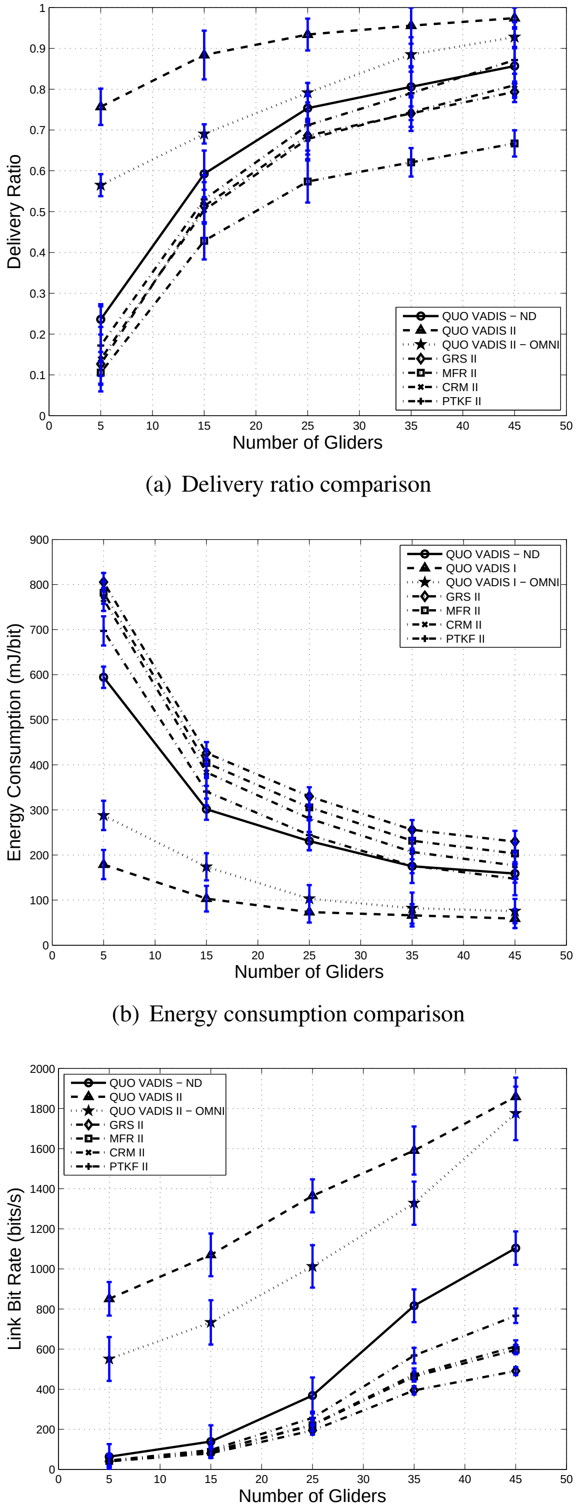


(a) Delivery ratio comparison

(b) Energy consumption comparison

(c) Link bit rate comparison

Figure 4.6: Performance comparison for Class I traffic with *geographic routing* protocols.

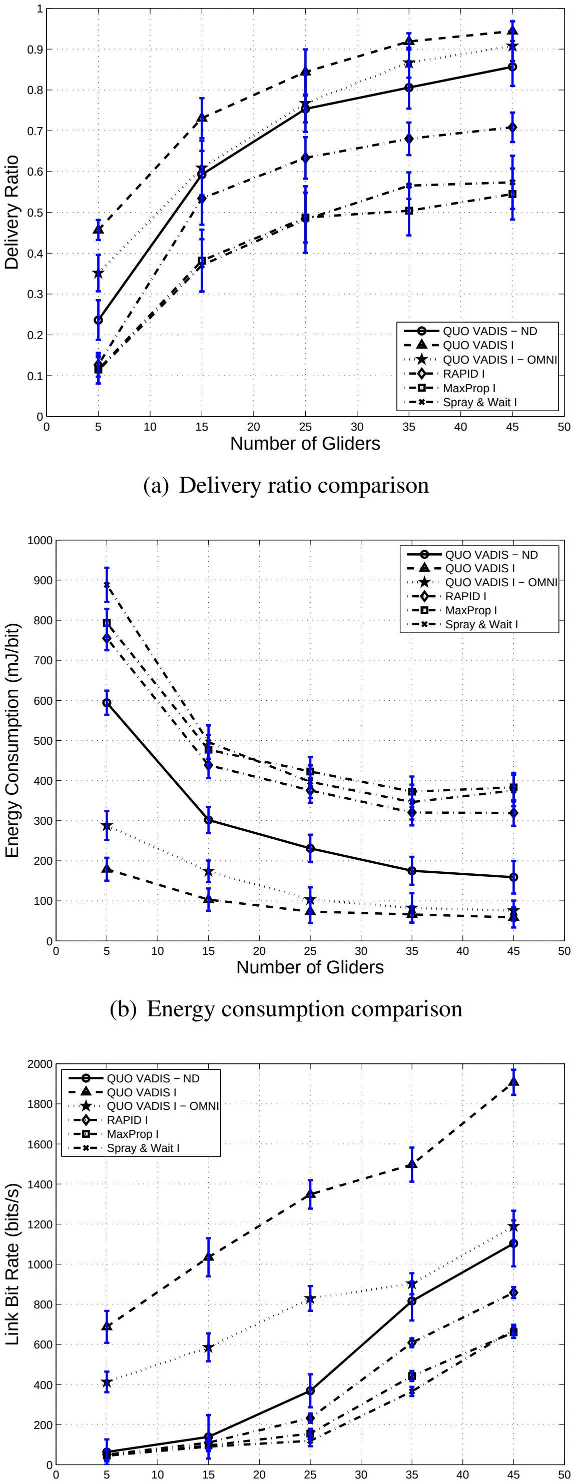


(a) Delivery ratio comparison

(b) Energy consumption comparison

(c) Link bit rate comparison

Figure 4.7: Performance comparison for Class II traffic with *geographic routing* protocols.

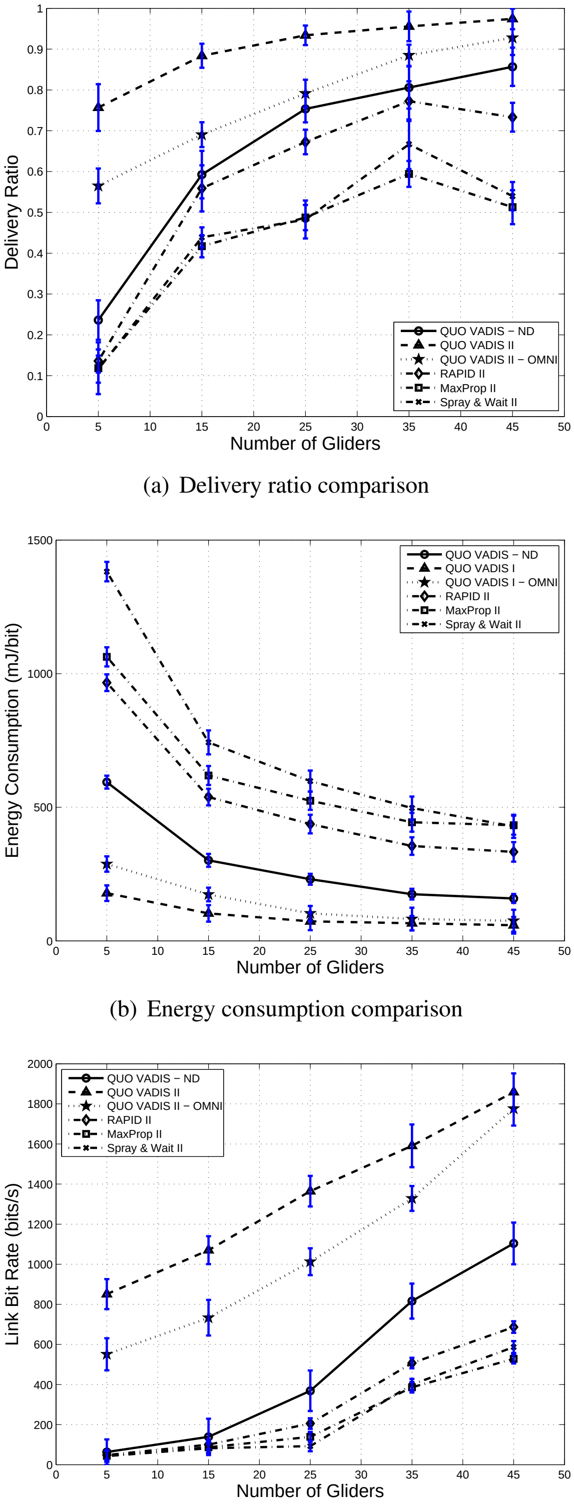


(a) Delivery ratio comparison

(b) Energy consumption comparison

(c) Link bit rate comparison

Figure 4.8: Performance comparison for Class I traffic with *DTN* protocols.

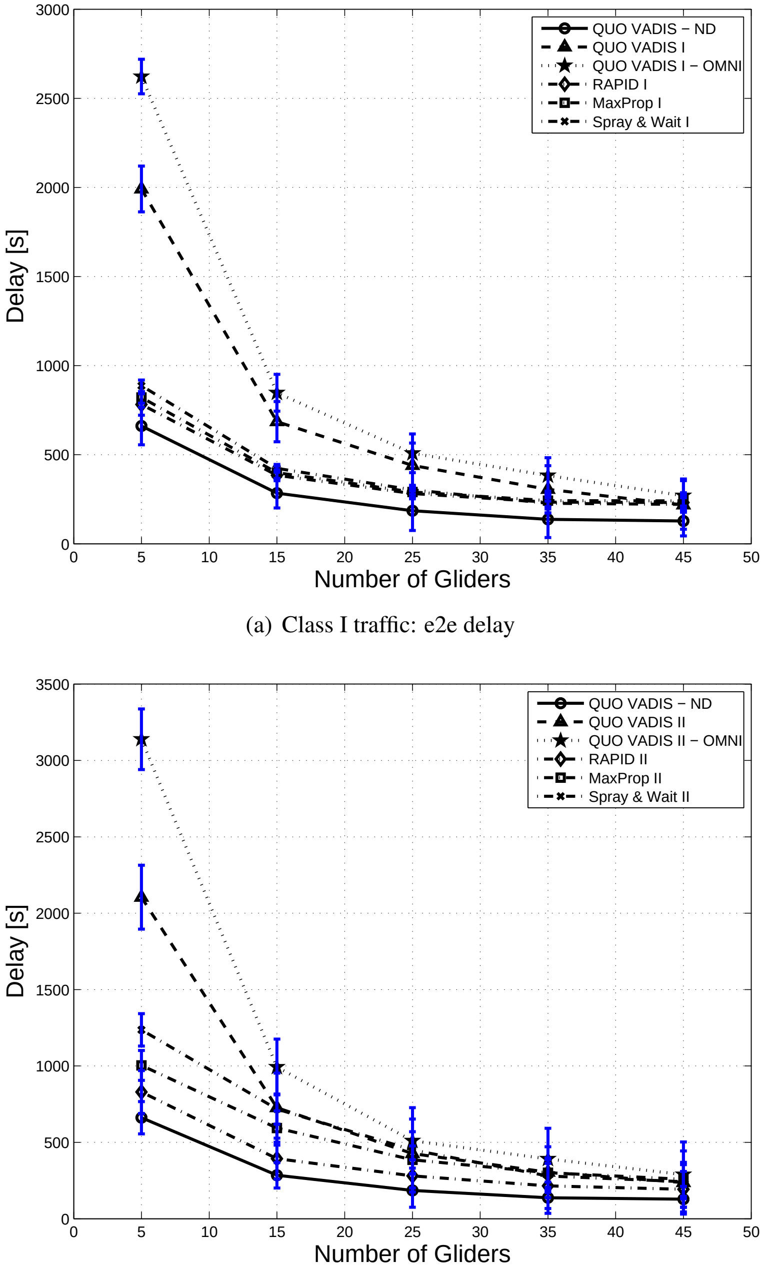


(a) Delivery ratio comparison

(b) Energy consumption comparison

(c) Link bit rate comparison

Figure 4.9: Performance comparison for Class II traffic with *DTN* protocols.



(a) Class I traffic: e2e delay

(b) Class I traffic: e2e delay

Figure 4.10: Comparison of e2e delay with *DTN* protocols.

transducer has better performance than that using the omni-directional transducer. Due to the QoS requirements, retransmissions are needed to recover link errors, resulting in higher e2e delivery ration for Class II traffic than for Class I traffic. On the other hand, this leads to more energy consumption.

Different versions of our QUO VADIS solutions also perform better than geographic routing protocols GRS, MFR, CRM and PKTF. This is because that uncertainty in location leads to errors in route selection, packet transmissions and transmission power estimates. Also these geographic routing protocols do not consider the propagation delay underwater, which results in degraded communication performance. Interesting enough, we can see that among these geographic routing protocols, PKTF offers the best performance. This is because it jointly considers the transmission power and routing to minimize the e2e energy consumption. Therefore it performs better than the other geographic routing protocol, which only consider the distance or angle metrics for routing (not closely related to network performance). GRS gives the worst performance since it generally needs to forward the packet to the node that is far from the transmitter, which introduces bad link performance. Similarly, CRM performs better than MFR as the CRM has less probability to forward packets to node that is far away than MFR does.

4.3.2 Comparison with DTN Solutions

We further compare QUO VADIS with the DTN solutions – RAPID, MaxProp and Spray and Wait. As shown in Figs. 4.8 and 4.9, QUO VADIS gives improved performance over RAPID, MaxProp and Spray and Wait. The is mainly due to that these DTN solutions transfer packets once the neighbors are in the transmission range. Such schemes may be good for scenarios where the connectivity is intermittent. However, the performance may not be optimal since this may not be the time to achieve the best link performance. In contrast, QUO VADIS predicts and waits for the best network configuration, where nodes move closer for the best communications. So the e2e delivery ratio and link bit rate of QUO VADIS is the highest while its energy consumption is minimal. Note that among these compared DTN solutions, RAPID performs the best. This is because RAPID prioritizes old packets so they won't be dropped. MaxProp gives priority to new packets; older, undelivered packets will be dropped in the middle. Spray and Wait works in a similar way, which does not give priority to older packets. On the other hand, Spray and Wait is slightly better than MaxProp. This is because in our scenario, the network connectivity is not disrupt. The way MaxProp routes based on the e2e delivery ratio estimation will be very different from that Spray and Wait does, i.e., just transmits the packet to a neighbor then lets the neighbor continue to forward it. Moreover, MaxProp still needs to pay for the overhead to obtain the global e2e delivery ratio information.

4.3.3 End-to-end Delay Comparison

To see QUO VADIS can meet the delay requirement of the delay-tolerant traffic, we also calculate and plot the e2e delays of these solutions. As shown in Fig. 4.10, QUO VADIS - ND gives the least e2e delay. Compared to QUO VADIS and QUO VADIS - OMNI, QUO VADIS - ND does not wait for the vehicles to move to the optimal configuration yet more retransmissions are necessary. As the vehicle speed is much slower than the acoustic speed, QUO VADIS - ND still needs much less time than QUO VADIS and QUO VADIS - OMNI even though more retransmissions are needed (thus resulting in more communication delay). Similarly, the huge difference between vehicle speed and acoustic speed leads to the result that QUO VADIS and QUO VADIS - OMNI need more time than the DTN protocols (RAPID, MaxProp, and Spray and Wait), especially when the number of vehicles is small (where average inter-vehicle distance is large). On the other hand, by taking the position uncertainty into account, communications using QUO VADIS - ND is more reliable than those using RAPID, MaxProp or Spray and Wait so less delay is incurred. QUO VADIS has less delay than QUO VADIS - OMNI due to the improvement in communications by exploiting the directional transducer gain. Also Class II traffic generally has more e2e delay than Class I due to the need for retransmissions. Last, note that as the number of gliders increases, the delays of QUO VADIS and QUO VADIS - OMNI drop quickly. This is because average inter-vehicle distance becomes smaller and the number of close neighbors increases, which reduces the need for a glider to wait a long time until a neighbor moves close.

4.3.4 Overhead Comparison

We plot and compare the overheads (per node) of these protocols in Fig. 4.11. Note that as QUO VADIS, QUO VADIS - ND, and QUO VADIS - OMNI work almost the same way, i.e., the uncertainty region information is broadcast periodically (here the period is taken to be 60 s), their

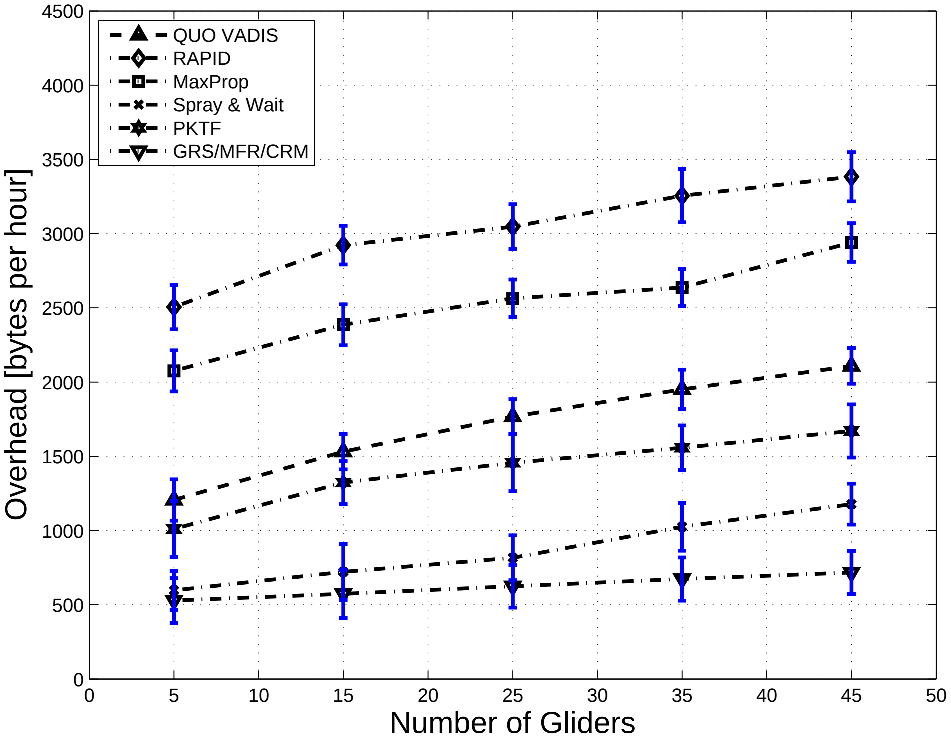


Figure 4.11: Comparison of the overhead.

overheads are the same and thus we use QUO VADIS in the figure to represent these variant versions. Similarly, nodes running the geographic routing protocols GRS, MFR and CRM only need to periodically broadcast the position information so their overhead is basically the same. Hence we use GRS/MFR/CRM to represent them.

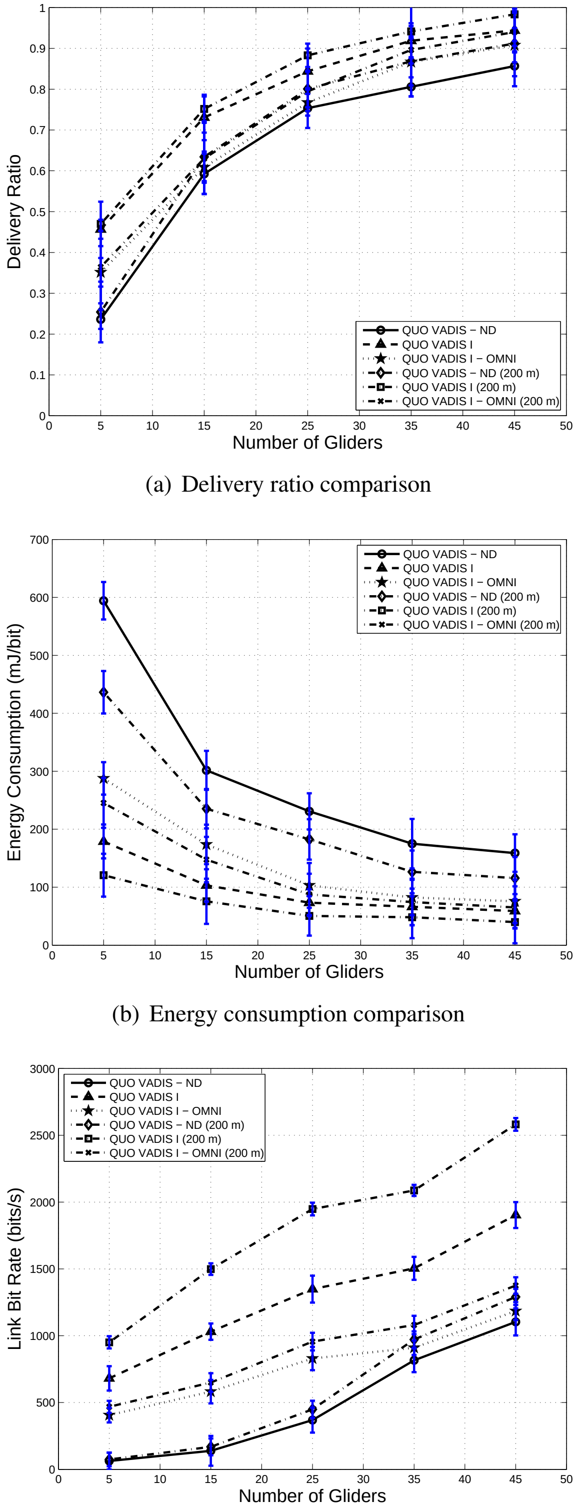
Surprisingly, even though QUO VADIS achieves the best network performance, its overhead is not the biggest. The protocols with the larger overhead are RAPID and MaxProp. In order to work, RAPID needs the following control information: average size of past transfer opportunities, expected meeting times with nodes, list of packets delivered since last exchange, the updated delivery delay estimate based on current buffer state, and information about other packets if modified since last exchange with the peer, which takes a large number of bytes. MaxProp needs to exchange a list of the probabilities of meeting every other node on each contact, which is basically global information. It also has the neighbor discovery overhead. Compared to RAPID and MaxProp, QUO VADIS only needs to exchange the external uncertainty information of itself and the destination node, which is obviously less. On the other hand, PKTF needs a probe message that has five data fields. Only the nodes in the selected path are required to respond with a probe – whether it is sent for the forwarding or reverse direction. The Spray and Wait protocol reduces transmission overhead by spreading only a few number of data packets to the neighbors. The source node then stops forwarding and lets each node carrying a copy perform direct transmission. In our emulation, we select the number to be one to make the comparison fair and hence the overhead is small. Lastly, for the other geographic routing protocols GRS, MFR and CRM, the nodes just need to know the geographic locations of the neighbors and the destination. Therefore the overhead required is the least. Note that here it is not necessary to differentiate the two classes of traffic since the overhead difference is small.

4.3.5 Performance in Shallow Water

So far the results are obtained using the setting in Table 4.2, which is for the deep water. We change the network scenario to the shallow water scenario by setting the depth of the 3D region to 200 m. In this shallow water scenario, the path loss estimated by the Urick's model is very different from that estimated by the Bellhop model. We had anticipated the performance will degrade because of this mismatch. Surprising enough, as shown in Fig. 4.12 and 4.13, we find the performance (in terms of e2e delivery ratio, energy consumption, and link bit rate) in the shallow water is actually better. A more careful analysis reveals the reason – the existence of the *surface duct* in the shallow water. Surface duct is basically a zone below the sea surface where sound rays are refracted toward the surface and then reflected. The rays alternately are refracted and reflected along the duct out to relatively long distances from the sound source. Hence the acoustic waves are relatively concentrated in the surface duct, leading to less path loss. This consequently leads to improved network performance.

4.3.6 Performance using Different Uncertainty Update Intervals

Emulations so far have been fixing broadcast interval of uncertainty region to 60 s. Our last interest is to evaluate the performance of the QUO VADIS variants when different broadcast intervals are used. Therefore we re-run the emulations for two more cases: i) half of interval (i.e., 30 s); and ii) double of interval (i.e., 120 s). From Fig. 4.14 and 4.15, we can see that the performance of the QUO VADIS variants becomes worse when the update interval is doubled. This is because when the interval is doubled, the position uncertainty information becomes less accurate. This leads to

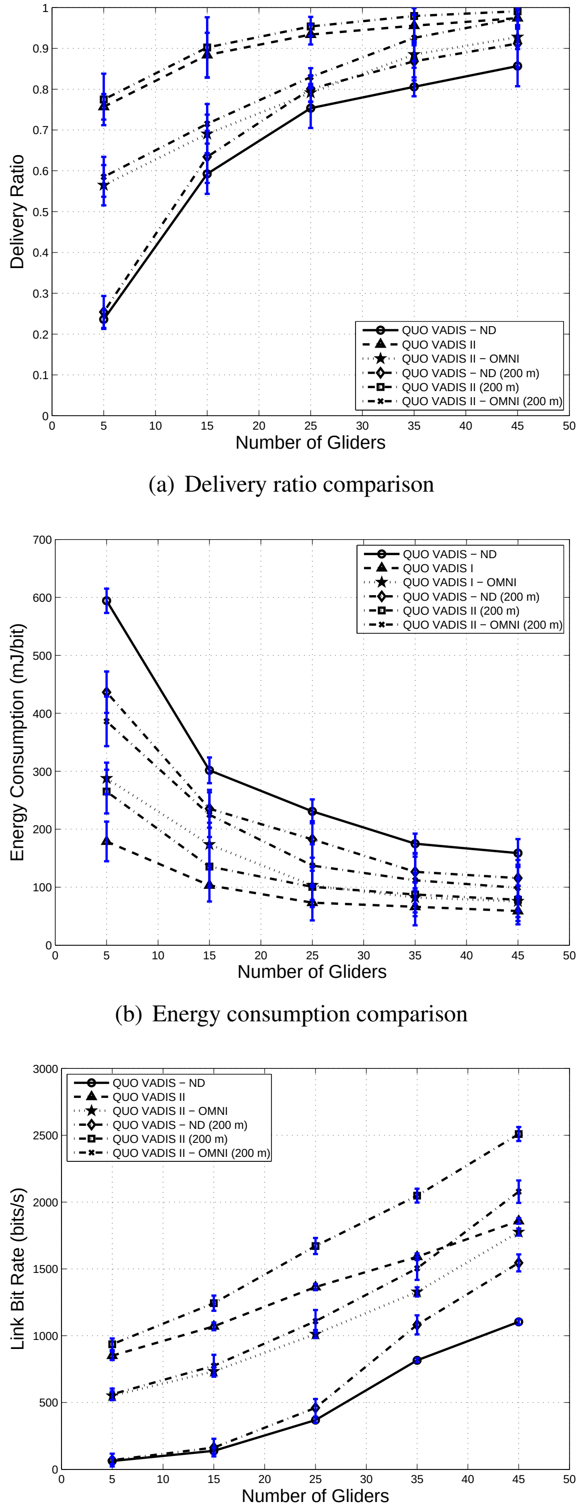


(a) Delivery ratio comparison

(b) Energy consumption comparison

(c) Link bit rate comparison

Figure 4.12: *Shallow water*: performance comparison for Class I traffic.

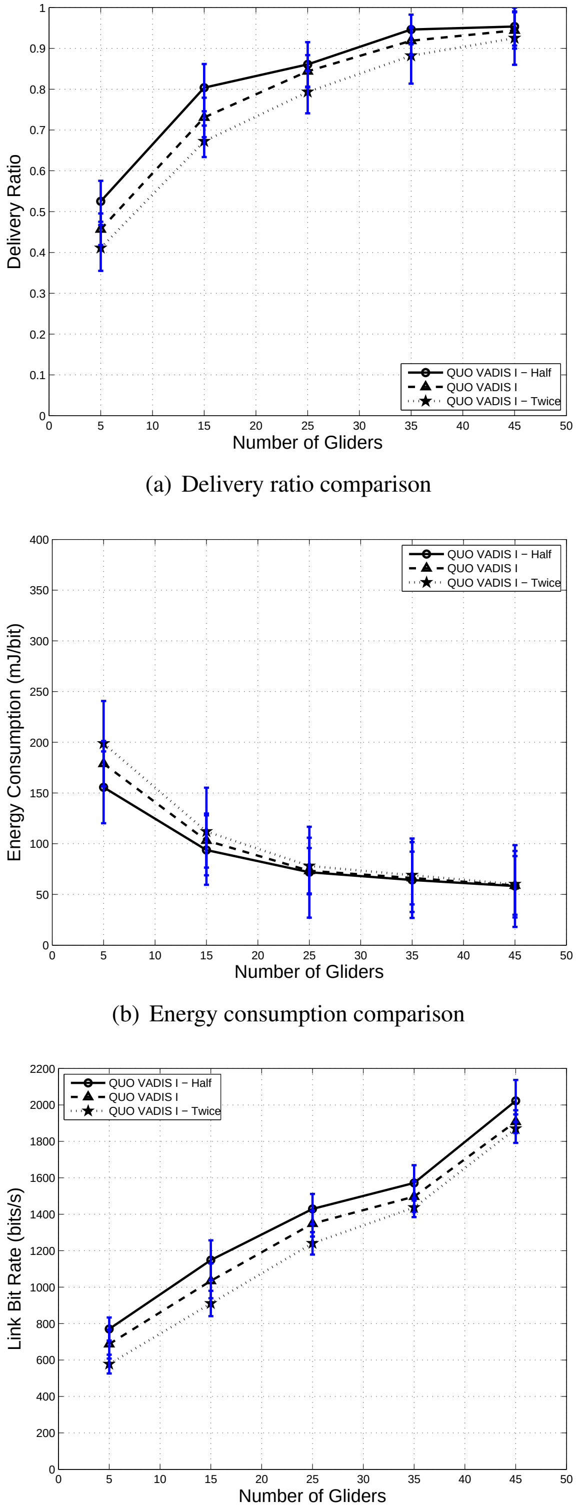


(a) Delivery ratio comparison

(b) Energy consumption comparison

(c) Link bit rate comparison

Figure 4.13: *Shallow water*: performance comparison for Class II traffic.

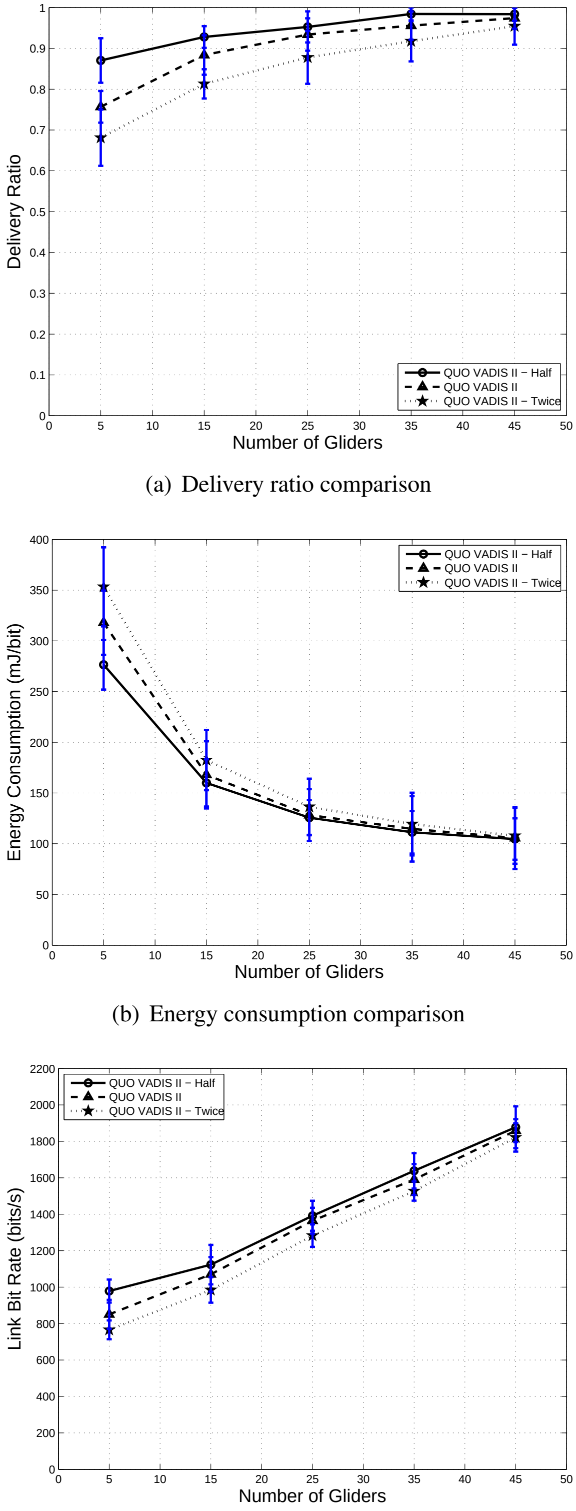


(a) Delivery ratio comparison

(b) Energy consumption comparison

(c) Link bit rate comparison

Figure 4.14: *Uncertainty update interval:* performance comparison for Class I traffic.



(a) Delivery ratio comparison

(b) Energy consumption comparison

(c) Link bit rate comparison

Figure 4.15: *Uncertainty update interval:* performance comparison for Class II traffic.

larger error in selection of neighbor for packet forwarding and estimation of transmission power. On the other hand, halving the interval leads to improvement of performance due to the uncertainty information is updated in a more timely manner (so routing error becomes smaller and transmission power is better estimated). However, this obviously leads to the overhead increase. Therefore the tradeoff between overhead and metrics such as delivery ratio, energy consumption and link bit rate should be carefully considered for different applications. Here we use “QUO VADIS - Half”, “QUO VADIS”, and “QUO VADIS - Twice” to denote the cases with update interval of 30 s, 60 s and 120 s, respectively.

To sum up, our proposed framework QUO VADIS improves the network performance for delay-tolerant applications in terms of e2e energy consumption, delivery ratio, and link bit rate by waiting for a ‘favorable’ topology configuration and by exploiting the gains of directional transducers. Through emulations for different setups, we demonstrated that they can offer better performance than the well-known geographic routing and DTN protocols when serving two classes of delay-tolerant traffic.

4.4 Geocasting Solution

Both versions of our proposed geocasting solution are implemented and tested via simulations. We are interested in evaluating the performance of our solution to see if it achieves our goal – maximizing the number of nodes receiving the geocasting packet in a given time. Our simulation is based on the Bellhop model. Simulation parameters are set as: number of nodes = 100, $\overrightarrow{\mathrm{v}} = (10, 0, 0)$ Km, $\mathbf{c} = (20, 0, 10)$ Km, $r = 5$ Km, $R = 2$ Km (denoted as “Setting I”). Nodes are uniformly distributed in the specified geocasting region with drifting model as in Sect. 4.2. The communication parameters are based on the specifications and measurements of the WHOI acoustic modem.

We compare the performance of our solution with two well-known geocasting solutions that were originally designed for terrestrial wireless networks, i.e., the Location-Based Multicast (LBM) [36] algorithm and GeoTORA [37]. In LBM, a node forwards packets to the geocasting region if it is within the *forwarding zone*, which is generally a region containing the geocasting region. If a node is in the geocasting region, it simply forwards the packets to all the neighbors. Outside the forwarding zone packets are discarded. Here we use the second scheme of LBM [36], where packets are forwarded when nodes are closer to the center of the geocasting region. GeoTORA is a geocasting solution based on the Temporally Ordered Routing Algorithm (TORA) [38], a unicasting algorithm for ad hoc networks. It maintains a single directed acyclic graph, where the directions are defined by assigning a height (the distance to the destination region) to each node. A packet is always forwarded to a neighbor with lower height. Nodes in the geocasting region are assigned height 0. Neither LBM nor GeoTORA consider the propagation delay.

We compare the performance of the two versions of our solution with LBM and GeoTORA in the following scenarios: i) source node located in the base of the cylinder region; ii) different radius sizes of the geocasting cylinder; iii) different node densities; and iv) source node located in the middle of the cylinder region surface. In order to study the pros and cons, we are interested in the percentage of nodes that received the geocasting packet at a given time. At the same time, we also want to measure the control overhead of each algorithm. Simulation results for these metrics are plotted in Figs. 4.16, 4.17, 4.18, and 4.19. The following is observed:

1) As shown in Figs. 4.16 and 4.18, our one-hop version solution performs the best, i.e., it takes the least time to geocast to all nodes within the region. However, this comes at the price of the largest overhead due to the need to exchange location information between neighbors (Figs. 4.17 and 4.19). Our no-hop version solution uses the second least time to finish geocasting the region. Due to the use of the NOTICE packet, the overhead it uses ranks the second among these four algorithms.

2) LBM algorithm performs the worst – using the largest amount of time to finish geocasting. This is because it simply floods the packet without coordination, leading to a large number of collisions. Therefore, retransmissions are needed, thus resulting in increased e2e delay. In this case, no control is needed to coordinate the nodes so the overhead is the lowest.

3) GeoTORA ranks the third among these four algorithms (Figs. 4.16 and 4.18). As it needs to use the TORA protocol to discover the geocasting routes, it waits the longest time before geocasting. Once the routes are discovered, the control overhead decreases as it is only needed when a route breaks (route maintenance). As GeoTORA does not rely on simple flooding, it has less collisions. Hence, its e2e geocasting delay is less than that for LBM. However, as propagation delay is not considered, it has more packet collisions than the two versions of our solution.

4) As shown in Fig. 4.18, it takes less time to finish geocasting from the middle of the cylinder

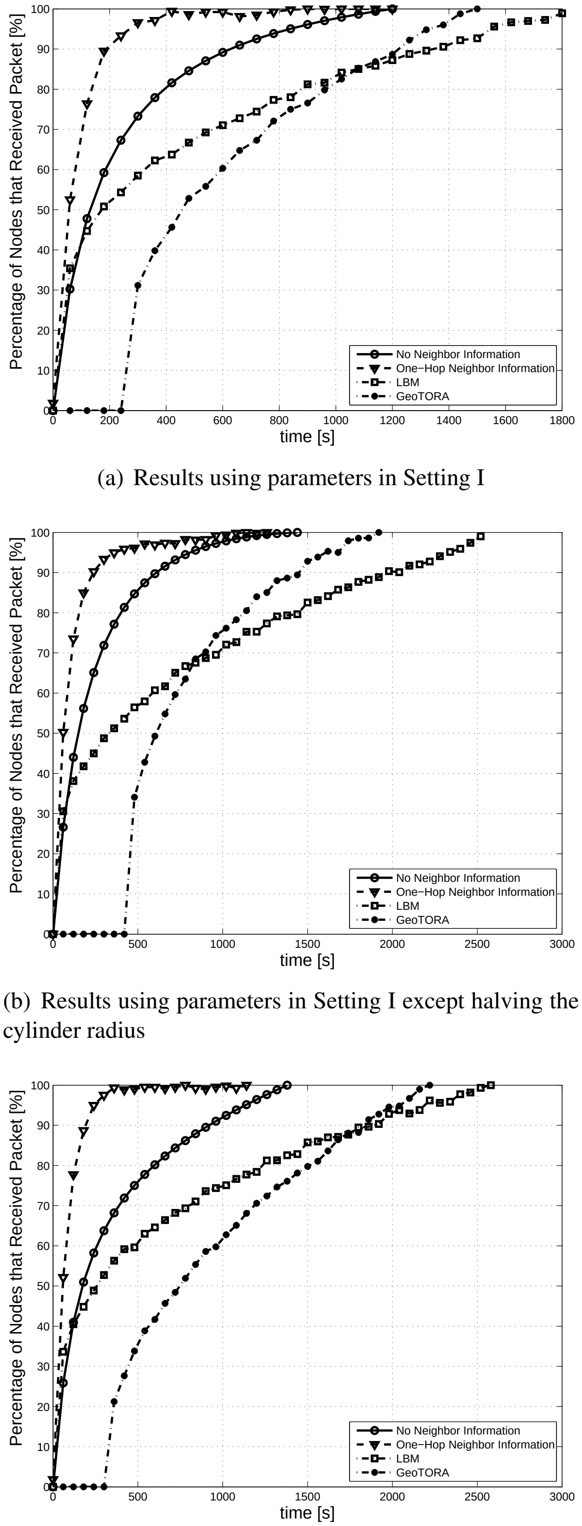
region than from the base of the cylinder region, which is obvious since geocasting can be done in both directions along the cylinder. This confirms the intuition that it is better that the geocasting begins from the middle of the geocasting region. It also gives a guideline for unicasting the geocasting packet from the surface station to the geocasting region.

5) By doubling the node density, the number of neighbors also double. Hence, the probability of packet collisions may increase, leading to longer geocasting finish time, as confirmed by the simulation results (Figs. 4.16(c) and 4.17(c)). Similar results can be observed by halving the cylinder radius (Figs. 4.16(b) and 4.17(b)). Interesting enough, the increase of geocasting finish time for both versions of our solution is much less than for LBM and GeoTORA. This is due to the selection of appropriate τs to de-synchronize transmissions. The increase of geocasting finish time for the one-hop version is less than for the no-hop version since τ is computed online for the one-hop version. Also, the control overhead for the one-hop version is relatively constant as nodes only need to broadcast location information periodically. Even though location information may be lost when the link is bad, nodes can use past information to predict the trajectory of a neighbor so the estimation of propagation delay is accurate. On the other hand, the no-hop version needs more retransmissions due to the lack of neighbor information, leading to increase in control overhead. Such situation is more severe in LBM and GeoTORA.

In sum, using more information from the neighborhood, nodes are able to schedule their packet transmissions in a better way so that collisions can be reduced or avoided, which leads to a higher e2e geocasting reliability. Moreover, our solution performs better than LBM and GeoTORA, two solutions originally designed for terrestrial wireless networks.

4.5 Under-ice Localization

The under-ice localization solution is implemented and tested on our underwater communication emulator. Our solution is compared against AAL, DNRL, and CNA, as introduced in Sect. 2.3, under an environment that is described by the Bellhop model [5]. We use the typical Arctic sound speed profile as in [79] and the corresponding Bellhop profile is plotted in Fig. 4.20. Note that we use 25 KHz, the sound frequency in use for our WHOI modem. We modify AAL, DNRL, and CNA, as they were originally designed for settings that are quite different from the under-ice environment. Specifically, AAL, DNRL and CNA all use the AUV that surfaces last as reference node because

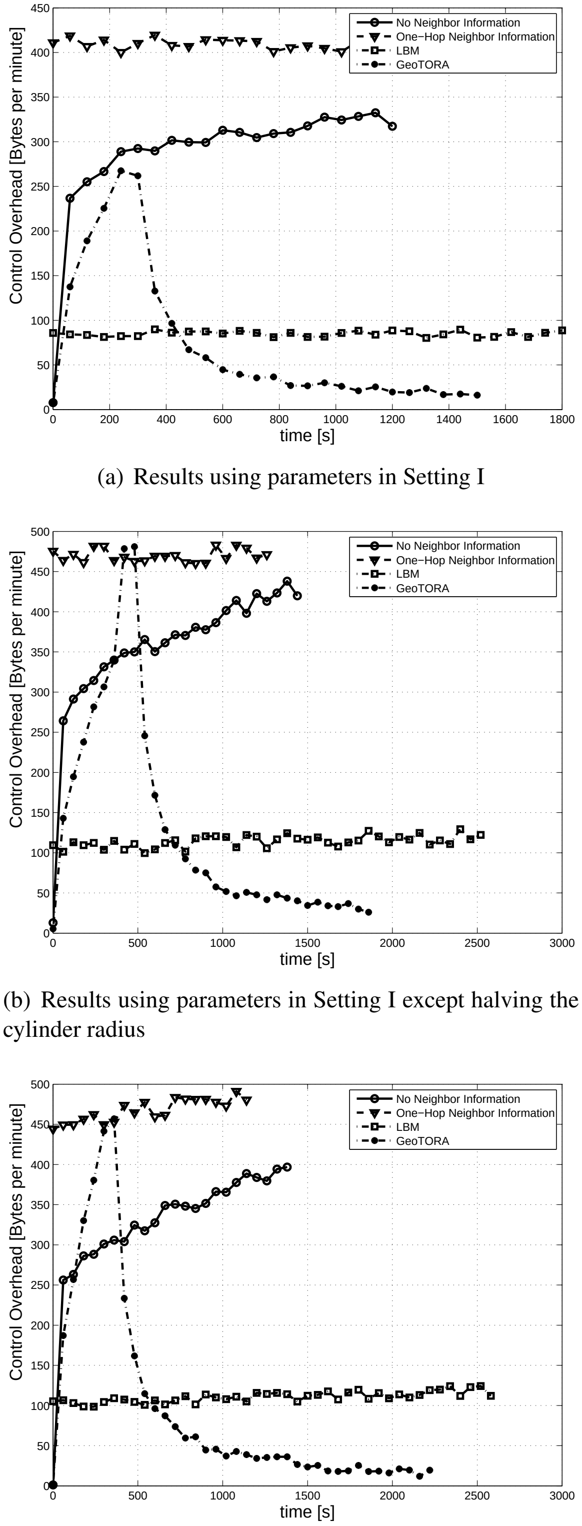


(a) Results using parameters in Setting I

(b) Results using parameters in Setting I except halving the cylinder radius

(c) Results using parameters in Setting I except doubling node density (i.e., doubling number of nodes)

Figure 4.16: Comparison of reliability.



(a) Results using parameters in Setting I

(b) Results using parameters in Setting I except halving the cylinder radius

(c) Results using parameters in Setting I except that node density becomes two times

Figure 4.17: Comparison of control overhead.

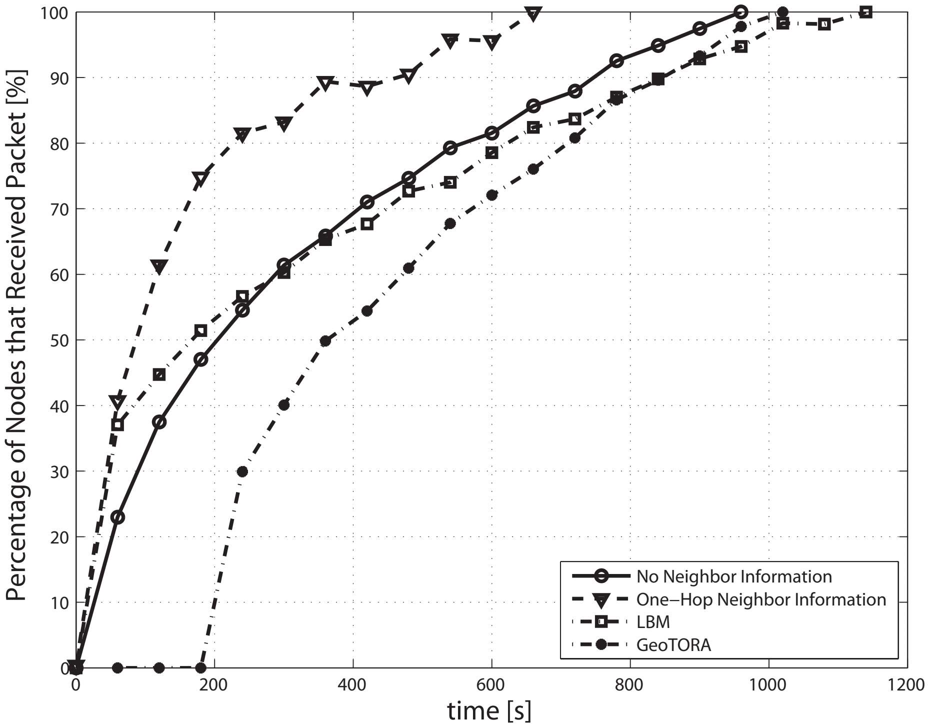


Figure 4.18: Geocasting from the middle of the cylinder surface.

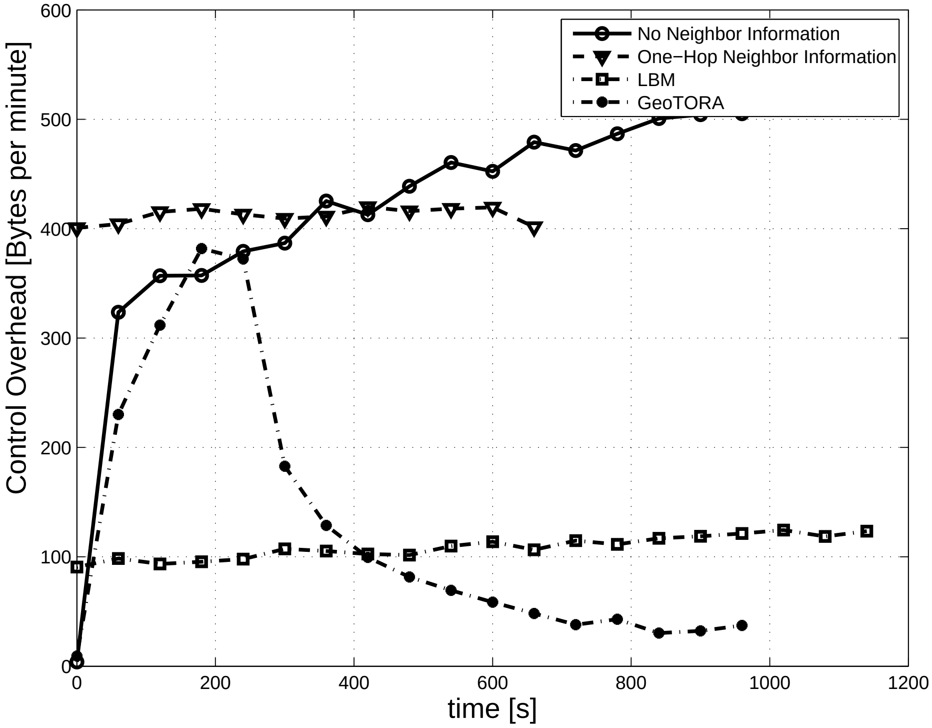


Figure 4.19: Control overhead (from the middle of the cylinder surface).

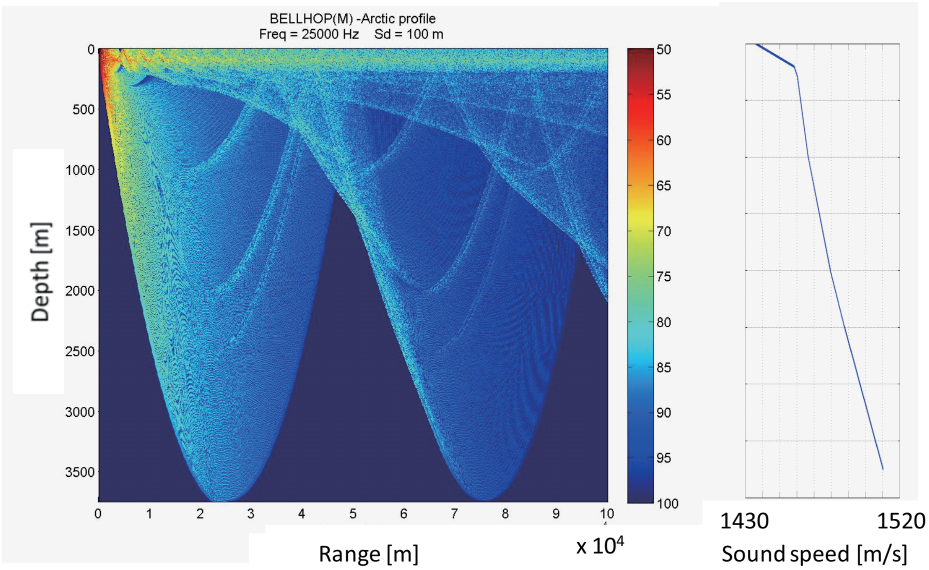


Figure 4.20: Bellhop profile for typical Arctic environment.

intuitively the shorter an AUV stays underwater (the less time it stays in an uncertain environment after a GPS fix), the less uncertain its position is. Triangulation is employed for position calculation in AAL and DNRL, while EKF filtering is used in CNA. We are also interested in seeing the performance improvement that we get using the external uncertainty notion. Therefore, we implement another version of our proposed localization solution without using external uncertainty, i.e., forcing the position uncertainty to be zero. We denote this modified version and the original version by 'Proposed solution w/o EU' and 'Proposed solution w/ EU', respectively (in this section, they denote the solutions for under-ice localization).

In order to evaluate the localization performance, two metrics, the *localization error* and the *deviation of error*, are used. Localization error is defined as the distance between the actual and the estimated AUV position. The deviation of error is the amount the localization error deviating from the total averaged error. The average localization error $\bar{E}$ and deviation of error σ are plotted. The formulae of $\bar{E}$ and σ are expressed as,

$$\bar{E} = \frac{1}{L_t} \sum_{j=1}^{L_t} \left(\frac{1}{N} \sum_{i=1}^{N} E_i \right), \sigma = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (E_i - \bar{E})^2}, \tag{4.1}$$

where N is the number of AUVs in the UW-ASN, E_i represents the localization error for each AUV operating in the UW-ASN at that particular time, and L_t is the number of times the localization is performed, such that $L_t = \frac{T_{end}}{\Delta T}$.

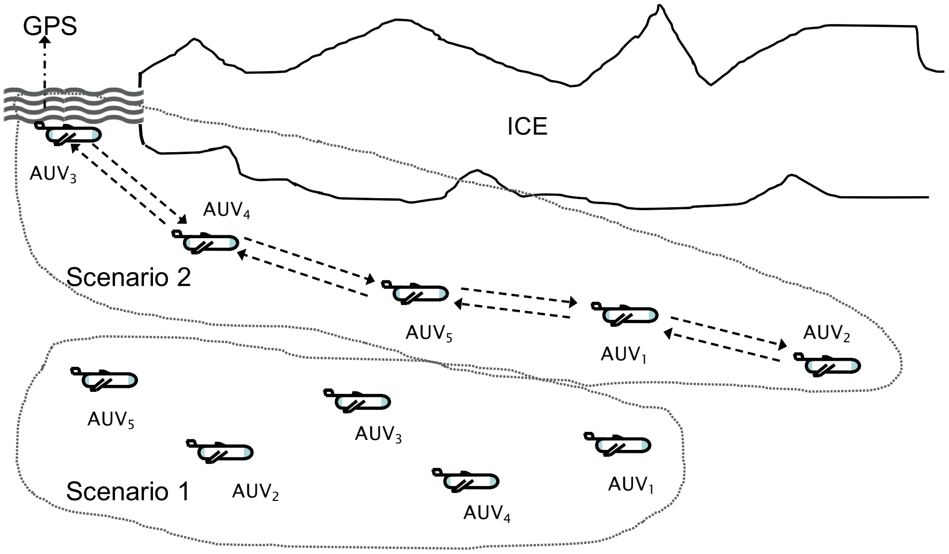
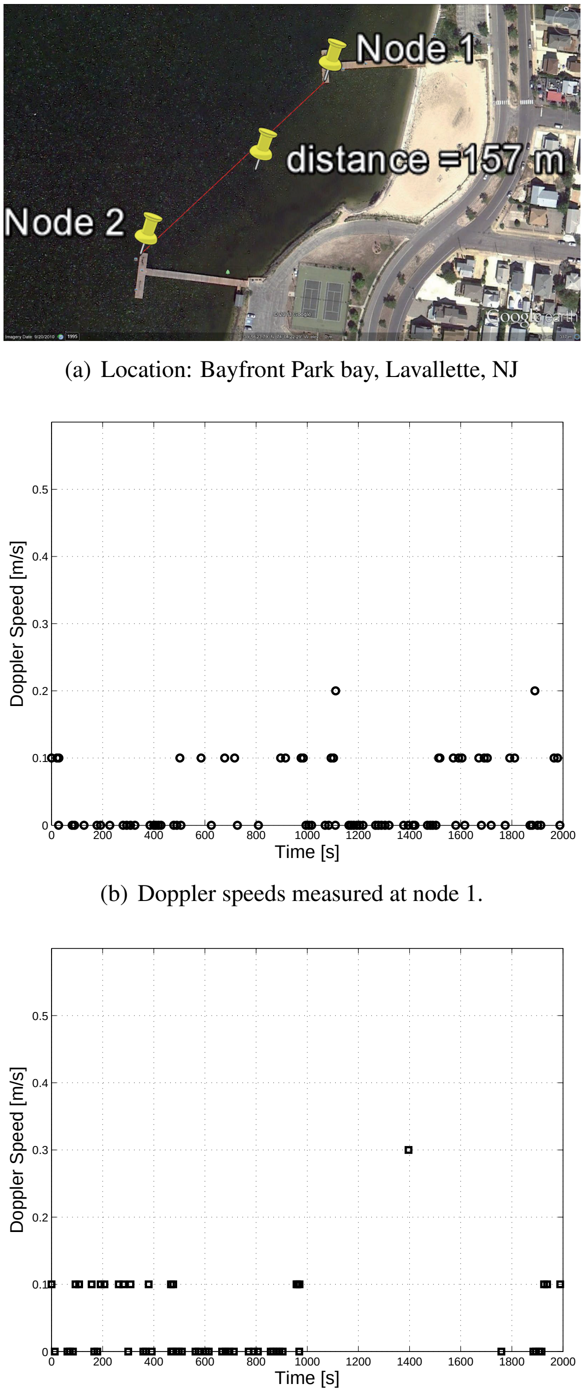


Figure 4.21: Two scenarios for simulations: different dotted circles represent different scenarios.

Table 4.3: Simulation Parameters for Evaluating Under-ice Localization Solution

Total Time	10600 s ($\sim$2.94 h)
Time Interval, ΔT	60 s
Deployment 3D Region	2000(L)$\times$2000(W)$\times$1000(H) m^3
Confidence Parameter, α	0.05
AUV Velocity	0.25-0.40 $\mathrm{m/s}$
AUV Depth Range	[0,1000] m
Typical Currents	0.01-0.03 $\mathrm{m/s}$ [80]
Extreme Currents	0.04-0.06 $\mathrm{m/s}$ [80]
Water Temperature Range	[-2,2] $^\circ\mathrm{C}$
Salinity Range	[32.5,35] ppt

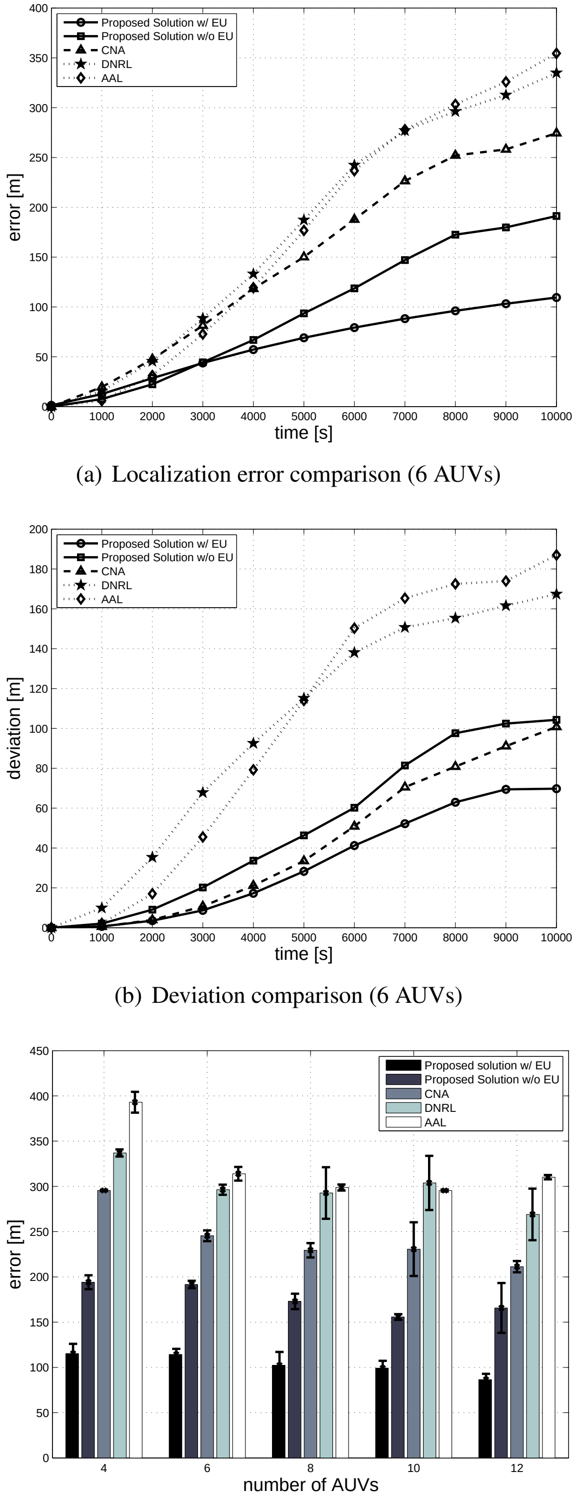


(a) Location: Bayfront Park bay, Lavallette, NJ

(b) Doppler speeds measured at node 1.

(c) Doppler speeds measured at node 2.

Figure 4.22: Doppler speed measurement. Only part of the measurements are plotted for clear visualization. Time coordinates vary due to different reception time.

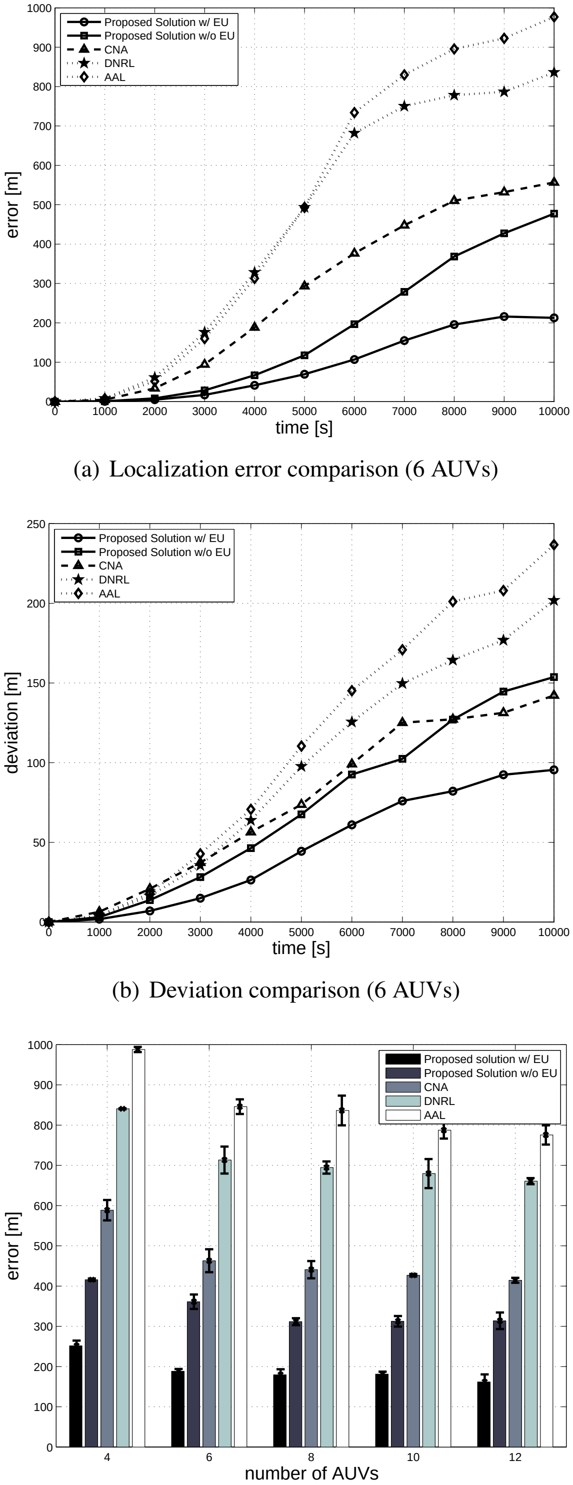


(a) Localization error comparison (6 AUVs)

(b) Deviation comparison (6 AUVs)

(c) Localization error for number of AUVs

Figure 4.23: Scenario 1 with Typical Currents: under the ice mission with no resurfacing.

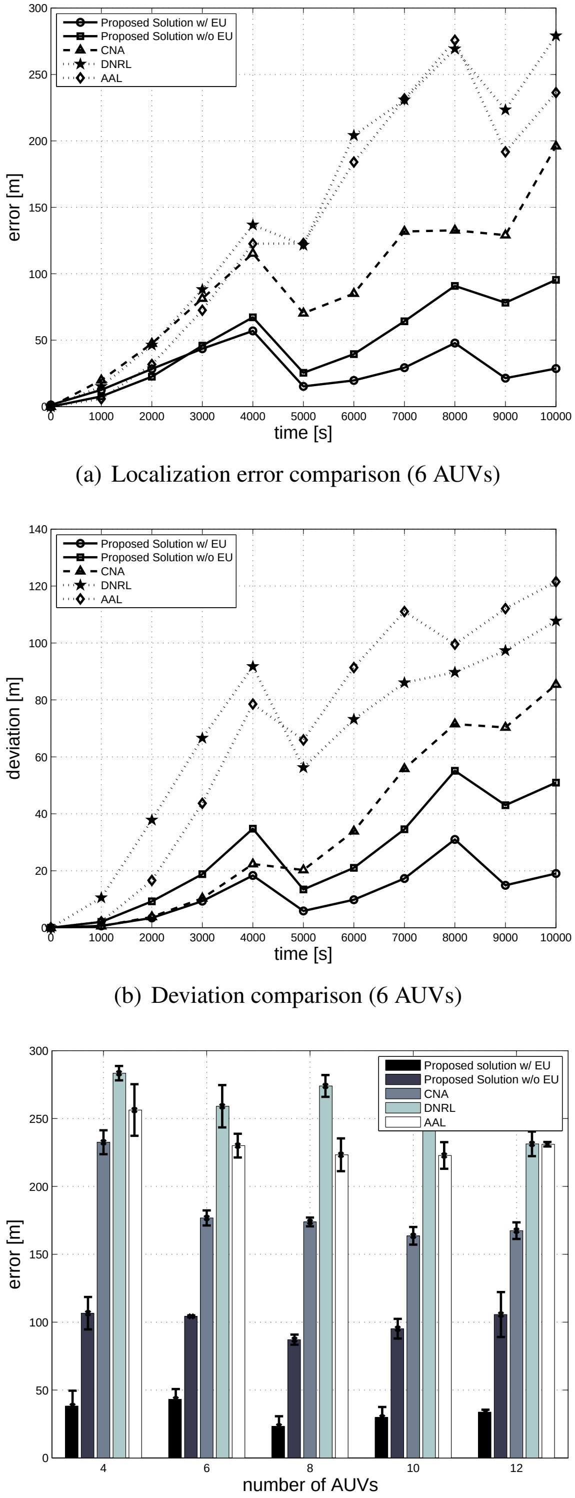


(a) Localization error comparison (6 AUVs)

(b) Deviation comparison (6 AUVs)

(c) Localization error for number of AUVs

Figure 4.24: Scenario 1 with Extreme Currents: under the ice mission with no resurfacing.

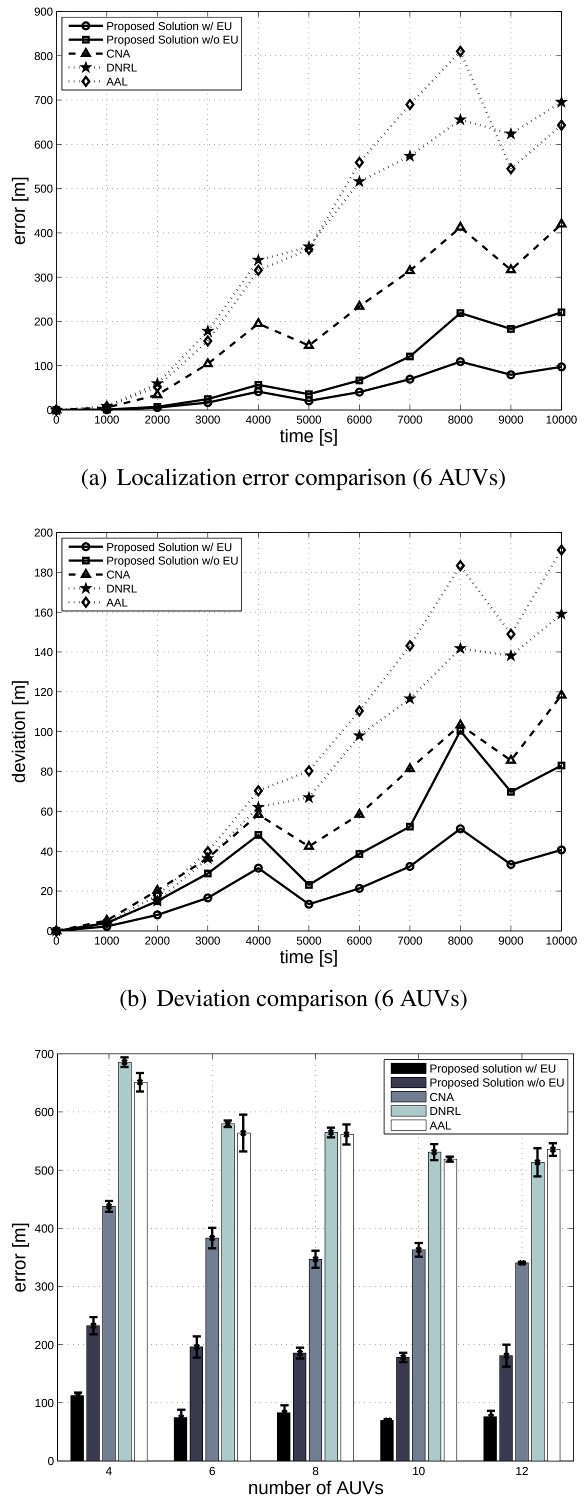


(a) Localization error comparison (6 AUVs)

(b) Deviation comparison (6 AUVs)

(c) Localization error for number of AUVs

Figure 4.25: Scenario 2 with Typical Currents: under the ice mission with resurfacing.



(a) Localization error comparison (6 AUVs)

(b) Deviation comparison (6 AUVs)

(c) Localization error for number of AUVs

Figure 4.26: Scenario 2 with Extreme Currents: under the ice mission with resurfacing.

4.5.1 Simulation Scenarios

The parameters for our simulations are listed in Table 4.3. We further assume that ongoing communication packets are generated according to the Poisson traffic model with arrival rate being 3 packets per minute. As shown in Fig. 4.21, we utilize the following two specific scenarios.

Scenario 1: This scenario involves a team of AUVs who collaboratively explore an underwater region located under ice. These AUVs remain under-ice for the duration of the mission and do not return to the surface until the mission is completed.

Scenario 2: This scenario is similar to the first except that individual AUVs will periodically surface to update their positioning via GPS. These AUVs take turns to go back to the surface at a predefined interval, which is 4000 s in our simulations. In order to avoid ice cover, these AUVs return to the edge of the ice sheet where they were deployed. Once an AUV surfaces, it acquires a GPS fix and updates its current coordinate position (position uncertainty is also reset).

Both scenarios are tested with typical and extreme currents, whose speed ranges are listed in Table 4.3. A random 3D direction is chosen for the current throughout one round of simulation. The Doppler data is based on the 6-hour Doppler speed measurement that we took using WHOI modems on November 20th, 2011 in the Bayfront Park bay, Lavallette, NJ, as shown in Fig. 4.22. Our measurement shows that most of the Doppler speeds are low, similar to the part we plot here. Note that the right hand side in (3.49) is replaced with the measured Doppler speed here as there is no need to calculate the Doppler shifts.

4.5.2 Evaluation Results

Real time (one simulation run) localization errors and deviations of error are plotted in the first two subfigures of Figs. 4.23-4.26. Moreover, to obtain results of statistical significance, 250 rounds were conducted for varying numbers of AUVs. The average errors for the AUV's predicted location are plotted in Figs. (4.23-4.26)(c) with 95% confidence intervals.

Scenario 1: As shown in Figs. 4.23 and 4.24, our original solution 'Proposed solution w/ EU' performs the best. In the typical current setting, 'Proposed solution w/ EU' obtains about 74.6% less error than 'Proposed solution w/o EU' while it obtains 80.4% less error in the extreme current setting. This is mainly due to the use of the external uncertainty model to predict the position and distribution of the AUVs and the ability to minimize the localization uncertainty. 'Proposed solution w/o EU' ranks the second in terms of error performance because an AUV can leverage the ongoing communications and cooperation of other AUVs for localization. Even though CNA uses EKF to predict the positions, its performance is worse than 'Proposed solution w/o EU' since the AUV can only use its own states for position estimation. On the other hand, CNA is still better than DNRL and AAL due to the use of EKF filter, and DNRL performs better than AAL since it takes the current influence into account.

By comparing Figs. 4.23 and 4.24, we can see that under extreme conditions, the localization error keeps increasing, since more dislocation is incurred by the extreme currents. Interestingly enough, we can see that the performance of our solution without using external uncertainty is not much better than that using CNA. In this case, using Doppler information does not help improve the localization much since the position uncertainties associated with other AUVs are also large and thus the performance is not too much better than that of using EKF. However, our solution using external uncertainty still performs the best due to the ability to estimate the position uncertainty and then use such information to minimize uncertainty.

Scenario 2: As shown in Figs. 4.25 and 4.26, the performance ranking for these solutions closely resembles that in Scenario 1. However, the localization performance in Scenario 2 is much better than that in Scenario 1 since AUVs can obtain position correction periodically, as seen by comparing Figs. 4.23 with 4.25 (or Figs. 4.24 with 4.26). From these figures, we can see that localization error and deviation decrease when AUVs surface, i.e., at 4000 s and 8000 s in the results. Moreover, we can see that for typical current settings in Scenario 2, the localization error and its deviation can stay within certain threshold for 'Proposed Solution w/ EU', while the error of other solutions tends to increase. This shows the effectiveness of our proposed solution in minimizing the localization uncertainty.

Communication Overhead: Last, we compare the communication overhead of our solutions against other solutions. As shown in Fig. 4.27, 'Proposed solution w/o EU' achieves less overhead than CNA, DNRL and AAL due to the ability to exploit the Doppler shifts of ongoing communications for localization, reducing the use of ranging packets. 'Proposed solution w/ EU' has the biggest communication overhead in the beginning because of the need to broadcast external uncertainty information (such as pdf information). However, due to the ability to optimize the update

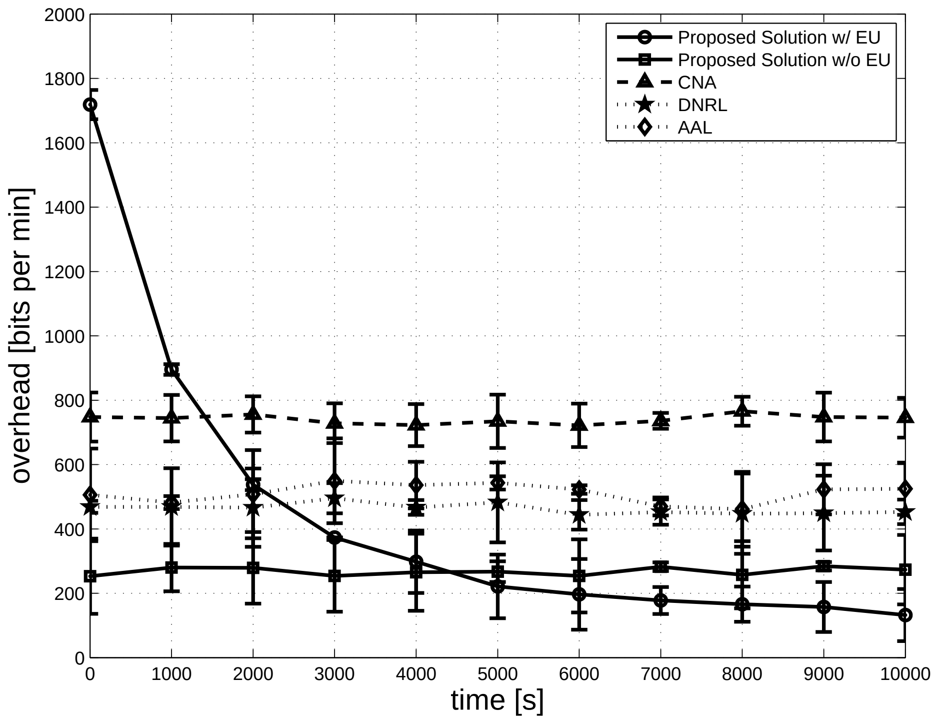


Figure 4.27: Comparison of communication overhead.

intervals T_s and T_p as in Sect. 3.6.3, its communication overhead drops quickly to a level that is lower than CNA, DNRL and AAL. CNA has higher overhead than DNRL and AAL as CNA needs to broadcast additional information such as velocities and sensor readings for EKF while DNRL and AAL only need to broadcast the position and time information that is embedded in the ranging packet. Note that in 'Proposed solution w/ EU', to save the overhead, when the AUVs broadcast the pdf information, they only broadcast the key parameters if the pdf is one of the well-known distributions (e.g., the average and standard deviation for a normal distribution). Otherwise, the point mass function of a finite number of points is broadcast.

4.6 Team Formation and Team Steering

In this section, we first outline the objectives of our emulation and its setup for the AUV coordination solution; then, we discuss the results for representative scenarios.

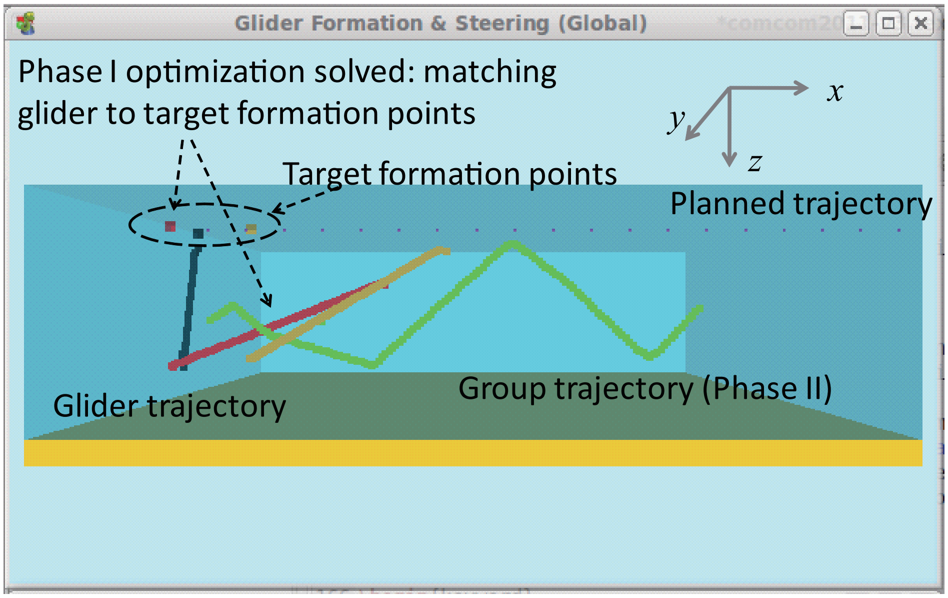

Glider Formation & Steering (Global)
Phase I optimization solved: matching glider to target formation points
Target formation points
Planned trajectory
x
y
z
Glider trajectory
Group trajectory (Phase II)

Figure 4.28: Screenshot of the 3D visualization of our solution: global view of a team with 3 gliders.

4.6.1 Emulation Overview and Setup

We are interested in comparing the performance of our coordination algorithms (which use underwater acoustic communications) in terms of coordination errors with the solutions without using underwater communications or coordination algorithms. Specifically, our solution is compared with the following two solutions. The first one is the solution using satellite to exchange coordination information instead of using acoustic communications. In this solution, all gliders surface for satellite communications every 2 hours, while underwater they do not exchange coordination information. Once they have exchanged the control information, they use the AFA algorithm to set their steering angles and then keep steering with the calculated angles until the next surface time. The second one is the solution where gliders do not coordinate at all. Each glider just steers itself to the destination without exchanging coordination information with other gliders. For convenience, in the following figures, we denote our proposed solution that uses acoustic communications, the solution using only satellite communications, and the solution without coordination as "Acoustic," "Satellite," and "No Coordination," respectively.

The team formation and steering solution is implemented and tested on our hybrid testbed in Sect. 4.1. We are interested in the performance of different solutions in the presence of ocean currents. A 3D visualization demo is also made during the implementation of our solution as shown

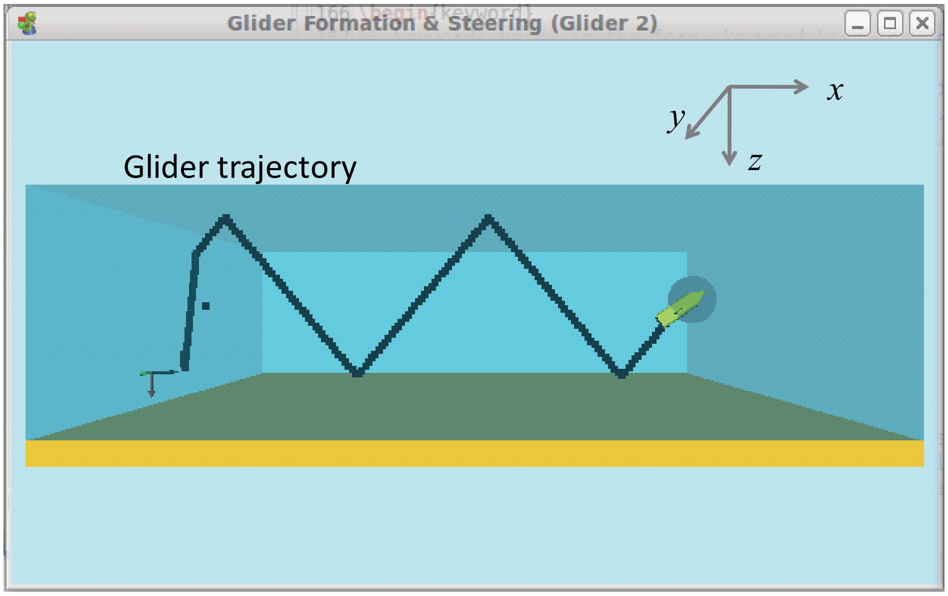


Figure 4.29: Screenshot of the 3D visualization of our solution: individual view of glider 2. The transparent sphere (halo) at the glider head means that it is team leader.

Table 4.4: Emulation Parameters for Evaluating the Team Formation and Steering Algorithms

Parameter	**Value**
Initial deployment region	3000(L)×3000(W)×500(H)m^3
Interval Δ	30 s
Transmission Power	10 W
Glider Horizontal Speed (Relative)	0.3 m/s
Gliding Depth Range	[0, 500] m
Pitch Angle Range	[10°, 35°]
Trajectory Length	8000 m

in Figs. 4.28 and 4.29, so that the movement and trajectories of the gliders can be visualized. More details about the demo can be found at [81].

We simulated these solutions considering the following different ocean current profiles. 1) **Current Profile 1**: current along x direction with constant velocity; 2) **Current Profile 2**: current along y direction with constant velocity; and 3) **Current Profile 3**: ocean gyre current model as in Fig. 4.30, i.e., a circular eddy with counter-clockwise tangential velocity profile $s_H \cdot r \cdot \exp(-2r^2)$, where r is the ratio of the distance from a point to gyre center to a reference distance D [1]. We assume the current profiles are vertically constant, i.e., the current velocities are the same if the horizontal locations are the same. More realistic ocean models will be studied and solutions dealing with these models will be proposed in our future work.

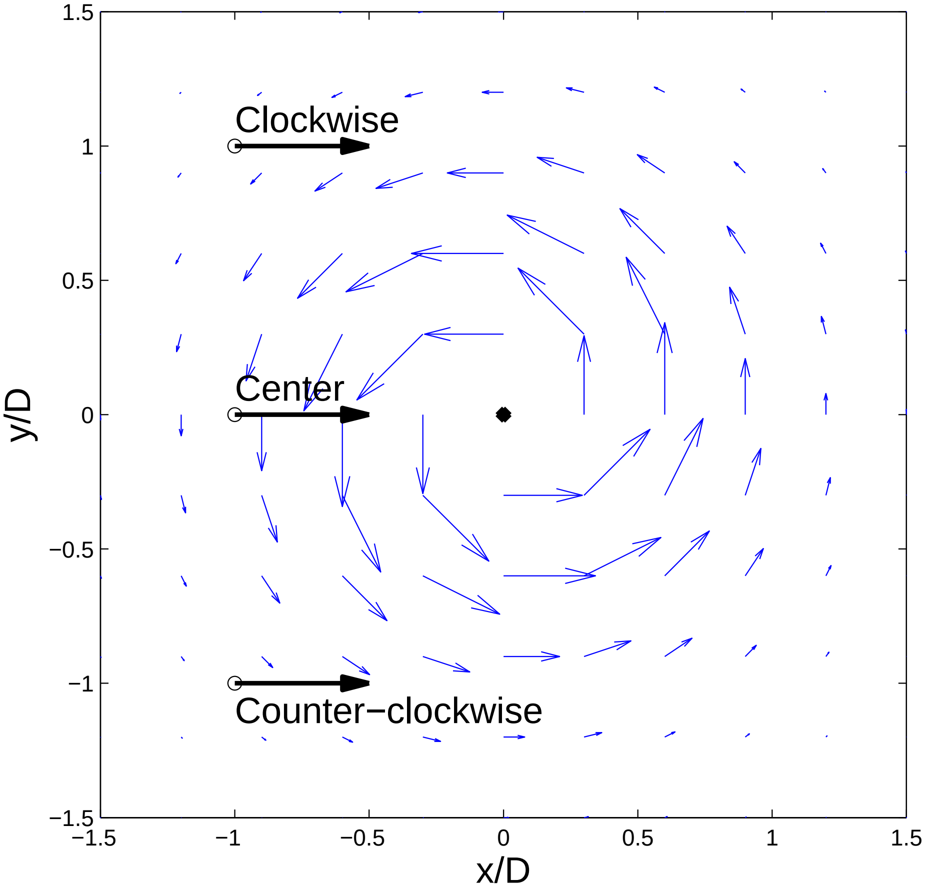
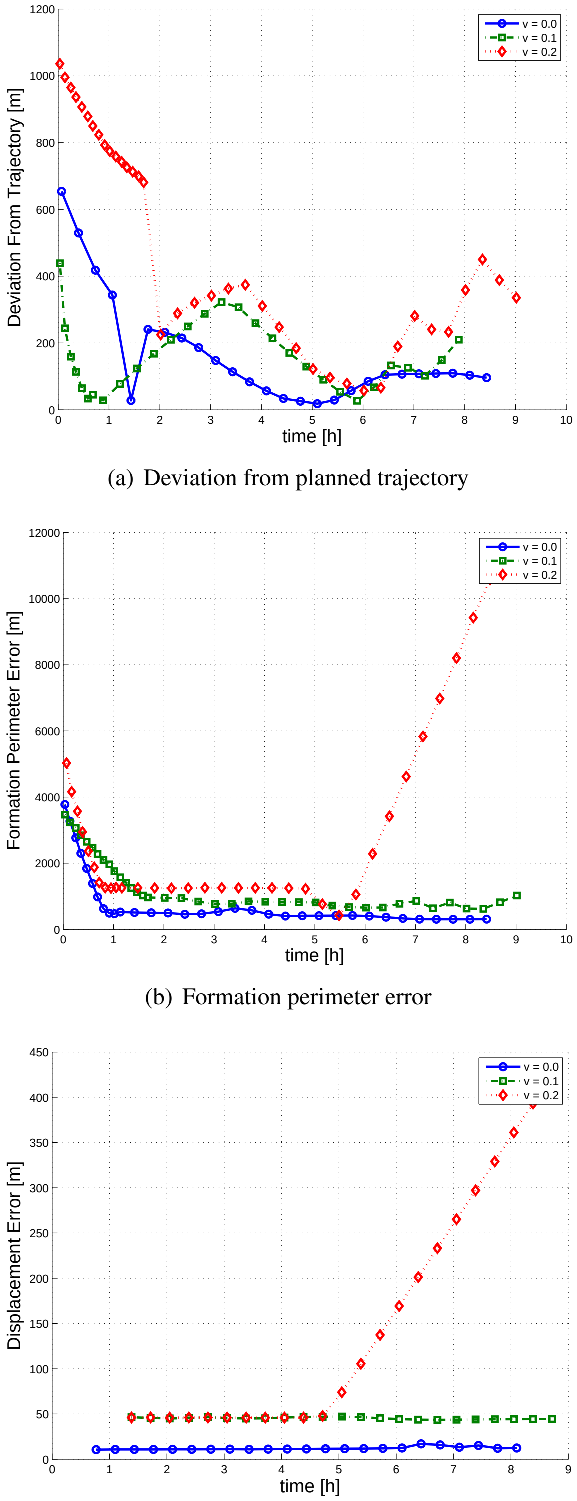


Figure 4.30: Ocean current profile [1]. Each thin arrow indicates ocean current speed and direction at that point, while each thick arrow indicates the starting position and direction of the planned trajectory in Sect. 4.6.4. The position is relative to the gyre center and scales relative to reference distance D.

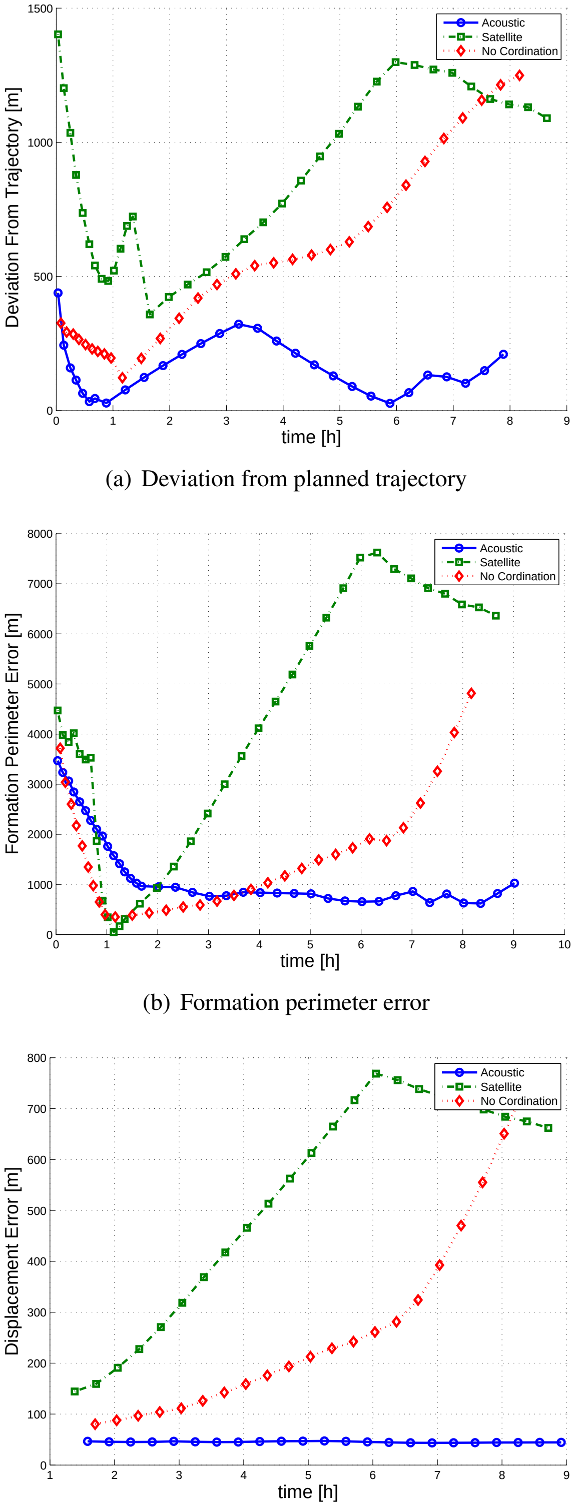


(a) Deviation from planned trajectory

(b) Formation perimeter error

(c) Displacement error during steering

Figure 4.31: Performance of the proposed solution for ocean current profile 1 (speeds are in $\mathrm{m/s}$).

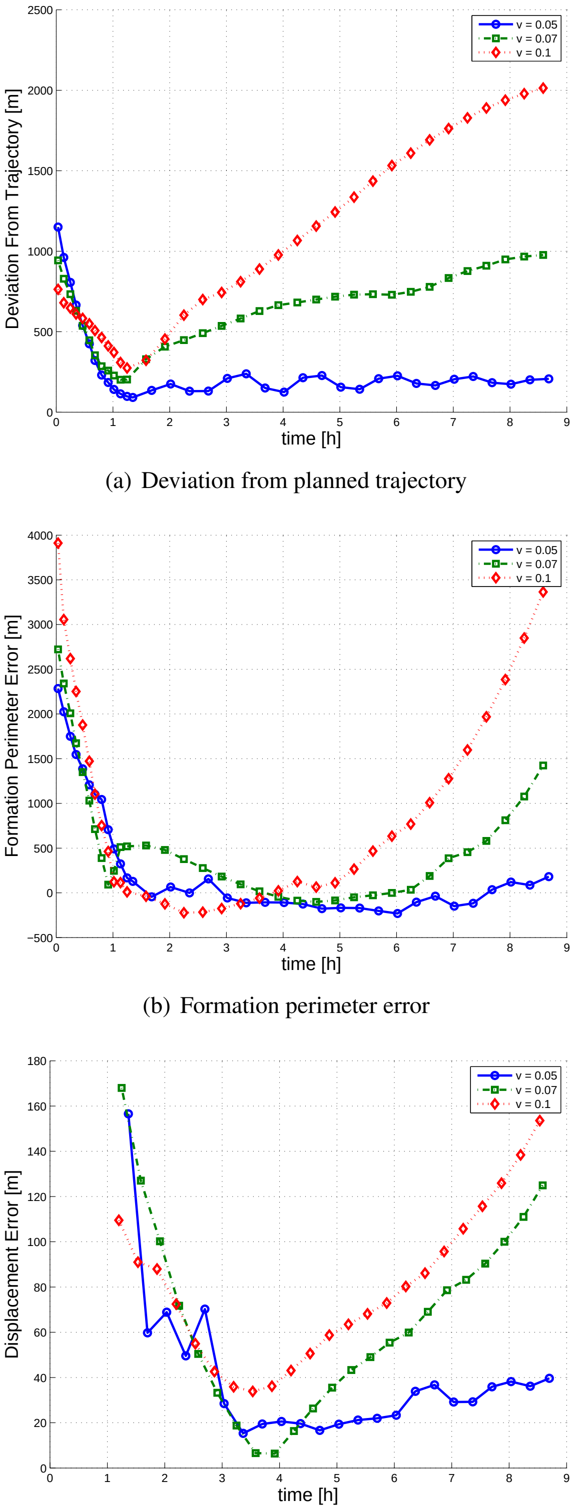


(a) Deviation from planned trajectory

(b) Formation perimeter error

(c) Displacement error during steering

Figure 4.32: Performance comparison of different solutions for ocean current profile 1 at $v = 0.1\mathrm{m/s}$.

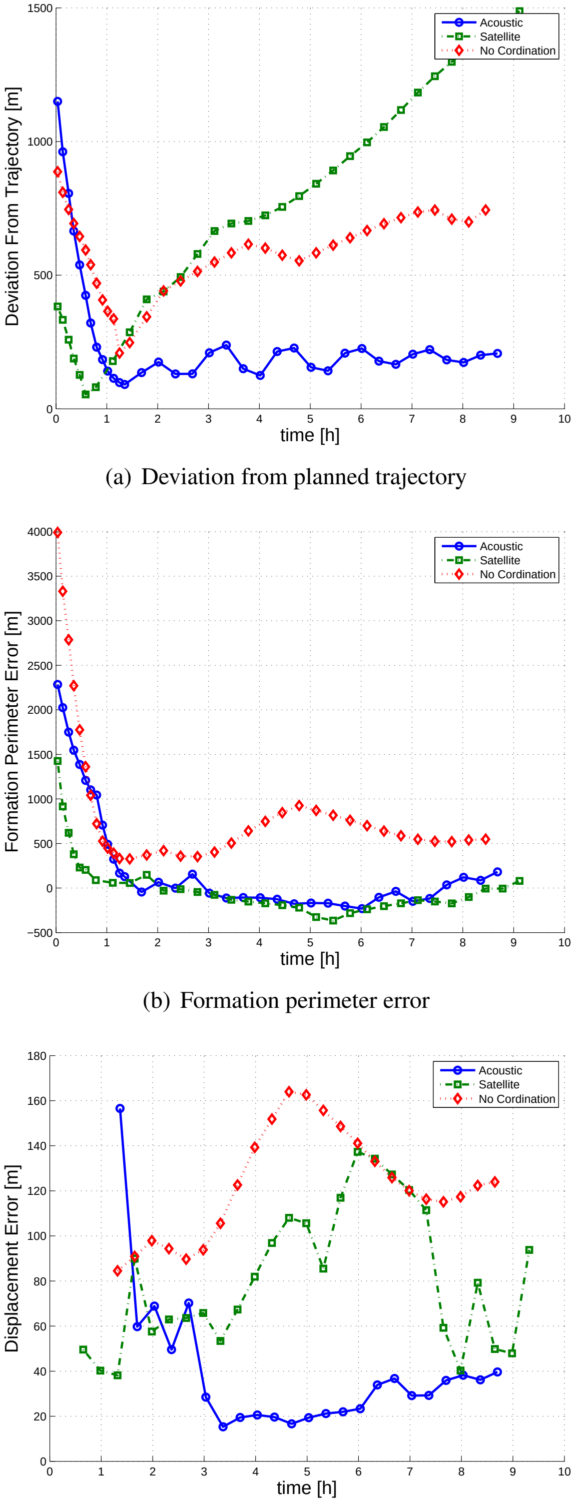


(a) Deviation from planned trajectory

(b) Formation perimeter error

(c) Displacement error during steering

Figure 4.33: Performance for ocean current profile 2 (speeds are in $\mathrm{m/s}$).



(a) Deviation from planned trajectory

(b) Formation perimeter error

(c) Displacement error during steering

Figure 4.34: Performance comparison of different solutions for ocean current profile 2 at $v = 0.05\mathrm{m/s}$.

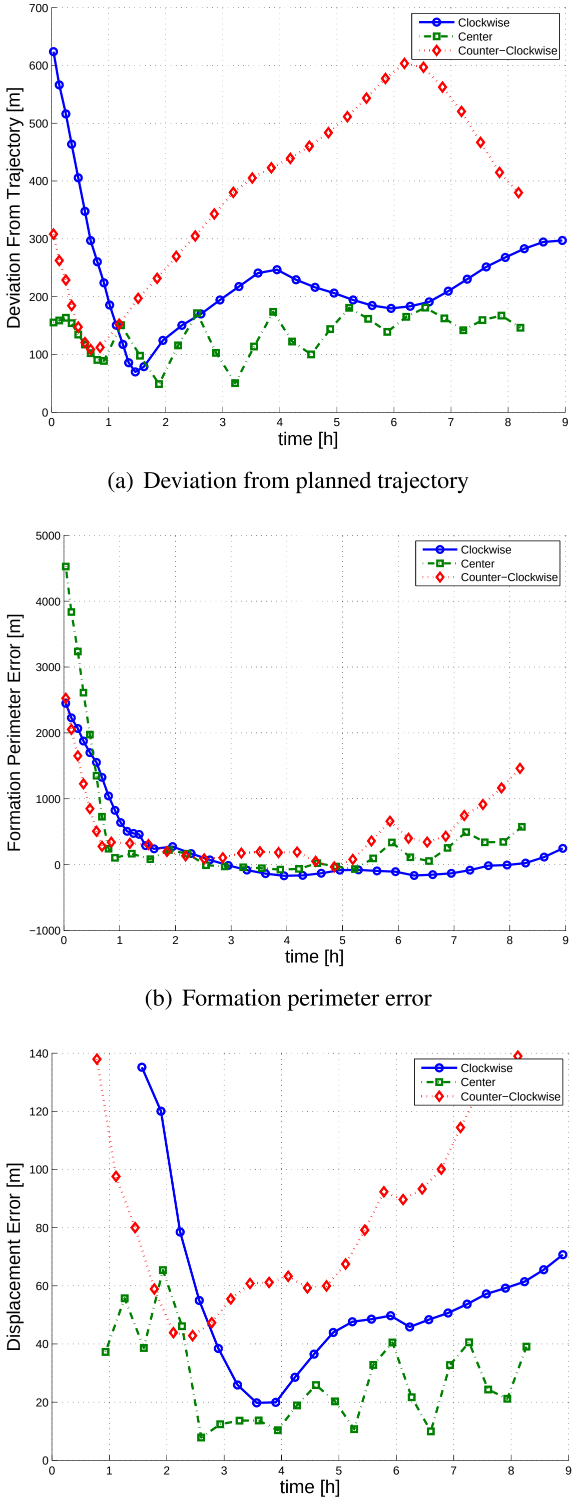
Emulation parameters are listed in Table 4.4. The direction of the planned trajectory is *along the x-axis direction*. In the beginning, gliders are randomly deployed in an initial 3D region. They are expected to form an equilateral triangle with inter-glider distance of $400\ \mathrm{m}$ and then steer along the planned trajectory.

We use the following three metrics to evaluate and compare performance. 1) **Deviation From Trajectory (DFT)**: the distance from the centroid of the team to the planned trajectory. This is used to characterize how well the algorithms work to keep the glider team on the planned trajectory; 2) **Formation Perimeter Error (FPE)**: the difference between the actual formation perimeter and the perimeter of target formation geometry. This offers a way to estimate the distortion of the whole team's formation; and 3) **Displacement Error (DE)**: the average displacement distance of each glider from its expected location, i.e., the average distance from one glider's actual position to its expected position. This metric quantifies how well the gliders can maintain the expected formation.

Emulations are run and the above metrics vs. the time from when the gliders are deployed are plotted in Figs. 4.31-4.36, which are discussed in the following. Note that different solutions and different ocean current models have different finish times. This is because different solutions or current models lead to different movement trajectories and, thus, different finish times. We stop the emulation when it is clear that getting to the end point of the planned trajectory is impossible and plot the metrics in intervals where another solution succeeds. Last, but not least, note that the plot of DE starts from Phase II as DE in Phase I is meaningless.

4.6.2 Performance using Ocean Current Profile 1

In this case, the performance of our proposed solution is plotted in Fig. 4.31 for different velocities v. As the ocean current speed increases, DFT, FPE and DE all increase, which is not difficult to understand. After all, the greater the current, the harder for the team is to stay in the expected position. From this figure, we can see that when the ocean speed is at $0\ \mathrm{m/s}$ and $0.1\ \mathrm{m/s}$, the glider team using our proposed solution can stay close to the planned trajectory. Note that even at $v = 0$, these error metrics are not perfectly zero. This is because the physical constraints of the glider (such as sawtooth movement and pitch angle range) make it impossible to achieve perfect coordination. At a speed of $0.2\ \mathrm{m/s}$, increase rate of FPE and DE becomes large when time is after about 5 or 6 hours. This is because the team is pushed over the target trajectory end point by the strong current,



(a) Deviation from planned trajectory

(b) Formation perimeter error

(c) Displacement error during steering

Figure 4.35: Performance for ocean current profile 3.

which is difficult to compensate for.

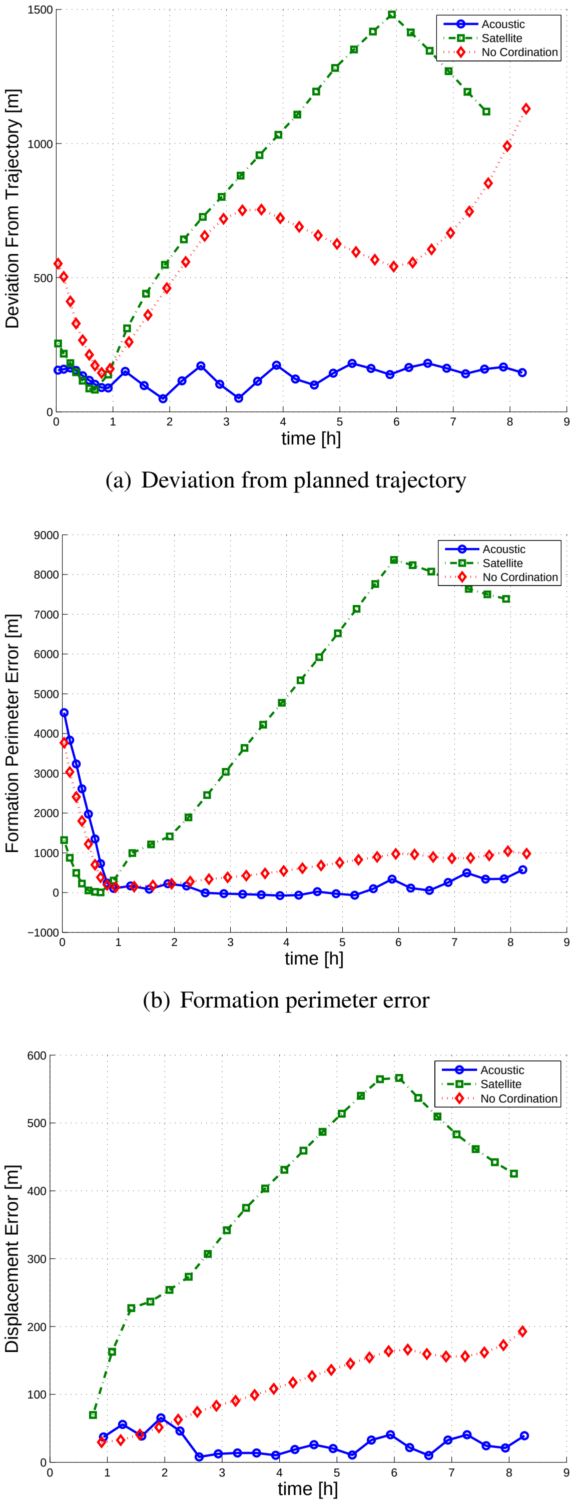
As shown in Fig. 4.32, when compared to the other two solutions, our proposed solution achieves lower errors in terms of DFT, FPE, and DE. By exchanging the position information and extracting Doppler shifts from ongoing communications, our algorithms can adjust the gliding angles for error minimization in a timely manner. On the other hand, the “Satellite” solution only adjusts the angles of the gliders when they surface; hence, the error keeps accumulating during the long intervals between surfacing. Steering error is adjusted only when the gliders surface. For the “No Coordination” solution, though it has less error than the “Satellite” solution in the beginning, the error keeps increasing without a way to decrease it due to no coordination. In the end, it accumulates more error than the “Satellite” solution.

4.6.3 Performance using Ocean Current Profile 2

In this case, the direction of the ocean current is perpendicular to the direction of the planned trajectory. Such an ocean current pushes the gliders sideways and, therefore, away from the planned trajectory. As shown in Fig. 4.33, when the current speed is at 0.05 $\mathrm{m/s}$, the proposed solution is able to keep DFT, FPE, and DE within a certain threshold. This verifies the effectiveness of our solution for team steering. As shown in Fig. 4.34, our solutions leads to the least error in terms of DFT, FPE, and DE among the three solutions. In fact, through extensive emulations we found that this is the maximum speed for which our solution is still effective.

4.6.4 Performance using Ocean Current Profile 3

Depending on the relative position of the team in the gyre current model, the performance of our proposed solution varies. As shown in Fig. 4.30, three cases, “counter-clockwise,” “center,” and “clockwise” are simulated for $D = 4000\mathrm{m}$. If the glider team moves counter-clockwisely around the gyre center (corresponding to “counter-clockwise” in Fig. 4.35), the vehicle and the current speeds add up together, leading to fast accumulation of error. Therefore, in this case the performance of our solution is worse than that in the “clockwise” case and that in the “center” case. In the “center” case, the team moves through the gyre via the center, where currents on opposite sides of the center tend to cancel the error as they move in opposite direction. Interesting enough, in Fig. 4.36 the performance of our solution is not much better than that of the “No Coordination”,



(a) Deviation from planned trajectory

(b) Formation perimeter error

(c) Displacement error during steering

Figure 4.36: Performance comparison of different solutions for ocean current profile 3 (“center” case).

which is because varying current speed makes it more difficult to coordinate.

In sum, our proposed coordination solution is effective when the ocean current speed is within a certain threshold (which depends on the current model). Compared to the other two solutions, our solution leads to lower formation and steering errors.

Chapter 5

Conclusions and Future Work

Position information is of vital importance in mobile underwater sensor networks. However, factors such as drifting and localization errors cause uncertainty when estimating an AUV's trajectory. In this dissertation we first offered the distinction between two forms of position uncertainty - the internal and external uncertainty - and then we proposed a statistical approach to estimate an underwater glider's internal uncertainty. By using the previous estimated positions, a glider is able to estimate the uncertainty associated with itself (internal uncertainty). This internal uncertainty is then forwarded to other vehicles, and used as the base to estimate the external uncertainty. We also extended internal- and external-uncertainty estimation to the general type of AUVs whose trajectories are predictable.

Based on this uncertainty model, we proposed the following communication solutions.

1) A QoS-aware communication optimization framework called QUO VADIS, which provides energy-efficient communications among AUVs for delay-tolerant applications. By leveraging the predictability of AUV trajectories, QUO VADIS waits for a favorable network topology and hence it can optimize communications by delaying packet transmissions. QUO VADIS also exploits the frequency-dependent radiation pattern of underwater acoustic transducers to reduce communication energy consumption by adjusting the transducer directivity on-the-fly. QUO VADIS can serve two-classes of delay-tolerant traffic, which is a complement of current solutions for underwater networks. Performance results showed that QUO VADIS is effective in serving two different classes of delay-tolerant traffic. Compared with existing well-known geographic routing solutions and DTN solutions, our proposed solution has lower transmission energy consumption and higher data rates. Communication energy consumption is further reduced by exploiting the frequency-dependent radiation pattern of underwater acoustic transducers.

2) A reliable geocasting solution, which can be used to efficiently mutlicast data to AUVs in a specified geographic region with minimal amount of time. We proposed two versions of geocasting solution for AUVs based on different degrees of neighbor information whose objective is to reach the highest number of nodes within a pre-defined directional 3D region in a given amount of time when the positions of the nodes are uncertain. Based on the external uncertainty estimates, packets are forwarded along the path that can reach the nodes in the region along the specified direction in minimal time while maximizing link reliability. Moreover, packet transmissions are scheduled in an optimal manner in order to avoid collisions and save the number of transmissions. Both versions of the proposed solution are implemented and tested via simulations, whose results show that higher reliability can be achieved when more neighbor information is available and that our solution performs better than LBM and GeoTORA, two well-known geocasting solutions originally designed for terrestrial networks.

3) An under-ice localization solution, which can minimize the localization uncertainty for AUVs with high position uncertainty. External uncertainty offers a way to model the position of the AUV in a probabilistic way. This model is used to estimate the uncertainty resulted from localization techniques, as shown for our proposed Doppler-based localization and the standard distance-based localization. Algorithms are then proposed to minimize the position uncertainty and communication overhead. Our solution is implemented on WHOI modems and compared with several existing localization techniques using an acoustic communication emulator. It is shown that our approach achieves excellent localization results with low localization overhead.

Based on these inter-vehicle communication solutions, we proposed a team formation and steering solution using acoustic modems. Our team formation algorithm minimizes the time to form the planned geometry formation in minimal time while avoiding glider collisions. A novel team steering algorithm is devised to minimize the communication overhead for coordination. This algorithm uses Doppler shift information extracted from ongoing opportunistic communications to estimate the inter-vehicle distance and hence formation error is minimized. Our algorithm is a distributed solution as only local neighborhood information is used. Our team formation and steering algorithms were shown to be robust against ocean currents and communication impairments. The advantage of our solution over the solution that only uses satellite communications for coordination and over the solution without coordination is verified by simulations using a hybrid underwater acoustic communication emulator.

Future work will be to implement our proposed communication and coordination solutions on AUV platforms and evaluate their performance in ocean experiments. We are building our own AUV platform for performance evaluation of our solutions. Our proposed communication and coordination solutions can be used as part of the adaptive sampling solution using AUVs to reconstruct the ocean temperature field. Specifically, given a field to sense, the AUVs should coordinate to take measurements with minimal cost (such as time or energy) in order to reconstruct the field with admissible error. To achieve this, we will propose a framework that jointly considers the online field measurements and multi-AUV trajectory planning. Novel distributed algorithms that can minimize the sampling cost for one-round sampling will be proposed, and more underwater communication solutions will be designed to coordinate the vehicles. In this way, low sampling cost can be achieved while the computation complexity per vehicle can be reduced by using distributed algorithms.

We also plan to seamlessly integrate the underwater acoustic sensor networks, underwater robots or vehicles, and ocean observation infrastructure together. The performance of Ocean Observing Infrastructure (OOI), robots/vehicles, and sensors can all be improved when they share information with each other. On one hand, information from existing OOI can be leveraged by underwater robots, vehicles, or sensors for performance improvement. For example, trajectory planning of the robots can be made easier with the help of OOI. On the other hand, the sensed data can be used in existing OOI to reduce sensing errors (e.g., reducing ocean forecasting error). The integration of these systems can enable the seamless sensing of the ocean. To achieve this goal, we will further study integration issues among different systems, and develop solutions to enable the ubiquitous computing and sensing, e.g., solutions to jointly optimize the performance of the integrated system.

References

[1] R. E. Davis, N. E. Leonard, and D. M. Fratantoni. Routing Strategies for Underwater Gliders. *Deep-Sea Research II (Elsevier)*, 56(3):173–187, February 2009.

[2] I. F. Akyildiz, D. Pompili, and T. Melodia. Underwater Acoustic Sensor Networks: Research Challenges. *Ad Hoc Networks (Elsevier)*, 3(3):257–279, May 2005.

[3] Xiaojing Xiang, Zehua Zhou, and Xin Wang. Self-Adaptive On Demand Geographic Routing Protocols for Mobile Ad Hoc Networks. In *Proc. of IEEE International Conference on Computer Communications (INFOCOM)*, Anchorage, AL, May 2007.

[4] Milica Stojanovic. On the Relationship Between Capacity and Distance in an Underwater Acoustic Communication Channel. In *Proc. of ACM International Workshop on UnderWater Networks (WUWNet)*, Los Angeles, CA, September 2006.

[5] William S. Burdic. Underwater acoustic system analysis. In Alan V. Oppenheim, editor, *Prentice-Hall Signal Processing Series*, chapter 2, page 49. Prentice-Hall, 1984.

[6] M. Porter. BELLHOP Gaussian Beam/Finite Element Beam Code. `http://oalib.hlsresearch.com/Rays/index.html`.

[7] J. Jalbert, J. Baker, J. Duchesney, P. Pietryka, W. Dalton, D.R. Blidberg, S. Chappell, R. Nitzel, and K. Holappa. A Solar-powered Autonomous Underwater Vehicle. In *Proc. of IEEE OCEANS*, San Diego, CA, September 2003.

[8] R. Hine, S. Willcox, G. Hine, and T. Richardson. The Wave Glider: A Wave-Powered Autonomous Marine Vehicle. In *Proc. of IEEE OCEANS*, Biloxi, MS, October 2009.

[9] E. Fiorelli, N. E. Leonard, P. Bhatta, D. Paley, R. Bachmayer, and D.M. Fratantoni. Multi-AUV Control and Adaptive Sampling in Monterey Bay. *IEEE Journal of Oceanic Engineering*, 31(4):935–948, October 2006.

[10] D. Pompili and I. F. Akyildiz. A Cross-layer Communication Solution for Multimedia Applications in Underwater Acoustic Sensor Networks. In *Proc. of IEEE International Conference on Mobile Ad-hoc and Sensor systems (MASS)*, Atlanta, GA, October 2008.

[11] B. Chen and D. Pompili. QUO VADIS: QoS-aware Underwater Optimization Framework for Inter-vehicle Communication using Acoustic Directional Transducers. In *Proc. of IEEE Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, Salt Lake City, UT, June 2011.

[12] A.P. Aguiar and A.M. Pascoal. Coordinated path-following control for nonlinear systems with logic-based communication. In *Proc. of IEEE Conference on Decision and Control*, New Orleans, Louisiana, December 2007.

[13] A. Aguiary, J. Almeiday, M. Bayaty, B. Cardeiray, R. Cunhay, A. Hauslery, P. Mauryay, A. Oliveiray, A. Pascoaly, A. Pereira, M. Rufinoy, L. Sebastiaoy, C. Silvestrey, and F. Vanniy. Cooperative Autonomous Marine Vehicle Motion Control in the scope of the EU GREX Project: Theory and Practice. In *Proc. of IEEE OCEANS - Europe*, Bremen, Germany, May 2009.

[14] R. Ghabcheloo, A. P. Aguiar, A. Pascoal, C. Silvestre, I. Kaminer, and J. Hespanha. Coordinated Path-Following in the Presence of Communication Losses and Time Delays. *SIAM Journal on Control and Optimization*, 48:234–265, Feb 2009.

[15] R. Haraksim, L. Brignone, and J. Opderbecke. Multiple AUV control in an operational context: a leader - follower approach. In *Proc. of IEEE OCEANS - Europe*, Bremen, Germany, May 2009.

[16] D. Pompili and I. F. Akyildiz. Overview of Networking Protocols for Underwater Wireless Communications. *IEEE Communications Magazine*, 47(1):97–102, January 2009.

[17] H. Takagi and L. Kleinrock. Optimal transmission ranges for randomly distributed packet radio terminals. *IEEE Trans. on Communications*, COM-32(3):246–257, March 1984.

[18] G. G. Finn. Routing and addressing problems in large metropolitan scale Internetworks. Technical report, ISI, Storrs, CT, 1987.

[19] Evangelos Kranakis, Harvinder Singh, and Jorge Urrutia. Compass routing on geometric networks. In *Proc. of Canadian Conference on Computational Geometry*, Vancouver, Canada, August 1999.

[20] T. Melodia, D. Pompili, and I. F. Akyildiz. On the Interdependence of Distributed Topology Control and Geographical Routing in Ad Hoc and Sensor Networks. *IEEE Journal of Selected Areas in Communications*, 23(3):520–532, March 2005.

[21] Edward A. Carlson, Pierre-Philippe Beaujean, and Edgar An. Location-Aware Routing Protocol for Underwater Acoustic Networks. In *Proc. of IEEE International Conference on Engineering in the Ocean Environment (OCEANS)*, Boston, MA, September 2006.

[22] J. M. Montana, M. Stojanovic, and M. Zorzi. Focused Beam Routing Protocol for Underwater Acoustic Networks. In *Proc. of ACM International Workshop on Underwater Networks (WUWNet)*, San Francisco, CA, September 2008.

[23] Kevin Fall. A Delay-Tolerant Network Architecture for Challenged Internets. In *Proc. of ACM Special Interest Group on Data Communication (SIGCOMM)*, Karlsruhe, Germany, August 2003.

[24] S. Burleigh, A. Hooke, L. Torgerson, K. Fall, V. Cerf, B. Durst, K. Scott, and H. Weiss. Delay-Tolerant Networking:An Approach to Interplanetary Internet. *IEEE Communications Magazine*, 41(6):128C136, June 2003.

[25] Aruna Balasubramanian, Brian Levine, and Arun Venkataramani. DTN routing as a resource allocation problem. In *Proc. of ACM ACM Special Interest Group on Data Communication (SIGCOMM)*, August 2007.

[26] Thrasyvoulos Spyropoulos, Konstantinos Psounis, and Cauligi S. Raghavendra. Spray and wait: an efficient routing scheme for intermittently connected mobile networks. In *Proc. of the ACM SIGCOMM Workshop on Delay-Tolerant Networking (WDTN)*, Philadelphia, PA, August 2005.

[27] John Burgess, Brian Gallagher, David Jensen, and Brian Neil Levine. MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks. In *Proc. of Conference on Computer Communications (INFOCOM)*, Barcelona, Spain, April 2006.

[28] Colin Y.M. Chan and Mehul Motani. An Integrated Energy Efficient Data Retrieval Protocol for Underwater Delay Tolerant Networks. In *Proc. of IEEE International Conference on Engineering in the Ocean Environment (OCEANS) - Europe*, Aberdeen, Scotland, June 2007.

[29] Zheng Guo, Gioele Colombi, Bing Wang, Jun-Hong Cui, and Dario Maggiorini. Adaptive Routing in Underwater Delay/Disruption Tolerant Sensor Networks. In *Proc. of IEEE/IFIP Conference on Wireless on Demand Network Systems and Services (WONS)*, Garmisch-Partenkirchen, Germany, January 2008.

[30] Zheng Guo, Bing Wang, and Jun-Hong Cui. Prediction assisted single-copy routing in underwater delay tolerant networks. In *Proc. of IEEE Global Communications Conference (GLOBECOM)*, Miami, FL, December 2010.

[31] Eugenio Magistretti, Jiejun Kong, Uichin Lee, Mario Gerla, Paolo Bellavista, and Antonio Corradi. A Mobile Delay-tolerant Approach to Long-term Energy-efficient Underwater Sensor Networking. In *Proc. of IEEE Wireless Communications and Networking Conference (WCNC)*, Kowloon, Hong Kong, March 2007.

[32] H. Luo, Z. Guo, W. Dong, F. Hong, and Y. Zhao. LDB: Localization with Directional Beacons for Sparse 3D Underwater Acoustic Sensor Networks. *Journal of Networks*, 5(1):28–38, January 2010.

[33] Peng Xie, Jun Hong Cui, and Li Lao. VBF: Vector-Based Forwarding Protocol for Underwater Sensor Networks. In *Proc. of IFIP Networking*, Waterloo, Ontario, Canada, May 2005.

[34] Raja Jurdak, Cristina Videira Lopes, and Pierre Baldi. Battery lifetime estimation and optimization for underwater sensor networks. In S. Phoha, T. LaPorta, and C. Griffin, editors, *IEEE Sensor Network Operations*. Wiley-IEEE Press, 2004.

[35] L. Freitag, M. Grund, S. Singh, J. Partan, P. Koski, and K. Ball. The WHOI Micro-Modem: An Acoustic Communcations and Navigation System for Multiple Platforms. In *Proc. of IEEE International Conference on Engineering in the Ocean Environment (OCEANS)*, Washington DC, September 2005.

[36] Y. Ko and N. Vaidya. Geocasting in Mobile Ad Hoc Networks: Location-Based Multicast Algorithms. In *Proc. of IEEE Workshop on Mobile Computer Systems and Applications*, New Orleans, LA, February 1999.

[37] Y. Ko and N. Vaidya. GeoTORA: A Protocol for Geocasting in Mobile Ad Hoc Networks. In *Proc. of International Conference on Network Protocols (ICNP)*, Osaka, Japan, November 2000.

[38] V. D. Park and M. S. Corson. A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks. In *Proc. of IEEE Conference on Computer Communications (INFOCOM)*, Kobe, Japan, April 1997.

[39] I. Stojmenovic, A. P. Ruhil, and D. K. Lobiyal. Voronoi Diagram and Convex Hull Based Geocasting and Routing in Wireless Networks. *Journal of Wireless Communcations and Mobile Computing*, (6):247–258, June 2006.

[40] I. Stojmenovic. Geocasting with guaranteed delivery in sensor networks. *IEEE Wireless Communications*, 11(6):29–37, December 2004.

[41] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia. Routing with Guaranteed Delivery in Ad Hoc Wireless Networks. *Journal of Wireless Networks (Springer)*, 7(6):609–616, November 2001.

[42] C. T. Chang and S.-C. Tu. Obstacle-Free Geocasting Protocols for Single/Multi-Destination Short Message Services in Ad Hoc Networks. *Journal of Wireless Networks (Springer)*, 9(2):143–155, March 2003.

[43] J. Lian, K. Naik, Y. Liu, and L. Chen. Virtual Surrounding Face Geocasting with Guaranteed Message Delivery for Ad Hoc and Sensor Networks. *IEEE/ACM Transactions on Networking*, 17(1):200 – 211, February 2009.

[44] J.A. Sanchez, P.M. Ruiz, J. Liu, and I. Stojmenovic. Bandwidth-Efficient Geographic Multicast Routing Protocol for Wireless Sensor Networks. *IEEE Sensors Journal*, 7(5):627–636, May 2007.

[45] James C. Kinsey, Ryan M. Eustice, and Louis L. Whitcomb. A Survey of Underwater Vehicle Navigation: Recent Advances and New Challenges. In *Proc. of IFAC Conference of Manoeuvering and Control of Marine Craft*, Lisbon, Portugal, Sept 2006.

[46] Vijay Chandrasekhar, Winston KG Seah, Yoo Sang Choo, and How Voon Ee. Localization in underwater sensor networks: survey and challenges. In *Proc. of the ACM International Workshop on Underwater Networks (WUWNet)*, Sept 2006.

[47] M. Erol-Kantarci, L. M. Vieira, and M. Gerla. AUV-Aided Localization for UnderWater Sensor Networks. In *Proc. of International Conference on Wireless Algorithms, System and Applications (WASA)*, Chicago, IL, Aug 2007.

[48] Melike Erol, Luiz F. M. Vieira, and Mario Gerla. Localization with Dive'N'Rise (DNR) beacons for underwater acoustic sensor networks. In *Proceedings of the ACM Workshop on Underwater Networks (WUWNet)*, Sept 2007.

[49] Maurice F Fallon, Georgios Papadopoulos, John J Leonard, and Nicholas M Patrikalakis. Cooperative auv navigation using a single maneuvering surface craft. *International Journal of Robotics Research*, 29:1461–1474, October 2010.

[50] D. Mirza and C. Schurgers. Motion-Aware Self-localization for Underwater Networks. In *Proc. of ACM International Workshop on UnderWater Networks (WUWNet)*, San Franscisco, CA, September 2008.

[51] D. Mirza and C. Schurgers. Collaborative Tracking in Mobile Underwater Networks. In *Proc. of ACM International Workshop on UnderWater Networks (WUWNet)*, Berkeley, CA, November 2009.

[52] A. Alvarez, A. Caffaz, A. Caiti, G. Casalino, L. Gualdesi, A. Turetta, and R. Viviani. Folaga: A low-cost autonomous underwater vehicle combining glider and AUV capabilities. *Ocean Engineering (Elsevier)*, 36(1):24–38, January 2009.

[53] P. V. Teixeira, D. V. Dimarogonas, K. H. Johansson, and J. Sousa. Event-based Motion Coordination of Multiple Underwater Vehicles Under Disturbances. In *Proc. of IEEE OCEANS - Sydney*, Sydney, Australia, May 2010.

[54] M. Blain, S. Lemieux, and R. Houde. Implementation of a ROV navigation system using acoustic/Doppler sensors and Kalman filtering. In *Proc. of IEEE International Conference on Engineering in the Ocean Environment (OCEANS)*, San Diego, CA, September 2003.

[55] S. Williams and I Mahon. Simultaneous localisation and mapping on the Great Barrier Reef. In *Proc. of IEEE International Conference on Robotics and Automation (ICRA)*, New Orleans, LA, April 2004.

[56] B. Chen and D. Pompili. Team Formation and Steering Algorithms for Underwater Gliders using Acoustic Communication. *to appear in Computer Communications (Elsevier)*, 2012.

[57] I. S. Kulkarni and D. Pompili. Task Allocation for Networked Underwater Vehicles in Critical Missions. *IEEE Journal of Selected Areas in Communications*, 28(5):716–727, June 2010.

[58] George Casella and Roger L. Berger. *Statistical Inference*. Duxbury Press, 2 edition, 2001.

[59] Sung Joon Ahn. *Least Squares Orthogonal Distance Fitting of Curves and Surfaces in Space*, chapter 2, page 17. Springer, 1 edition, 2008.

[60] G. Strang. *Linear Algebra and Its Applications*, chapter 7, page 368. Thomson Brooks/Cole, 3 edition, 1988.

[61] G. A. F. Seber and C. J. Wild. *Nonlinear Regression*. John Wiley & Sons, 1989.

[62] Ioannis Rekleitis. A Particle Filter Tutorial for Mobile Robot Localization. *Technical Report TR-CIM-04-02, Centre for Intelligent Machines, McGill University, Montreal, Quebec, Canada*, 2004.

[63] Peter S. Maybeck. *Stochastic models, estimation, and control - Volume 1*. Academic Press, New York, 1979.

[64] Rudolph Van Der Merwe and Eric A. Wan. The Square-Root Unscented Kalman Filter for State and Parameter-Estimation. In *Proc. of International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Salt Lake City, UT, May 2001.

[65] Robert Grover Brown and Patrick Y. C. Hwang. *Introduction to Random Signals and Applied Kalman Filtering, 3rd edition*. John Wiley & Sons, Inc., Hoboken, NJ, 1996.

[66] Baozhi Chen, Patrick Hickey, and Dario Pompili. Trajectory-aware Communication Solution for Underwater Gliders using WHOI Micro-Modems. In *Proc. of IEEE Communications Society Conference on Sensor, Mesh, and Ad Hoc Communications and Networks (SECON)*, Boston, MA, June 2010.

[67] Jim Partan, Jim Kurose, and Brian Neil Levine. A Survey of Practical Issues in Underwater Networks. In *Proc. of ACM International Workshop on UnderWater Networks (WUWNet)*, Los Angeles, CA, September 2006.

[68] K. Perrine, K. Nieman, K. Lent, T. Henderson, T. Brudner, and B. Evans. Doppler estimation and correction for shallow underwater acoustic communications. In *Proc. of Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, November 2010.

[69] S. Mason, C. R. Berger, S. Zhou, and P. Willett. Detection, Synchronization, and Doppler Scale Estimation with Multicarrier Waveforms in Underwater Acoustic Communication. *IEEE Journal of Special Area in Communcications*, 26(9):1638–1649, December 2008.

[70] B. Li, J. Huang, S. Zhou, K. Ball, M. Stojanovic, L. Freitag, and P. Willett. MIMO-OFDM for High Rate Underwater Acoustic Communications. *IEEE Journal on Oceanic Engineering*, 34(4):634–644, October 2009.

[71] V. Gazi and K.M. Passino. A class of attraction/repulsion functions for stable swarm aggregations. In *Proc. of IEEE Conference on Decision and Control*, Las Vegas, NV, December 2002.

[72] Naomi Ehrich Leonard and Edward Fiorelli. Virtual Leaders, Artificial Potentials and Coordinated Control of Groups. In *Proc. of IEEE Conference on Decision and Control*, Orlando, FL, December 2001.

[73] WHOI Acoustic Communications Group. Micro-Modem Software Interface Guide (Ver 2.98). `http://acomms.whoi.edu/documents/uModem%20Software%20Interface%20Guide.pdf`.

[74] M-Audio. Delta 1010LT PCI Audio Interface. `http://www.m-audio.com/products/en_us/Delta1010LT.html/`.

[75] Gumstix Inc. Gumstix Connex Motherboards. `http://www.gumstix.net/Hardware/view/Hardware-Spec-Sheets/Connex-Spec-Sheet/112.html/`.

[76] Robert Humphrey. Playrec Version 2.1.1. `http://www.playrec.co.uk/`.

[77] S. M. Ross. *Introduction to Probability Models*. Academic Press, 8 edition, 2003.

[78] K. A. Morgansen, B. I. Triplett, and D. J. Klein. Geometric methods for modeling and control of free-swimming fin-actuated underwater vehicles. *IEEE Transactions on Robotics*, 23(6):1184–1199, December 2007.

[79] Robert J. Urick. *Principles of Underwater Sound*. McGraw-Hill, 1983.

[80] A.A. Eroshko, V.M. Kushnir, and A.M. Suvorov. Mapping currents in the northern Black Sea using OLT-type profilers. In *Proc. of the IEEE Working Conference on Current Measurement*, San Diego, CA, March 1999.

[81] B. Chen and D. Pompili. Demo 3: Coordination of Underwater Gliders using Acoustic Communications. `http://nsfcac.rutgers.edu/CPS/demo.html`.